

Using SPARQL/UPDATE to Extend RDB-to-RDF: A Mapping Approach

Yanping Chen¹ and Qian Yang²

¹*School of Computer Science Xi'an University of Posts and Telecommunications
Xi'an, China*

²*School of Computer Science Xi'an University of Posts and Telecommunications
Xi'an, China.*

¹*chenyanping@xupt.edu.cn*, ²*wkaiyq@126.com*

Abstract

Converting a huge amount of Web data stored in databases to semantic data is a critical requirement for the development of current Web. The literal to URI and implicit information issues existing in some traditional approaches of mapping Relational Database (RDB) to RDF are studied in this paper. In order to solve the above problems, SPARQL/UPDATE is used to extend the traditional approaches. This extensible approach not only solves the literal to URI problem but also gives methods to process implicit information with rules in the original RDF document. Experimental results show that the new approach enable semantic query engines to find more exact results.

Keywords: *Relational Database; RDF; SPARQL/UPDATE*

1. Introduction

As the latest development goal of the Web, Semantic Web's role is to automatically find, organize and synthesize information. However, the fact is that current Web is still lack of rich Semantic data. Usually, a huge amount of Web data is stored in databases. In order to provide rich Semantic data to the Web, it is necessary to convert relation data to Semantic data, which makes databases become one of the most important data sources to the Semantic Web. Semantic data or Semantic information is represented as a set of assertions which are called statements on the Semantic Web. Statements are consist of subject, predicate and object. Resource Description Framework (RDF) [1] is used as a general method for statement description that is implemented in Web resources, using a variety of syntax notations and data serialization formats. Thus, converting relational data in the databases to Semantic data means converting RDB to RDF and how to covert database to RDF becomes one of the most main topics of the Semantic Web research.

At the same time, the concept of the Semantic Sensor Web (SSW) [2] arises along with the development of the Semantic Web. The Semantic Sensor Web merges the sensor and Semantic Web technologies. The encoding of sensor descriptions and sensor observation data with Semantic Web languages enables more expressive representation, advanced access, and formal analysis of sensor data. Sensors around the globe currently collect avalanches of data about the world and the sensor data are usually stored in relational databases. The rapid development and deployment of sensor technology is intensifying the existing problem of too much data and not enough knowledge. RDF, OWL and other semantic technologies can be key enabling technologies for sensor networks to alleviate this glut because they will improve semantic interoperability and integration. RDF (or OWL) for sensors provides a framework for describing data models, attribute and classes of sensors. And Semantic Web techniques can greatly help with the problem of data integration and discovery as it helps map between relational database schema and RDF (or OWL) schema in a structured way. Thus, the effective approach of converting

RDB-to-RDF can also be used to solve the similar problems of SSW.

Up to now, there are many different mapping approaches to make the relational databases available on the Semantic Web. Different mapping languages are given by these approaches according to different requirements. This lead to the ongoing standardization effort of the World Wide Web Consortium (W3C) carried out in the RDB2RDF Working Group (WG). The mission of the RDB2RDF WG is to standardize a language for mapping relational databases into RDF and OWL. Paper [3] gives a survey of current techniques, tools and applications for mapping RDB to RDF. It is intended to enable the members of the RDB2RDF Working Group to collate the existing state of the art in mapping approached between RDB and RDF. The goal and contribution of paper [4] is to provide a feature-based comparison of the current RDB to RDF mapping languages. Four categories of mapping languages: direct mapping, read-only general-purpose mapping, read-write general-purpose mapping, and special-purpose mapping are proposed. Users can select a mapping language according to his demands and domain-specific applications. Based on the summary of the traditional mapping approaches proposed by paper [3] and [4], we can see that the issues of literal to URI and implicit information are still unsolved effectively.

URI plays an important role in the Semantic Web. RDF-related syntaxes such as Turtle, N-Triple, and SPARQL [5] use the < > brackets to tell a processor that something is an actual URI. However, URI itself as a datatype are typically not supported in RDB systems. Values representing URI are all stored as character literals, such as email address. Such literal values should be converted to valid URI in the RDB-to-RDF mapping processes. This feature is not supported by direct mapping [6-8] as well as Relational.OWL[9] of special-purpose mapping.

