

# An Approach for Mapping Relational Database into Ontology

Mohamed A.G. Hazber<sup>1</sup>, Ruixuan Li<sup>1</sup>, Yuxi Zhang<sup>2</sup>, Guandong Xu<sup>3</sup>

1. School of Computer Science and  
Technology  
Huazhong University of Science and  
Technology  
Wuhan, China  
moh\_hazbar@yahoo.co.uk,  
rxli@hust.edu.cn

2. Department of Electronics and  
Information Engineering  
Huazhong University of Science and  
Technology  
Wuhan, China  
624374923zyx@gmail.com

3. Faculty of Engineering & IT  
University of Technology Sydney  
Sydney, Australia  
guandong.xu@uts.edu.au

**Abstract**—Sharing and reusing the big data in relational databases in a semantic way have become a big challenge. In this paper, we propose a new approach to enable semantic web applications to access relational databases (RDBs) and their contents by semantic methods. Domain ontologies can be used to formulate RDB schema and data in order to simplify the mapping of the underlying data sources. Our method consists of two main phases: building ontology from an RDB schema and the generation of ontology instances from an RDB data automatically. In the first phase, we studied different cases of RDB schema to be mapped into ontology represented in RDF(S)-OWL, while in the second phase, the mapping rules are used to transform RDB data to ontological instances represented in RDF triples. Our approach is demonstrated with examples and validated by ontology validator.

**Keywords**—Relational database; Semantic web ontology; RDF(S); OWL; Mapping rule

## I. INTRODUCTION

The continuous explosion of RDF (resource description framework) data opens door for new innovations in big data and semantic web initiatives, which can be shared and reused through application, enterprise and community boundaries. The World Wide Web Consortium (W3C) has recommended several formats for representing web ontology, such as RDF [1] data model, which is a standard model for data interchange on the web, RDF Schema [2], provides a data-modeling vocabulary for RDF data, and web ontology language (OWL) [3] as a formal language for authoring ontologies.

The bulk of existing web content is stored in RDBs [4], which characterized by high quality of storing and querying data but lack the ability to describe the semantics of data. In order to utilize today's RDB to support web applications and transparently participate in the semantic web, their associated database schemas need to be converted into semantically equivalent ontologies. Mapping RDB to RDF/OWL is an

attractive field of research and many approaches have been proposed. Most of these approaches are simple, equivalent matching and neglecting the formal definition which may lead to ambiguous when applying several transformation rules [5-8]. Through these approaches there are still difficulties for domain expert to understand the meaning between these approaches, such as unclear generation approach, un-unified ontology language and other related problems. Since, the manual ontology construction is a complex, cumbersome, mistakable, time consuming, high cost process, and requires the supports of domain experts in knowledge acquisition, the main goal of our approach is to generate ontology automatically from RDB and covers all possible concepts of relational model to find their best transformation into the ontology model.

The major contributions of this paper are as follows. (1) We propose a new approach for direct mapping RDB to ontology automatically containing: (a) Twenty-five unambiguous and well defined sequential mapping rules, and (b) Two well-defined functions identifying binary relations and generating identifier RowId. (2) The transformation rules are designed in an obvious form, so that the rules can be extended to reverse ontology to relational tables. (3) We design architecture that provides a uniform semantics between ontology mapping and information integration.

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 shows our proposed architecture and the rules for mapping RDB (schema and data) into semantic web ontology (schema and instance). The comparison is presented in Section 4. Finally, Section 5 concludes this paper with the future work.

