Mapping Relational Databases into OWL Ontology

Haiyun Ling [1], Shufeng Zhou [2]

[1] College of Computer Science
Liaocheng University, Liaocheng 252059, P.R. China
1inghaiyun@lcu.edu.cn
[2] College of Mathematics Science
Liaocheng University, Liaocheng 252059, P.R. China
zhoushufeng@lcu.edu.cn

***Abstract*—In this paper we present an approach of RDF graph generation from relational databases. This approach consists of structures creation of ontology including classes, properties, hierarchy, cardinality and instances creation. This approach not only saves efforts in developing the Semantic Web but also makes an amount of data in relational databases machine understandable for the Semantic Web. The experimental results show that the approach is feasible and efficient, and is implementable with Java language.**

Keyword- Semantic Web, RDF, Ontology, Relational Database

I. INTRODUCTION

The Semantic Web provides a common framework that allows data to be shared and reused across communities and applications [1]. The Semantic Web is based on Resource Description Framework (RDF) [2] data model, which is a standard model for data interchange on the Web. Thus the success of the Semantic Web mainly depends on mass creation of RDF. The development of the current web of documents into the Semantic Web requires the inclusion of large quantities of data stored in relational databases (RDB). The RDF graph generation from RDB has been the focus of research work in diverse domains.

Different researches are investigated in RDB migrations focusing on different domains. Most existing approaches are restricted by a range of assumptions and characteristics such as the respect of the 3rd Normal Form and the integrity constraints.

Several approaches have been presented that directly map relational schemas to ontology languages [3].The W3C RDB2RDF Working Group published a direct mapping standard that focuses on translating relational databases instances to RDF [4]. Therefore, it is necessary to study the RDF generation in the Semantic Web community.

RDF is a machine-readable language that can be used to describe various subject (resources), properties and values. But RDF provides no mechanisms for describing these properties, nor does it provide any mechanisms for describing the relationships between these properties and other resources. However, ontology is a formal explicit specification of a shared conceptualization of a domain of interest and may include descriptions of classes, properties and relationships between them [5]. Ontology usually defines terms used in RDF document. For simplicity, in this paper, we adopt the Web Ontology Language (OWL) [6] as ontology description language.

In literature [7] the authors propose an approach to migrate relational data into ontologies; some rules of learning ontologies from relational databases were represented in literature [8]. These approaches are used to acquire ontologies instances from relational databases. In this paper we propose another approach based on above, which can be used for integrating data that are scattered across many different domains.

In our approach we have developed a prototype to create ontology from relational database. This prototype extracts schema metadata of database then transforms it into a canonical data model to facilitate the migration process, after the system generates the structure of OWL ontology and the data of RDF is stored in an OWL document. We have conducted experiment tests on the effectiveness of the prototype. The experimental results show that the approach is implementable with Java language.

The paper is organized as follows. Section II briefly gives some basic concepts. Section III describes the rules of ontology construction from relational database schema in detail. Section IV demonstrates the implementation of prototype and Section V describes the related work. Section VI draws the conclusion of the paper with the future work.

II. BASIC CONCEPTS

RDF describes things by making statements about a resource. RDF statements are triples consisting of a subject, a predicate and an object in a uniform way and facilitate machine understandable. The subject denotes the resource being described; the predicate (the property) denotes aspects of the resource and expresses a relationship between the subject and the object; the object is a value of property. An RDF graph is a set of such RDF triples (also called RDF dataset, or simply a dataset). The set of nodes of an RDF graph is the set of subjects and objects of triples in the graph.

The OWL is a language for defining and instantiating ontologies and it can be used to explicitly represent the meaning of terms in vocabularies and the relationships between them. The OWL is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML and RDF by providing additional expressive power along with a formal semantics.

Relational databases usually is a collection of relations [9]. Each relation is defined as a set of tuples that have the same attributes; each tuple consists of a set of attribute values. A tuple usually represents an object and information about the object. Objects are typically physical objects or concepts. A relation is usually described as a table, which is organized into rows and columns. All the data referenced by an attribute are in the same domain and conform to the same constraints.

In addition, a primary key defines a relationship within a database. The primary key comprises a single attribute or a set of attributes. A foreign key is a reference to primary key in another relation, meaning that the referencing relation has the values of a key in the referenced relation. The foreign key need not have unique values in the referencing relation. Foreign keys effectively use the values of attributes in the referenced relation to restrict the domain of one or more attributes in the referencing relation. This way references can be made to link information together.

