Mapping of Legacy Relational Data for Semantic Web: A Survey

Smitha S Kumar
Research Scholar
Karpagam University
Coimbatore, India

M. Punithavalli, Ph.D.
Director, Dept of MCA
Sree Ramakrisha College of Engineering,
Coimbatore,India

ABSTRACT

Web has transformed the way people communicate with each other and the way business is conducted. Web in its current form is one of the most successful engineering artifacts, but the current technology suffers from limitations with respect to machine understandability. Semantic Web's promise of data integration requires the inclusion of the data in the relational databases. There are multiple approaches proposed to include this relational data into the sphere of semantic web. This paper is an attempt to compile and analyze the approaches proposed so far in literature and identify the various tools used to perform the mapping.

General Terms

Semantic Web, RDB2RDF

Keywords

Semantic Web, RDB2RDF, SPARQL, RDF.

1. INTRODUCTION

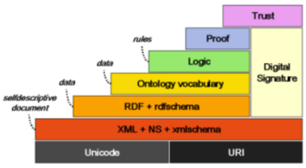
It was found in 2007, that internet accessible database contained 500 times more data compared to the static web [1]. The large quantity of data that is found on the web automatically generated from relational databases, is often referred to as the Deep Web (2).Semantic Web's promise of Web-scale data integration will only live up to its true promise with the inclusion of legacy relational database management systems (RDBMS) (3).The approach is to annotate the HTML pages with terms from ontology for machine consumption.

This paper is divided into sections, where section1 describes about the key terminologies w.r.t the Semantic web. Section2 describes about the approaches used to bridge the gap between RDBMS and Semantic web. Section3 describes a comparison on the above approaches and the existing tools to perform the translation.

2. SEMANTIC WEB LAYER CAKE

The Semantic Web principles are implemented in the layers of Web technologies and standards.

The layered architecture of the Semantic web is described below.



.

Fig 1: Semantic Web Layer Architecture

In Semantic Web, the resources are identified using the URI. Unicode make sure that we use the international character sets. The XML layer with namespace and schema definitions make sure we can integrate the semantic web definitions with xml based standards. RDF is used to make statements about the objects using URIs. RDF is the building block of semantic web. Ontology represents the domain using the constructs available. Additionally SPARQL is used as the query language of semantic web. The upper layers are currently being researched and focuses on the writing rules (Logic layer), executing rules (Proof layer) and the trust mechanism whether the application should trust the proof (Trust layer)[4].

RDF is a considered as the building block of semantic web. In RDF facts are represented as triples<subject, predicate, object>. An example is a sequence of (subject, predicate, object) terms, separated by whitespace and terminated by '.' after each triple[5].

<http://example.org/#spiderman><http://www.perceive.net/schemas/relationship/enemyOf><http://example.org/#green-goblin>.

subject is used to describe the resource. Here the subject refers to *spiderman*. The predicate indicate certain property of the resource, for example, name, dob, age etc...in the above example *enemyOf*. The object represents the value of the property or resource. Both subject and predicate is represented using an URI and the object can be represented as an URI or as literal.

SPARQL: SPARQL is a query language for the Semantic web like SQL for relational database.

3. MAP RELATIONAL DATA FOR SEMANTIC WEB

3.1 Motivation

One prime motivation for moving towards Semantic web is data Integration, integrating data residing in different sources. This includes integrating data in the relational database also. The flexibility of a graph model makes it's easy to integrate data. Graph can be merged by combining the nodes, and based on this merge, new information can be queried.

In September 2012, W3C published a standard language to describe the mapping between the relational database and RDF, R2RML (RDB to RDF Mapping Language).The standards are 1)Direct Mapping 2) R2RML.

3.2 Direct Mapping

Direct Mapping: This method generates an RDF graph from a relational database (data and schema). A simple example with two tables (People, Address) with single-column primary keys and one foreign key (address) reference between them:

Table 1. Table name: People

Id(int)	fname(char(10)	Address(int)
7	Bob	**18**
8	Sue	NULL

Table 2. Table name: Addresses

Id(int)	City (char(10))	State (char(2))
18	Cambridge	MA

Given a base IRI http://foo.example/DB/, the direct mapping of this database produces a direct graph:

@base <http://foo.example/DB/> .

@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<People/ID=7>rdf:type<People> .

<People/ID=7><People#ID>7 .

<People/ID=7><People#fname> "Bob" .

<People/ID=7><People#addr>18 .

<People/ID=7><People#ref-addr><Addresses/ID=18> .

<People/ID=8>rdf:type<People> .

<People/ID=8><People#ID>8 .

<People/ID=8><People#fname> "Sue" .

<Addresses/ID=18>rdf:type<Addresses> .

<Addresses/ID=18><Addresses#ID>18 .

<Addresses/ID=18><Addresses#city> "Cambridge" .

<Addresses/ID=18><Addresses#state> "MA" .

Subject: Each row in the table is a triple. The subject in RDF is formed by concatenating the base IRI, table name, primary key column name, primary key value.

Predicate for each column is an IRI formed from the concatenation of the base IRI, table name and the column name.

Each foreign key produces a triple with a predicate composed from the foreign key column names, the referenced table, and the referenced column names. The object of these triples is the row identifier (<Addresses/ID=18>) for the referenced triple. The direct mapping does not generate triples for NULL values. [4].

3.3 R2RML Mapping

In R2RML Mapping, user can customize the mapping, this permits the user to decide which columns, or which tables, should be used to generate the RDF Graph. The use of existing vocabularies is also allowed.