## II. RELATED WORK

Different researches have been established in this area to provide methods and tools that exposed or converted data in RDB as ontological data described in RDF. The W3C RDB2RDF Incubator Group [9] has formed a working group to create a standard for exposing RDB as RDF. Ontology has been extracted from RDB by two ways automatic [10, 11] and semi-automatic [6, 12, 13, 14]. An approach presented by [10]

---

This work is sponsored by National Natural Science Foundation of China under grants 61173170, 61300222, 61433006 and U1401258, and Innovation Fund of Huazhong University of Science and Technology under grants 2015TS069 and 2013QN120, and Science and Technology Support Program of Hubei Province under grant 2014BCH270.

ontologies have been extracted from RDB based on a reverse engineering method using SQL DDL as the RDB model and transformed it to RDFS ontology. Semi-automatic mapping integrated several sources of information, based on the use of ontologies. The limitation of these rules in transformation of data type properties haven't domain and range defined, the not-null constraint was translated to the number restriction, and the translation was not capable for expressing the primary keys. Moreover, in the case of SQL-DDL code did not show the minimal cardinality, to solve this problem experts need to add this cardinality after the ontology was built. Methods [12], [13] and [14] used semi-automatic mapping rules for learning and extracting a global OWL ontology from RDB. Some ambiguous situations can be raised in their approach when several rules were applied. The procedure in [12] had a disadvantage of losing information because only the schema structure of an RDB had been used, so actual data were not utilized. While [6] presented a method for automatic ontology building using the RDB resources to improve the efficiency, and named the ontology generation system OGSRD, they ignored some tables that express association data, which could not be counted in the concepts. [15] introduced an extensible Eclipse plug-in supported the RDB2RDF conversion process used R2RML. However, these approaches suffer from several problems including complex data integration scenario, ignored most referential constraints and table check constraints, semi-automatic and require much user interaction, some ambiguous situations can arise when applying several rules, and there are still difficulties for domain expert to understand the meaning between these approaches.

This paper proposes new rules for the direct mapping RDB (schema and data) to RDF(S)-OWL ontology automatically. Our rules approach and adopt the methods clearly, easy, cover all concepts of relational models, and no interference between transformed concepts. Many types of foreign keys, referential constraints, and table check constraints are considered.

### III. RULES FOR MAPPING RDB SCHEMA AND DATA INTO ONTOLOGY

In this section, we introduced our approach that provides direct mapping rules RDB to RDF(S)-OWL automatically. The mapping rules are divided in two parts: rules for mapping (transform) RDB schema and mapping RDB instances as shown in Fig. 1.

#### A. System architecture for mapping rules

Our proposed system architecture shown in Fig. 1 includes the following stages: (1) Metadata and data were extracted from RDB using JDBC driver engine in Java, (2) Outputs of stage 1 are used as input for mapping rules to transform RDB model to ontology model. During this stage, Jena and some functions are used to generate the output of this stage which is an OWL structure build on RDF(S), (3) The RDB data analyzed from stage 1 to rows includes data of tables to generate the triples of data model. Through this stage, Jena and sub function (generate identifier of RowId) were used to generate RDF triples, (4) The ontology was generated by collecting the output of stages 2 and 3, and (5) Ontology validator used to verify our generated ontology.

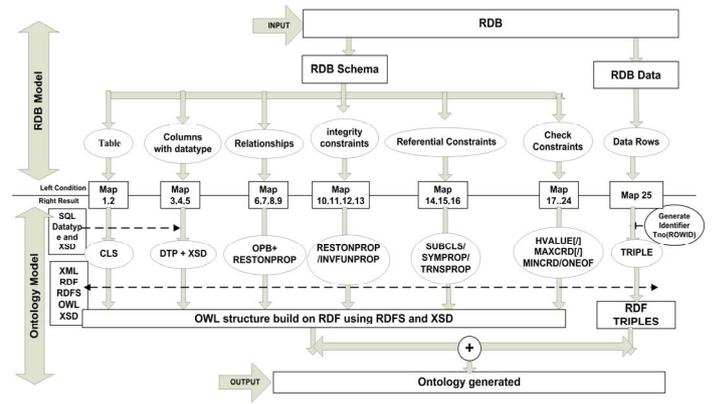


Fig. 1. System architecture for rules of mapping from RDB to Ontology

#### B. Rules for mapping RDB schema to ontology

This section defines the rules that mapping RDB schema to ontology built in OWL on top of RDF(S) vocabulary using XSD datatype. We applied each case rule by making an example of RDB (Fig. 2). Firstly, we identified the binary relation in order to transform RDB into an RDF triple with OWL vocabulary.