Through the above instruction, we know that there are some similarities between RDF and relational database model. Class in OWL ontology corresponds to relation in database; property is kind of binary relationship. Thus we give some principles for correspondences between relational databases schema and OWL ontology. Maps relations to OWL classes; maps attributes to OWL object properties or data-type properties and maps the values of attribute to instances of OWL classes.

Given a relational database schema shown in Table I as an example to illustrate the translation below.

TABLE I RELATIONAL DATABASE SCHEMA

Relation	**Primary key**	**Foreign key**
Customer(*cid* string, *name* string, *city* string)	*cid*	*city* referring to *City.cid*
VIPcustomer(*cid* string, *year* int)	*cid*	N/A
Product(*pid* string, *name* string, *sid* string)	*pid*	*sid* refferring to *Supplier.sid*
Order(*cid* string, *pid* string)	*cid*, *pid*	*cid* referring to *Customer.cid* *pid* referring to *Product.pid*
City(*cid* string, *name* string)	*cid*	N/A
Supplier(*sid* string, *name* string)	*sid*	N/A

III. RULES FOR ONTOLOGY CONSTRUCTION

OWL ontology is constructed from relational database according to rules described below. The ontology construction process proposed in our paper consists of four steps: capturing schema metadata, analysing metadata, validating and refining ontology. In this paper, we make assumptions that all relational databases are at least up to third normal form.

A. *Mapping for OWL Classes.*

Rule 1. Each relation in a relational database should be mapped to a class if the relation is not a binary relation.

According to Rule 1, the relations *Customer*, *VIPcustomer*, *Product*, *City* and *Supplier* can be mapped to classes respectively.

```
< owl:Class rdf :ID = "Customer"/ >
< owl:Class rdf :ID = "VIPcustomer"/ >
< owl:Class rdf :ID = "Product"/ >
< owl:Class rdf :ID = "City"/ >
< owl:Class rdf :ID = "Supplier"/ >
```

Rule 2. For two relations, if they have the same primary key and the values of tuples of one relation in primary attributes belongs to others, then the relations can be mapped to sub-class of the other.

According to Rule 2, the relation *VIPcustomer* can be mapped to sub-class of Customer.

```
< owl:Class rdf :ID = "VIPcustomer" >
   < rdfs:subClassOf  rdf :resource = "Customer" / >
< / owl:Class >
```

B. *Mapping for Properties.*

Rule 3. For relation in relational databases, an attribute should be mapped to a data-type property, if the attribute does not belong to foreign keys of the relation. The domain of the data-type property is class corresponding to the relation, range the built-in data-type in XML schema [10] corresponding to the attribute.

Fox example, according to the Rule 3 we can get the data-type properties *Customer.cid* and *Customer.name* corresponding to attributes *cid* and *name* in relation *Customer* respectively.

```
< owl:DatatypeProperty rdf :ID = "Customer.cid" >
  < rdfs:domain rdf :resource = "Customer" / >
  < rdfs:range rdf :resource = "xsd:string" / >
< /owl:DatatypeProperty >
< owl:DatatypeProperty rdf :ID = "Customer.name" >
  < rdfs:domain rdf :resource = "Customer" / >
  < rdfs:range rdf :resource = "xsd:string" / >
< /owl:DatatypeProperty >
```

Rule 4. For relation in relational databases, two inverse object properties should be generated for an attribute, if the attribute belongs to foreign keys of non-binary relation. For one object property, the domain is the class corresponding to the relation; range is the class corresponding to the referred relation.

According to the Rule 4, we can get two object properties *Customer.inCity* and *City.hasCustomer*.

```
< owl:ObjectProperty rdf :ID = "Customer.inCity" >
  < rdfs:domain rdf :resource = "Customer" / >
  < rdfs:range rdf :resource = "City" / >
< / owl:ObjectProperty >
< owl:ObjectProperty rdf :ID = "City.hasCustomer" >
  < rdfs:domain rdf :resource = "City" / >
  < rdfs:range rdf :resource = "Customer" / >
  < owl:inverserOf  rdf :resource = "Customer.inCity" / >
< / owl:ObjectProperty >
```

Likewise we can get two object properties *Product.ofSupplier* and *Supplier.product*.

```
< owl:ObjectProperty rdf :ID = "Product.ofSupplier" >
  < rdfs:domain rdf :resource = "Product" / >
  < rdfs:range rdf :resource = "Supplier" / >
< / owl:ObjectProperty >
< owl:ObjectProperty rdf :ID = "Supplier.hasProduct" >
  < rdfs:domain rdf :resource = "Supplier" / >
  < rdfs:range rdf :resource = "Product" / >
  < owl:inverserOf  rdf :resource = "Product.ofSupplier" / >
< / owl:ObjectProperty >
```

Rule 5. For relation in relational databases, two object properties should be generated for two attributes, if the relation is a binary relation. The domain and range of the two object properties is reversed.