In the process of relational database design, much of the phases known as normalization are involved in dealing with redundancy of the data, including the storage of data that could instead be calculated dynamically when necessary. For example, in a family, Jane's parent is Gene and Gene's parent is Pat, so we can easily infer that Pat is Jane's grandparent and we don't need to store the relationship between Jane and Pat in DB. Information like this is called the implicit information. A relational database is a closed world with very fixed boundaries and users should combine two relational databases when they search for the implicit relationships between table rows from the different databases. But the combining process in DB is always much easier said than done. In Semantic Web and Linked Data [10] applications, the combination of two datasets like this is very common. And the easier data aggregation is one of RDF's greatest benefits. Users can make use of this benefit to look for relationships among the data and then to create new triples that make the implicit relationships explicit in RDF. Showing implicit information is necessary and it will enable query engine to deeply understand Web data and get exact answers that exist in the implicit information. However, no support for the process of implicit information is provided by the traditional mapping approaches.

According to the disadvantages of traditional approaches [6-9], this paper proposes an extensible approach to convert RDB to RDF with SPARQL/UPDATE [11] which is a new query language for Semantic Web. SPARQL/UPDATE makes up the read-only limitation of SPARQL and provides the methods to insert, delete, or modify RDF data. It can help to solve the issues of literal to URI and implicit information in the RDB-to-RDF mapping process.

The remainder of this paper is organized as follows. Section 2 describes the related work of this paper. The implementation details of literal to URI and implicit information are discussed in Section 3. The experiments and results are shown in Section 4. Finally, Section 5 provides the conclusion of this research.

2. Related Work

RDF is a data integration platform of different sources (especially for RDB), which

makes the mapping of RDB-to-RDF becomes an active field of Semantic Web research. The common steps in the traditional mapping approaches are as follows. The first step is extracting RDB schema and the Semantic in the relational data. Second, a mapping between RDB schema and ontology structure should be founded. Finally, based on the mapping, there are two different ways: (1) All relational data is converted to a Semantic Web format (*e.g.*, RDF) at once, eliminating the need for the original database or (2) Keep the data in the RDB and convert relational data on demand. For example, a mediator can translate Semantic Web requests (*e.g.*, SPARQL) to SQL and return the results in RDF formats.

Current approaches for mapping RDB-to-RDF as described in the last paragraph can be summarized as three steps (*i.e.*, ① in Figure 1):

- Extract Semantic information: Get the RDB schema and extract the implicit Semantic information in the relational data by analyzing the feature and the relationship in the RDB.
- Schema mapping: Map RDB tables and attributes to RDF classes and properties according to the mapping rules and the Semantic information. 1-M and M-M relationships in RDB tables are mapped to RDF properties.
- Data mapping: After the schema mapping, users can map RDB records to instance of RDF.

Although RDB2RDF Working Group are making efforts to standardize a language for mapping RDB-to-RDF, paper [4] argues that alternative languages still have the right to exist as they may provide features or simplicity requirement in certain use cases that the standard mapping languages cannot provide or explicit excludes. The direct mapping, as its name implies, is a direct approach for RDB-to-RDF mapping. Therefore, it is recommended in application scenarios where a direct representation of the relational schema is acceptable. Relational.OWL is highly specialized and does therefore neither implement nor require many of the describe features. In summary, it is important to use SPARQL/UPDATE to solve the literal to URI problem existing in the direct mapping and Relational.OWL.

The main goal of converting relational data to RDF is to enable Web agent to understand this data. However, Web agent cannot deeply understand RDF data because of the implicit information that exists in the relational data of RDB. It is necessary to look for relationships among the relational data and to then create new triples that make implicit relationship explicit. Adding the new triples to the original RDF document not only can provide the exact results to users but also can give more answers to users. Paper [6] is a simple table to class and attribute to property mapping extended with heuristics to find implicit subclass relationships in the RDB schema. The processing of the implicit information is not sufficient. Paper [12] proposes an approach to convert DB to RDF with extra user-defined rules. These rules that add the implicit data are considered as an extension to the DB. Unlike paper [12], this paper use SPARQL/UPDATE to deal with implicit information which makes the implicit information processing is more likely to achieve. SPIN [13] is a group of RDF properties that can be used to express rules. SPIN is designed to enable the easy development of software for managing SPARQL rules and constraint violates. However, rules that expressed by SPIN are added to ontology level. It is always difficult for users to update the standard ontology to add his rules. SPARQL/UPDATE allows data publishers to add rules to RDF level, which gives more flexibility to the users and avoids rules conflicts on ontology level.