**Identifying binary relation IsBinRel(T1):** A table T1 is a binary relation between two tables (T2 and T3), if (1) T1 contains only two columns A1 and A2, which are a primary key PK of T1, (2) T2=(B1,...,Bn), where Bi is a primary key, and T3=(E1,...,En), where Ei is a primary key, assuming that  $i \in \{1..n\}$ , (3) A1 in table T1 is a foreign key FK, refers to column Bi in table T2, and (4) A2 in table T1 is a foreign key, refers to column Ei in table T3. In a mapping rule, negation is represented with the symbol !, and upper case letters are used to denote variables.

**Definition 1:** T1 is a binary relation between two tables (T2,T3):

$$T1(A1,A2) \wedge T2(B1,..,Bn) \wedge T3(E1,..,En) \wedge PKs(A1,A2,T1) \wedge [FK(A1,T1) \rightarrow ATTB(B1,T2) \wedge T1 \neq T2] \wedge [FK(A2,T1) \rightarrow ATTB(E1,T3) \wedge T1 \neq T3] \rightarrow \text{BinRel}(T1,A1,A2,T2,B1,T3,E1) \gg \text{IsBinRel}(T1).$$

The expression ATTB(B1,T2) indicates that B1 is one of attributes of T2. Expression of [FK(A1,T1)→ATTB(B1,T2)] indicates that A1 is the column of a foreign key in table T1 that points to table T2 through its column B1. Mapping process is done progressively in individual ways to include map number, input (relational case concept with its properties, and case condition to avoid interference and ambiguous situation), and output (ontology vocabulary with its properties) as shown in Table I.

Fig. 3 shows the generated ontology by applying above mapping rules on RDB example (Fig. 2). The oval shapes represented basic concepts in ontology, solid line rectangles represented xml schema data type, and the other shapes that describe the rest of the vocabulary ontology are represented in the left side of the figure.

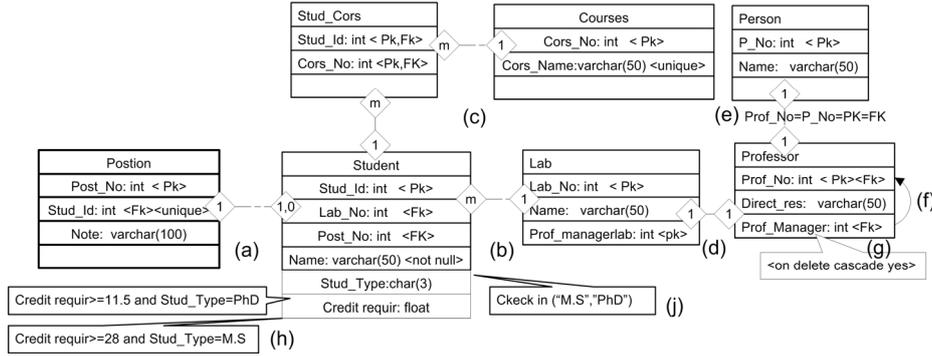


Fig. 2. Relationships and constraints in laboratory of a relational database (RDBLAB)