According to the Rule 5, two object properties should be generated for two attributes in the relation *Order* as below.

```
< owl:ObjectProperty rdf :ID = "Custsomer.ordering" >
  < rdfs:domain rdf :resource = "Customer" / >
  < rdfs:range rdf :resource = "Product" / >
< /owl:ObjectProperty >
< owl:ObjectProperty rdf :ID = "Product.orderingCutomer" >
  < rdfs:domain rdf :resource = "Product" / >
  < rdfs:range rdf :resource = "Customer" / >
  < owl:inverserOf  rdf :resource = "Customer.ordering" / >
< /owl:ObjectProperty >
```

C. *Rules for Cardinality.*

Rule 6. For a relation and an attribute, the maximal and minimal cardinality of the property corresponding to the attribute is set as 1, if the attribute is primary key or foreign key.

Rule 7. For a relation and an attribute, the minimal cardinality of the property corresponding to the attribute is set as 1, if the attribute is declared as NOT NULL.

Rule 8. For a relation and an attribute, the maximal cardinality of the property corresponding to the attribute is set as 1, if the attribute is declared as UNIQUE.

According to Rule 6, the maximal cardinality and minimal cardinality of property *Customer.cid* is 1.

```
< owl:Restriction >
   < owl:onProperty rdf:resource = "Customer.cid" / >
   < owl:maxCardinality > 1 < / owl:maxCardinality >
   < owl:minCardinality > 1 < / owl:minCardinality >
< / owl:Restriction >
```

D. Mapping for Instances.

The tuples of the relation can be translated to the instances applying to the mapping rules below.

Rule 9. For a relation, each tuple can be mapped to instances of class corresponding to the relation and each value of attribute can be mapped to the value of property corresponding to the attribute, if the relation is translated to class.

For instances, the tuples of relation *Customer* are (‘*CU201*’, ‘*Han*’, ‘*C01*’), (‘*CU202*’, ‘*Li*’, ‘*C03*’); the tuples of relation *Product* are (‘*P01*’, ‘*Computer*’, ‘*S02*’), (‘*P02*’, ‘*Router*’, ‘*S02*’); the tuples of relation *Order* are (‘*CU201*’, ‘*P01*’), (‘*CU202*’, ‘*P02*’); the tuples of relation *City* are (‘*C01*’, ‘*Beijing*’), (‘*C03*’, ‘*Guangzhou*’). To uniquely identify the resource, we use the primary key value as the URIs. The instances of Class *Customer* are listed below.

```
< Customer rdf:ID = "CU201" >
   < Customer.cid rdf:datatype = "xsd:string" > CU201 < /Customer.cid >
   < Customer.name rdf:datatype = "xsd:string" > Han < /Customer.name >
   < Customer.inCity rdf:resource = "C01"/ >
   < Customer.ordering rdf:resource = "P01"/ >
< /Customer >
< Customer rdf:ID = "CU202" >
   < Customer.cid rdf:datatype = "xsd:string" > CU202 < /Customer.cid >
   < Customer.name rdf:datatype = "xsd:string" > Li < /Customer.name >
   < Customer.inCity rdf:resource = "C03"/ >
   < Customer.ordering rdf:resource = "P02"/ >
< /Customer >
```

The instances of Class *Product* are listed below.

```
< Product rdf:ID = "P01" >
   < Product.pid rdf:datatype = "xsd:string" > P01 < /Product.pid >
   < Product.name rdf:datatype = "xsd:string" > Computer < /Product.name >
   < Product.ofSupplier rdf:resource = "S02"/ >
   < Product.orderingCustomer rdf:resource = "CU201"/ >
< /Product >
< Product rdf:ID = "P02" >
   < Product.pid rdf:datatype = "xsd:string" > P01 < /Product.pid >
   < Product.name rdf:datatype = "xsd:string" > router < /Product.name >
   < Product.ofSupplier rdf:resource = "S02"/ >
   < Product.orderingCustomer rdf:resource = "CU202"/ >
< /Product >
```