Based on traditional mapping steps, this paper adds SPARQL/UPDATE operation to original RDF document. The added step is shown in ② of Figure 1.

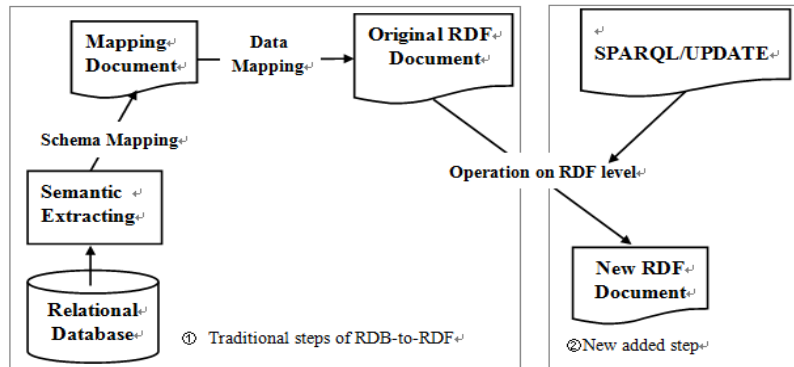


Figure 1. Steps of RDB-to-RDF

3. SPARQL/UPDATE

SPARQL is the W3C recommendation of a query language for RDF. It is limited to read-only access to RDF data. In order to close this gap, the SPARQL/UPDATE is proposed as a new query language to manipulate RDF data. It consists of three update operations: (1) INSERT DATA (Figure 2). This operation can insert new triples into an RDF graph. (2) DELETE DATA (Figure 3). This operation can remove triples from a RDF graph. (3) MODIFY DATA (Figure 4). This operation can delete and/or insert data based on triple templates that are matched against a triple pattern in a shared Where clause. The specification of MODEFY operation refers to as “DELETE/INSERT”. The query processor performs everything in the DELETE clause, and then performs the INSERT clause’s instructions.

To illustrate the added mapping process with SPARQL/UPDATE, we take the ISWC (International Semantic Web Conferences) database for example. ISWC database contains information about some conferences in Semantic Web field and other information such as published papers, authors and so on. Users can query information about the authors, the research topic related to the conferences and the publications of the authors.

```
INSERT DATA {
    TRIPLES
}
```

Figure 2. INSERT DATA Operation

```
DELETE DATA {
    TRIPLES
}
```

Figure 3. DELETE DATA Operation

```
MODIFY
DELETE {
    TEMPLATE}
INSERT {
    TEMPLATE}
WHERE {
    TEMPLATE}
```

Figure 4. MODIFY DATA Operation

3.1. Literal to URI Processing

Direct mapping and Relational.OWL are not support the feature of converting literals to URI. The MODIFY operation can help users to solve this problem. Using Relational.OWL mapping language as an example, character literals in the ISWC database are converted to original RDF statement in Turtle syntax:

```
...
dbs:conference    dbs:conferernce.uri

“http://annotation.semanticWeb.org/iswc/iswc.daml#ISWC_2002” .
...
```

As we know that the URI of this statement is invalid. So the corresponding MODIFY operations which are used to convert literals to valid URI are as follows:

```
DELETE{?s    dbs:conference    ?uri.}
INSERT{?s                                         dbs:conference
<http://annotation.semanticWeb.org/iswc/iswc.daml#ISWC_2002>.}
WHERE{?s    dbs:conference    ?uri.}
```

After the execution of this UPDATE operation on the original RDF document, character literals can be converted to valid URI in the RDF statement. Operations in Direct mapping are the same, so there is no need to say more.