TABLE I. RULES OF MAPPING RDB SCHEMA TO ONTOLOGY

Map	RDB Concept	Case Condition	Ontology Component
1	Table (T)	$\neg \text{IsBinRel}(T)$	Owl:Class(T), rdfs:comment, rdfs:label
2	Table (T)	$\text{IsBinRel}(T)$ , $T(A1, A2, A3)$ , PKs(A1, A2), simple column(A3)	Owl:Class(T), rdfs:comment, rdfs:label
3	Datatype column $T(A_{type})$	$\neg(\text{IsBinRel}(T) \vee \text{FK}(A, T))$	xsd datatype such as xsd:int, byte, string, token, long, Boolean, float, double, data, time, anyURL (Table II)
4	Simple Column T(A)	$\neg(\text{PK}(A, T) \vee \text{FK}(A, T))$	Owl:DatatypeProperty(A), rdfs:label, rdfs:comment, rdfs:domain(T), rdfs:range(xsd)
5	Composite column	$T(A1(a1, a2), A2, \dots)$	owl:DatatypeProperty(A1), owl:FunctionalProperty(a1), rdfs:domain(T), rdfs:range(xsd), rdfs:subPropertyOf(A1)
6	1:1(0), FK $\notin$ PK (Fig. 2a)	$T1(A1..An) \neq T2(B1..Bn), \text{FK}(A1, T1), \text{PK}(B1, T2), A1 \neq \text{null}, \neg(\text{IsBinRel}(T1) \vee \text{IsBinRel}(T2))$	ObjectProperty(A1, T1 <sub>domain</sub> , T2 <sub>range</sub> ), RestrictionProp(A1, hasValue, T2), RestrictionProp(A1, minCard, xsd <sup>int</sup> 1) If the Rel 1:0 delete the last RestrictionProp of minCard
7	1:1, FK $\in$ PK (Fig. 2d)	Cond(map6), PK(A1, T1), FK(A1, T1)	ObjectProperty(A1, T1 <sub>domain</sub> , T2 <sub>range</sub> ), RestrictionProp(A1, hasValue, T2), RestrictionProp(A1, Card, xsd <sup>int</sup> 1)
8	1:m (Fig. 2b)	$T1(A1..An) \neq T2(B1..Bn), \text{FK}(A1, T1), \text{PK}(B1, T2), \text{valueOf}(T1, A1, \text{From}, T2, B1), A1 \neq \text{null}, \neg(\text{IsBinRel}(T1) \vee \text{IsBinRel}(T2))$	ObjectProperty(A1, T1 <sub>domain</sub> , T2 <sub>range</sub> ), RestrictionProp(A1, allValueFrom, T2), RestrictionProp(A1, minCard, xsd <sup>int</sup> 1)
9	m:n (Fig. 2c)	$T1(A1, A2), T2(B1..Bn), T3(C1..Cn), \text{IsBinRel}(T1) \rightarrow \text{BinRel}(T1, A1, A2, T2, B1, T3, C1)$	OBP(B1, T2 <sub>domain</sub> , T3 <sub>range</sub> ), RestrictionProp(B1, allValueFrom, T3), RestrictionProp(B1, minCard, xsd <sup>int</sup> 1), OBP(C1, T3 <sub>domain</sub> , T2 <sub>range</sub> ), RestrictionProp(C1, allValueFrom, T2), RestrictionProp(C1, minCard, xsd <sup>int</sup> 1)
10	Not null	Col(A <sub>notnull</sub> , T1)	Owl:Class(T1), RestrictionProp(A, minCard, xsd <sup>int</sup> 1),
11	Column unique constraint	Col(A <sub>unique</sub> , T1), !FK(A, T)	Owl:InverseFunctionProperty(A1), rdfs:domain(T1), rdfs:range(type of Col(A) as xsd type)
12	Column unique and FK	Col(A <sub>unique</sub> , T1), FK(A, T1, B, T2)	Owl:InverseFunctionProperty(A), rdfs:domain(T1), rdfs:range(T2)
13	PK constraint	PK(A, T)	Call (Map 10(A, T), iff(!FK(A, T)), Call (Map 11(A, T)), Call (Map 12(A, T)))
14	Referential constraints when FK=PK (Fig. 2e)	SuperTable(T1), SubTable(T2)	Owl:Class(T1), owl:Class(T2) < rdfs:subClassOf(T1) >
15	Column FK as recursive relationship (Fig. 2f)	$\neg \text{IsBinRel}(T)$ , PK(Col, T), FK(Col_Ref, T, Col, T), Vaue(Col_Ref, from, Col, T)	Owl:Class(T), owl:SymmetricProperty(Col_Ref, T <sub>domain</sub> , T <sub>range</sub> )
16	FK with cascade delete constraint (Fig. 2g)	ParentTable(Col, T1), ChildTable(Col_Ref, T2), $\neg(\text{IsBinRel}(T1) \vee \text{IsBinRel}(T2))$	Owl:Class(T1), owl:Class(T2), owl:TransitiveProperty(Col_Ref, T2 <sub>domain</sub> , T1 <sub>range</sub> )
17	Check with =	Ck(T, Col, V, =, Vck)	Hasvalue(T, Col, V, Vck)
18	Check >	Ck(T, Col, V, >, Vckmix)	maxCard(T, Col, V, Vckmix)
19	Check $\geq$ (Fig. 2h)	Ck(T, Col, V, $\geq$ , Vckmix)	Hasvalue(T, Col, V, Vck), maxCard(T, Col, V, Vckmix)
20	Check <	Ck(T, Col, V, <, Vckmin)	minCard(T, Col, V, Vckmin)
21	Check $\leq$	Ck(T, Col, V, $\leq$ , Vckmin)	Hasvalue(T, Col, V, Vck) and minCard T, Col, V, Vckmin)
22	Check !null	Ck(T, Col, V, Vck!null)	minCard T, Col, V, Vckmin=1)
23	Check Between	Ck(T, Col, V, BT, Vckmin, Vckmix)	minCard T, Col, V, Vckmin), maxCard(T, Col, V, Vckmix)
24	Check In (Fig. 2j)	Ck(T, Col, V, In, {Vck1, ..., Vckn})	OnOf(T, Col, V, list {V1, ..., Vn})