The instances of Class *City* are listed below.

```
< City rdf:ID = "C01" >
   < City.cid rdf:datatype = "xsd:string" > C01 < /City.cid >
   < City.name rdf:datatype = "xsd:string" > Beijing < /City.name >
   < City.hasCustomer rdf:resource = "C01"/ >
< /City >
< City rdf:ID = "C03" >
   < City.cid rdf:datatype = "xsd:string" > C01 < /City.cid >
   < City.name rdf:datatype = "xsd:string" > Guangzhou < /City.name >
   < City.hasCustomer rdf:resource = "C03"/ >
< /City >
```

From the instances above we know that the values of attribute *inCity* in *Customer* are not simply represented as plaint literal instead of object properties added to the class *Customer*. This object properties link the resources between nodes of *Customer* and the nodes of *City*. Likewise the relation *Order* in database is not mapped as a class instead of two inverse object properties. But the *Order* should be mapped as a class if the relation *Order* has another attribute except two foreign keys.

IV. IMPLEMENTATION.

To demonstrate the effectiveness and validity of our approach, a prototype has been developed, which can directly and automatically maps relational database into RDF graph. The main processes are: First, the prototype uses database analyser to extract schema information such as primary keys, foreign keys and inclusion relation from relational databases utilizing reverse engineering technique [11]. Second, maps relations into classes or subclasses, attributes into properties respectively according to the rules above and writes the ontology into a document.

The prototype is implemented in Java [12] Version 1.6 based on Jena [13] 2.6.0 API. Jena is a Java framework for building Semantic Web applications. In addition, we use MySQL Version 5.0.28 [14] as the backend database.

V. RELATED WORK.

There are several approaches for addressing the issue of mapping from relational databases to RDF.

In literature [15], the authors describe DB2OWL using the tables to concepts and columns to predicates approach and the mapping correspondences are saved in a R2O [16] document. In [17], Li proposes a semi-automatic ontology acquisition method from relational database using a group of rules. However, other approaches, such as D2R [18], use a declarative, XML-based language to describe mappings between relational database models and ontologies implemented in RDF Schema [19]. In D2R, basic concept mappings are defined using class maps that assign ontology concepts to database sets. The class map is also the container of a set of attributes and relation mapping elements called bridges. Many mapping languages and approaches were explored leading to the ongoing standardization effort of the World Wide Web Consortium (W3C) carried out in the RDB2RDF Working Group [20]. *RDOTE* is a recent proposal for the automatic and custom mapping and transportation of data residing in RDB into RDF [21]. The combination of RDF/OWL has been used in many specific scenarios for the construction of flexible data semantic models [22-24]. In [22], the method is to build ontology from conceptual databases schemas, which similar to our approach. Our approach generates RDF graph from relational database directly and automatically.

VI. CONCLUSION

In this paper, a set of mapping rules is presented to generate RDF graph from relational database schema and content. This approach can restore the semantics of relational database schema and data information more efficiently. There is a need to improve the approach because ontology generation from relational database plays an import role for information interoperability. The main contributions of this paper are listed as follows. The ontology is constructed automatically from relational databases. This approach can explicitly describe the implicit conceptual relationships in relational database.

ACKNOWLEDGMENT

We wish to thank the anonymous referees for their valuable comments and suggestions. And the work was financially supported by: Key Laboratory of Intelligent Information Processing and Network Security in Liaocheng University, Natural Science Foundation of Liaocheng University (X051034).

REFERENCES

[1] Grigoris Antoniou, Frank Van Harmelen, A Semantic Web Primer, The MIT Press, Cambridge Massachusetts, London, England, 2004.

[2] Graham Klyne, Jeremy J. Carroll, Resource Description Framework (RDF): Concepts and Abstract Syntax (W3C Recommendation 10 February 2004)[EB/OL]. *http://www.w3.org/TR/rdf-concepts/*, (last modified on 10 February 2004).

[3] Juan F. Sequeda, Syed Hamid Tirmizi, Oscar Corcho, Daniel P. Miranker, "Survey of Directly Mapping SQL Databases to the Semantic Web", The Knowledge Engineering Review, vol.26, no.4, pp.445-486, 2011.

[4] Marcelo Arenas, Alexandre Bertails, Eric Prud'hommeaux, Juan Sequeda, A Direct Mapping of Relational Data to RDF (W3C Recommendation 27 September 2012)[EB/OL]. *http://www.w3.org/TR/rdb-direct-mapping/*, (last modified on 27 September, 2012).