The first approach is to convert the data stored in the RDBMS into RDF, and then store it in a RDF triple store. (Extract-Transform-Load). RDF Triplestore is a database system specifically for Semantic web. The data model for Triple store is RDF. These database systems were called as triplestores, as the data stored in them were in the form of triples. The key feature of a triplestore is to perform inference [6].

Triple store examples:

- Jena,
- Mulgara
- AllegroGraph
- Virtuoso

The second approach is to create a mapping between the Relation data and the RDF (Wrapper Systems). In this approach the data remains in the relational database. A virtual RDF is generated. A SPARQL query is translated into an SQL query which can then be executed on the relational database. The results of this SPARQL query will then be translated into a SQL query. Hence this approach is to extract information from the existing relational databases and render it for the semantic web[7].

The above approaches could use the Direct Mapping or R2RML.

4. COMPARATIVE STUDY OF BOTH THE APPROACHES AND TOOLS

This section actually compares the above two approaches of using relational data in semantic web. The ETL based system demands the need of an additional triplestore, which could not be quite acceptable to the existing web applications. The reason for the same would be the additional investment for the triplestore. Another issue would be if the data is updated frequently, regular conversion becomes necessary to make the RDF data in sync with the RDBMS. These conversion processes are expensive. There can also be situations where the data in the triplestore is outdated in case if the conversion is not often done.

RDB2RDF Wrapper systems differ from the ETL based system in some of the above aspects. Here there is no need to convert and store the triples. Hence legacy system can still coexist with the Semantic web applications. As no conversion is done, the real time data will be looked at and hence no inconsistency in the data will be faced. The issue with the Wrapper system is the effectiveness of the SPARQL to SQL conversion and the execution speed. The performance factor is key to the success of such Wrapper systems.

4.1 Tools

Many Techniques and tools have been designed over the years in the area of RDBtoRDF. The following are some of the R2RML-compliant tools.

1) Ultrawrap

2) Virtuoso Universal Server

3) D2RQ

4.1.1 Ultrawrap

As discussed above the wrapper system has an advantage that it does not replicate the relation database content to support the web applications. Many of the wrapper systems have suffered performance issues, when it comes to translating SPARQL to SQL queries [7]

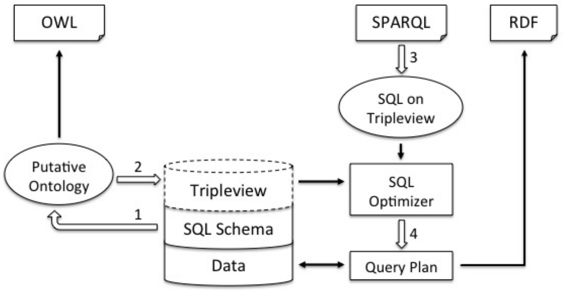


Fig 2: Architecture of Ultrawrap

Ultrawrap is a system that can execute SPARQL as fast as SQL. Ultrawrap has the following characteristics, supports W3C's R2RML and Direct Mapping, Automatic translation of Relational data to RDF, GUI-based Mapping, Integrated Linked Data and SPARQL end point [8].

4.1.2 Virtuoso universal server

This is a system designed for data management, access and integration. This system supports RDB2RDF query based transformation engine. The DBpedia project is operated on Virtuoso triplestore [9].

4.1.3 D2RQ

D2RQ is a system to access relational database as virtual read-only RDF graphs. This provides an integrated environment with multiple options to access relational data using different methods such as the SPARQL endpoint, Linked Data (content negotiation, HTTP 303 dereferencing), RDF dump, and Jena API based access (API calls are rewritten to SQL)[10].

5. CONCLUSION

This paper has emphasized the need for the relational data in the legacy systems to be included in the semantic web. This can enable software agents to work on the RDF data and perform reasoning for the users. The various approaches used and its effectiveness have also been discussed. RDB2RDFWrapper systems does not replicate database and hence is preferred for scenarios where data is very often updated .For systems where there are infrequent data updates, the ETL systems can be adopted as it could avoid the efficiency issues with the SPARQL-to-SQL conversions. The paper also discusses about the various commercial the R2RML-compliant tools

6. ACKNOWLEDGEMENTS

The author would like to that Dr.M .Punithavalli for her valuable inputs while writing this paper.

7. REFERENCES

[1] Accessing the deep web, Bin He, Mitesh Patel, Zhen Zhang, Kevin Chen-Chuan Chang.

[2] Siegfried Handschuh, Raphael Volz, Steffen Staab, Annotation for the Deep Web, IEEE Intelligent Systems, v.18 n.5, September 2003, pp.42-48.

[3] http://www.rdb2rdf.org/eswc2013-tutorial/

[4] http://www.w3.org/2001/12/semweb-fin/w3csw

[5] http://www.w3.org/TR/n-triples/#simple-triples

[6] W. Chang. Conversion of Relational Database into Triplestores. U.S. Patent 8,037,108, filed July 22, 2009, issued October 11, 2011

[7] Ultrawrap: SPARQL Execution on Relational Data - Juan F. Sequeda, Daniel P. Miranker

[8] http://www.w3.org/2001/sw/wiki/Ultrawrap

[9] http://virtuoso.openlinksw.com/whitepapers/relational%20rdf%20views%20mapping.html

[10] http://hal.archivesouvertes.fr/docs/00/90/35/68/PDF/Michel_Montagnat_Faron_2013_-_A_survey_of_RDB_to_RDF_translation_approaches_and_tools.pdf