TABLE II. XSD DATA TYPE USING FOR MAPPING FROM SQL DATA TYPE

Type	RDB Data Type	Ontology data type as XSD
<i>Byte</i>	BIT VARYING, Tinyint	xsd:Byte,xsd:unsignedByte
<i>Logical</i>	BIT, Boolean	xsd:Boolean
<i>Char</i>	CHAR,VCHAR, VARCHAR,NCHAR, LONGTEXT,Memo Text NVARCHAR, BOLOB, TINBLOB,TINTEXT, MEDIUMTEXT ,MEDIUMBLOB,SET('V1',..., 'Vn'), ENUM('V1',..., 'Vn')	xsd:String,xsd:normalizedString, xsd:token
<i>Numeric, Currency</i>	INTEGER, LONG	xsd:Integer,xsd:positiveInteger,xsd:negativeInteger,xsd:nonPositiveInteger,xsd:nonNegativeInteger ,xsd:unsignedInt,xsd:unsignedLong, xsd:int, xsd:long
	SMALLINT ,TINYINT, MEDIUMINT	xsd:Short, xsd:unsignedShort, xsd:int
	FLOAT,REAL	xsd:Float
	INTERVAL	xsd:Duration
	NUMERIC,DECIMAL,MONY	xsd:Decimal
	DOUBLE PRECISION	xsd:Double
<i>Date/DateTime</i>	DATE, TIMESTAMP,TIMESTAMP WITH TIME, TIME,TIME WITH TIME ZONE	xsd:Date , xsd:Datetime, xsd:Time
<i>Part of datetime</i>		xsd:gYear,xsd:gMonth,xsd:gDay,xsd:gYearMonth, xsd:gMonthDay, xsd:duration
<i>XML</i>	XML	xsd:anyType
<i>Memo(Binary) or Character(Binary)</i>	Binary, Varbinary, BLOB,IMAGE,LONGBLOB, MEDIUMBLOB,TINYBLOB	xsd:hexBinary, xsd:base64Binary
<i>Link(URL)</i>	Hyperlink to URI	xsd:anyURI

### C. Rules for mapping RDB instances into ontology

The goal of this task is the extraction of ontological instances based on the rows of the RDB tables.