Issues like this also exist in SSW. Literals that represent the location of sensors in databases can be converted to valid URI, which helps query engine to find the exact resource quickly.

3.2. Implicit Information Processing

Through querying the ISWC database, users can get information about authors, research topics of the conference, published papers and so on. For example: they can know the people that are participated in the 7th ISWC conference, the topic of their research and the email of the authors. Information like these is explicit in the database. Users can get the results through ordinary SPARQL query. However, what if users want to know more about the implicit information like: “knows” relationship that does not exist neither in relational database nor in the generated original RDF document. To meet the requirements of users querying questions like this, rules can be added to the original RDF document using SPARQL/UPDATE. For example, if A is the author of a paper Y, and B is also the author of the same paper Y, then we can know that: A knows B. Corresponding rules can be constructed as follows:

```
Rule:
<Body:
{ ?paper    rdf:type    vocab:papers.
  ?person1  rdf:type    vocab:persons.
  ?person2  rdf:type    vocab:persons.
  ?paper    dc:creator  ?person1.
  ?paper    dc:creator  ?person2.
  filter (?person1!=?person2) }
Head:
{ ?person1  foaf:knows  ?person2 }>
```

This rule contains two parts: Body and Head. The processor will execute the Head part only when the Body part is true. Besides, “knows” is a relation in foaf standard ontology.

Users can get the exact “knows” relationship after the execution of SPARQL CONSTRUCT sentence on the original RDF document. The specification phases of this process are as follows:

```
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix vocab: <http://localhost:2020/vocab/>
```

```
prefix dc: <http://purl.org/dc/elements/1.1/>
prefix foaf: <http://xmlns.com/foaf/0.1/>
construct {?person1 foaf:knows ?person2}
where {
    ?paper rdf:type vocab:papers.
    ?person1 rdf:type vocab:persons.
    ?person2 rdf:type vocab:persons.
    ?paper dc:creator ?person1.
    ?paper dc:creator ?person2.
    Filter (?person1!=?person2)}
```

Figure 5 shows the results using RDF/XML syntax.

The “knows” relationships between the authors are shown in Figure 5. The new triples that make the implicit relationships explicit can be added to the original RDF data using SPARQL/UPDATE INSERT operation. Considering the following SELECT sentences that are used to query the “knows” relationship:

```
SELECT DISTINCT ?x
```

```
WHERE {?x foaf:knows <http://localhost:2020/mapping-iswc.ttlpersons/10>.}
```

By running this query on the original RDF data, no results will be returned. However, after adding the new triples users can get the correct result:
<http://localhost:2020/mapping-iswc.ttlpersons/11>.