[5] Rudi Studer, V. Richard Benjamins, Dieter Fensel, "Knowledge engineering: Principles and methods", Data & Knowledge Engineering, vol.25, no.1-2, pp.161-197, 1998.

[6] Michael K. Smith, Chris Welty, Deborah L. McGuinness, OWL Web Ontology Language Guide (W3C Recommendation 10 February 2004)[EB/OL]. *http://www.w3.org/TR/owl-features/*, (last modified on 10 February 2004).

[7] Ljiljana Stojanovic, Nenad Stojanovic, Raphael Volz, "Migrating data-intensive web sites into the Semantic Web", In Proceedings of the 2002 ACM symposium on Applied computing (SAC '02), pp.1100-1107, ACM, 2002.

[8] Man Li, Xiaoyong Du, Shan Wang, "Learning ontology from relational database", In Proceedings of 2005 International Conference on Machine Learning and Cybernetics (ICML2005), pp.3410-3415, 2005.

[9] E. F. Codd, "A relational model of data for large shared data banks", Commun. ACM, vol.13, no.6, pp.377-387, 1970.

[10] David C. Fallside, Priscilla Walmsley, XML Schema Part 0: Primer Second Edition (W3C Recommendation 28 October 2004)[EB/OL]. *http://www.w3.org/TR/xmlschema-0/*, (last modified on 28 October 2004).

[11] Roger H. L. Chiang, Terence M. Barron, Veda C. Storey, "A Framework for the Design and Evaluation of Reverse Engineering Methods for Relational Databases", Data & Knowledge Engineering, vol.21, no.1, pp.57-77, 1996.

[12] Sun Microsystems, The Source for Java Developers, http://java.sun.com/, (last access on 1 October 2013).

[13] HP, Jena Semantic Web Framework, http://jena.sourceforge.net/, (last access on 1 October 2013).

[14] Sun Microsystems, MySQL Guide to MySQL Documentation, http://dev.mysql.com/doc/mysqldoc-guide/en/index.html, (last access on 1 October 2013).

[15] Nadine Cullot, Raji Ghawi, Kokou Yétongnon, "DB2OWL: A Tool for Automatic Database-to-Ontology Mapping", In Proceedings of 15th Italian Symposium on Advanced Database System (SEBD 2007), pp.491-494, 2007.

[16] Jesús Barrasa, Óscar Corcho, Asunción Gómez-Pérez, "R2O: an Extensible and Semantically based Database-to-Ontology Mapping Language", In Proceedings of Second Workshop on Semantic Web and Database (SWDB2004), pp.1-17, 2004.

[17] Man Li, Xiaoyong Du, Shan Wang, A Semi-automatic Ontology Acquisition Method for the Semantic Web. Advances in Web-Age Information Management, vol. 3739, pp.209-220. Springer Berlin Heidelberg 2005.

[18] Christian Bizer, "D2R MAP - A Database to RDF Mapping Language", In Proceedings of the 12th International World Wide Web Conference (WWW2003), 2003.

[19] Dan Brickley, Ramanathan V. Guha, RDF Vocabulary Description Language 1.0: RDF Schema (W3C Recommendation)[EB/OL]. *http://www.w3.org/TR/2004/REC-rdf-schema-20040210/*, (last modified on 10 February 2004).

[20] Matthias Hert, Gerald Reif, Harald C. Gall, "A Comparison of RDB-to-RDF Mapping Languages", In Proceedings of 7th International Conference on Semantic System (I-Semantics 2011), ACM, 2011.

[21] Konstantinos N. Vavliakis, Theofanis K. Grollios, Pericles A. Mitkas, "RDOTE - Publishing Relational Databases into the Semantic Web", Journal of Systems and Software, vol.86, no.1, pp.89-99, 2013.

[22] Heru Agus Santoso, Su-Cheng Haw, Ziyad T. Abdul-Mehdi, "Ontology extraction from relational database: Concept hierarchy as background knowledge", Knowledge-Based Systems, vol.24, no.3, pp.457-464, 2011.

[23] Kārlis Čerāns, Guntars Būmans, "RDB2OWL: A RDB-to-RDF/OWL Mapping Specification Language", In Proceedings of the 2011 conference on Databases and Information Systems VI: Selected Papers from the Ninth International Baltic Conference (DB&IS 2010), pp.139-152, IOS Press, 2011.

[24] K. Munir, M. Odeh, R. McClatchey, "Ontology-driven relational query formulation using the semantic and assertional capabilities of OWL-DL", Knowledge-Based Systems, vol.35, no.0, pp.144-159, 2012.