This conversion from relational database to RDF with added new triples is easy for data publishers and it provides more flexibility for query engines to search for more exact results.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:vocab="http://localhost:2020/vocab/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://purl.org/dc/elements/1.1/" >
  <rdf:Description rdf:about="http://localhost:2020/mapping-iswc.ttlpersons/10">
    <foaf:knows rdf:resource="http://localhost:2020/mapping-iswc.ttlpersons/11"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:2020/mapping-iswc.ttlpersons/11">
    <foaf:knows rdf:resource="http://localhost:2020/mapping-iswc.ttlpersons/10"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:2020/mapping-iswc.ttlpersons/1">
    <foaf:knows rdf:resource="http://localhost:2020/mapping-iswc.ttlpersons/2"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:2020/mapping-iswc.ttlpersons/6">
    <foaf:knows rdf:resource="http://localhost:2020/mapping-iswc.ttlpersons/9"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:2020/mapping-iswc.ttlpersons/2">
    <foaf:knows rdf:resource="http://localhost:2020/mapping-iswc.ttlpersons/1"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://localhost:2020/mapping-iswc.ttlpersons/9">
    <foaf:knows rdf:resource="http://localhost:2020/mapping-iswc.ttlpersons/6"/>
  </rdf:Description>
</rdf:RDF>
```

Figure 5. Results of the rule

4. Analysis of Experiments and Results

The proposed approach and model can be used to deal with different ontologies (RDF, OWL) and different databases (MySQL, SQL). A prototype for this experiment is implemented using MySQL, JAVA programming language, Eclipse and Jena tools. First, the relationships between tables of ISWC relational database are shown in Figure 6.

Second, D2RQ [14] is used as an auxiliary tool to convert ISWC database to RDF automatically. For example: Tables in database are mapped to ontology classes. Attribute is mapped to ontology property. The m-m relationship existing in database is considered as two 1-m relationship. And relationship like this will be mapped to RDF properties instead of classes. The D2RQ mapping language provides the URI pattern mechanism of assigning identifiers to the instances in the database. This paper will use absolute URI

pattern to configurate each table object. A part of the mapping result is shown in Figure 7. Users can get the original RDF data according to the mapping rules below.

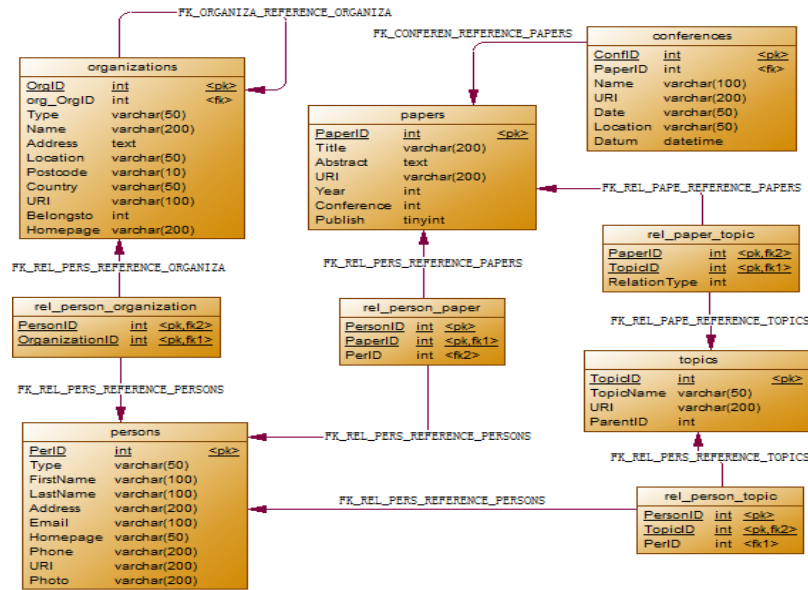


Figure 6. Relationships Between Tables of ISWC

```
# Table papers
map:Papers a d2rq:ClassMap;
d2rq:dataStorage map:database;
d2rq:uriPattern "papers/@ @papers.PaperID@@";
d2rq:condition "papers.Publish = 1";
d2rq:class iswc:InProceedings;
map:papers_Title a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:Papers;
d2rq:property dc:title;
d2rq:column "papers.Title";
d2rq:lang "en";
map:papers_Abstract a d2rq:PropertyBridge;
d2rq:belongsToClassMap map:Papers;
d2rq:property dcterms:abstract;
d2rq:column "papers.Abstract";
```

Figure 7. Mapping Results of ISWC to RDF

Based on the original RDF document, this paper gives the algorithm of literal to URI and implicit information:

1) Operation of literal to URI

Input: Original RDF document

Output: new RDF document

For(each instance)

{ for(each triple of instance)

{check up each object of triple;

if (the content of object start with "http://")

{

Construct a SPARQL/UPDATE DELETE sentence;

Use <> brackets with the content of object inside to replace the original object;

Construct a SPARQL/UPDATE INSERT sentence with new object;

Execute the MODIFY operation;

```
    }  
  }  
}  
2) Implicit information Operation  
Inputs: original RDF document, rules  
Output: new RDF document  
For (each rule)  
{  
  Extract the Head part of the rule to construct CONSTRUCT sentence of  
SPARQL;  
  Extract the Body part of the rule to construct WHERE sentence of SPARQL;  
  Execute the SPARQL query operation on the original RDF document;  
  if( SPARQL query results exist)  
  {  
    Extract results to construct INSERT sentence;  
    Execute the INSERT operation;  
  }  
}
```

Comparing to paper [6] which only find implicit subclass relationships, SPARQL/UPDATE can find all possible information that is implicit and make it explicit.

The extensible approach in this paper can be also used to handle with issues that exist in the SSW. For example, users need to get information of the sensor location and time attribute to describe the space-time information in the environment monitoring. However, in traditional monitoring system, data of location and time are usually stored separately in different tables of relational databases. When they are mapped to RDF, users can use SPARQL/UPDATE method to construct rules to get exact space-time information instead of making much calculation. This can help to form fine granularity focus service mode for the user.

5. Conclusion

In order to make the relational data available on the Semantic Web, an extensible RDB-to-RDF approach is proposed. After the analysis of traditional RDB-to-RDF mapping approaches and steps, SPARQL/UPDATE is used to solve the problem of literal to URI and implicit information existing in the direct mapping and Relational.OWL. Users can get more results from the new triples. Besides, this approach gives more flexibility for SSW organization to handle with monitoring data.

Acknowledgments

Supported by Natural Science Basic Research Plan in Shannxi Province of China (Program No.2012JQ8029) and Scientific Research Program Funded by Shannxi Provincial Education Department (Program No.12JK0938 and Program No.2013JK1130).

References

- [1] G. Klyne, J. J. Carroll, B. McBride, "Resource Description Framework (RDF): concepts and abstract syntax", (2004). <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>.
- [2] A. Sheth, C. Henson and S. S. Sahoo, "Semantic sensor Web", vol. 12, no. 4, (2008), pp. 78-83.
- [3] S. S. Sahoo, W. Halb, S. Hellmann and K. Ldehen *et al.*, "A Survey of Current Approaches for Mapping of Relational Databases to RDF", http://www.w3.org/2005/Incubator/rdb2rdf/RDB2RDF_SurveyReport_01082009.pdf., (2009).
- [4] M. Hert, G. Reif and H. C. Gall, "A comparison of RDB-to-RDF mapping languages", I-Semantics'11 Proceedings of the 7th International Conference on Semantic Systems", (2011), pp. 25-32.
- [5] E. Prud'hommeaux and A. Seaborne, "SPARQL Query Language for RDFW3C Recommendation", (2008) January 15, <http://www.w3.org/TR/rdf-sparql-query/>.

- [6] T. Berners-Lee, "Relational Databases On The Semantic Web", (2009), <http://www.w3.org/DesignIssues/RDB-RDF.html>.
- [7] W. Hu and Y. Z. Qu, "Discovering Simple Mappings Between Relational Database Schemas and Ontologies", In proceedings of the 6th International and 2nd Asian Semantic Web Conference, (2007), November 11-15, vol. 4825, pp. 225-238, Busan, Korea.
- [8] SquirrelRDF, (2011). <http://jena.sourceforge.net/SquirrelRDF/>.
- [9] C. P. De Laborda and S. Conrad, "Relational, OWL-A Data and Schema Representation Format Based on OWL", In Proceedings of the 2nd Asia-Pacific Conference on Conceptual Modelling, (2005).
- [10] A. Sarkar, U. Marjit and U. Biswas, "Linked data generation for the university data from legacy database", International Journal of Web & Semantic Technology 2, (2011), pp. 21-31.
- [11] A. Seaborne, G. Manjunath, C. Bizer, J. Breslin, S. Das and I. Davis et al., "SPARQL Update-A Language for Updating RDF Graphs", (2008), W3C Member Submission. <http://www.w3.org/Submission/2008/SUBM-SPARQL-Update-20080715/>.
- [12] M. Farouk, M. Ishizuka. An Extensible Approach for Mapping Relational DB to RDF, In Proceedings of the 2012 Japan-Egypt Conference on Electronics, Communications and Computers, JEC-ECC2012, (2012), March 6-9, pp. 163-167.
- [13] H. Knublauch, J. A. Hendler, K. Idehen, "SPIN-Overview and Motivation", (2011), <http://www.w3.org/Submission/2011/SUBM-spin-overview-20110222/>.
- [14] C. Bizer and A. Seaborne, "D2RQ-Treating Non-RDF Databases as Virtual RDF Graphs", In Proceedings of the 3rd International Semantic Web Conference, (2004).

Authors



YANPING CHEN



QIAN YANG