**Map 25:** If a table T is mapped to the class then all rows of the table are transformed to the instance of a class. Each column in table T can be transformed to the data properties of the instance. A **Tno** is identifier of rows (RowId) that generated according to the following rule.

Generate identifier **Tno**:

**RowId(Tno,T,Row,Col1,...,Coln,V1,V2,...,Vn)**

← PKn(T,Col1,Col2,...,Coln),

Value(T,Rw,Col1,V1),

Value(T,Rw,Col2,V2),...,Value(T,Rw,Coln,Vn)

,**Collect**(T,"\_",V1,V2,...,Vn,**Tno**).

RowId(Tno,T,Row,Col1,...,Coln,V1,V2,...,Vn) generated the identifier Tno of a row. In fact, PK1("Student","Stud\_Id") and Value ("Student",Tno1,"Stud\_Id",1) were satisfied in our example. The (Tno1=Student\_1) is the identifier for the tuple in table Student with value 1 in the primary key. For instance, assume the rows of the table student are (1,'mohamed') and (2,'shadi') after applied Map 25 with generated identifier Tno using RowId function the generated triples are follows:

### Triple 1:

<Student\_1, type, Student>, <Student\_1, Stud\_Id, 1>, <Student\_1, Name, 'mohamed'>

### Triple 2:

<Student\_2, type, Student>, <Student\_2, Stud\_Id, 2>, <Student\_2, Name, 'shadi'>

### D. Example

This section provides an example on how to represent the schema and data of RDBLAB (Fig. 2) to ontology using our approach. For instances, the tuples of table Student (1, 'Mohamed', 501,), (2, 'Shadi'); the tuples of table Lab (501, 'SW', 1), (502, 'IR', 2); the tuples of table Professor (1, 'Semantic Web Ontology'), (2, 'Big Data', 1); the tuples of table Person (1, 'Ahmed'), (2, 'Han' ); the tuples of table Stud\_Cors (1, 1), (2, 1); the tuples of table Courses (1, 'Information Retrieval'), (2, 'Semantic Web'). Therefore, the ontology generated shown in Fig. 4. The Fig. 4a represents the ontology schema generated from RDB schema while the Fig. 4b represents the RDF triples and graph generated from RDB data.

The RDF validator [16] used to validate RDF generated code after applied our rules.



From the instances above we know that the values of column Lab\_No in table Student not simply represented as plain literal instead of object properties added to the class Student. This object properties link the resources between nodes of Student and the nodes of Lab. Likewise the table Stud\_Cors database is not mapped as a class instead of two inverse object properties. But the Stud\_Cors should be mapped as a class if the table Stud\_Cors has another column except two foreign keys.

#### IV. IMPLEMENTATION AND COMPARISON

In order to implement the transformation from an RDB Schema and data to ontology, we propose a method using Apache Jena in Java language. Apache Jena™ is a Java framework for building semantic web applications. Jena provides a collection of tools and Java libraries to develop semantic web and linked-data apps, tools and servers [17]. To examine it, the new method should be applied to an RDB. A sample RDB named, RDBLAB (have several cases applied by our rules), created by MYSQL5.5 and connecting with Java JDBC by “com.mysql.jdbc.Driver”.

A comparison of existing methods with the proposed method is shown in Table III. Table III shows several aspects of approaches compared with our approach.

TABLE III. A COMPARISON BETWEEN OUR PROPOSED APPROACH AND OTHER EXISTING APPROACHES

Methods	Model	Relationship	Check constraint	Data transform	Generated RowId	Strategy used for solution	Validation
Astrova	Auto	1:1	No	simple	No	Examples	No
Li et al.	Semi-auto	M:N	No	No	No	Expository examples	No
Shen et al. Hu et al.	Semi-auto	M:N	No	simple	No	Examples	No
Zhang et al.	Semi-auto	1:1	No	simple	No	OGSRD	No
Proposed approach	Auto	1:1,1:0, 1:m,m:m, inheritance, recursive	All check constraint types with logical operation	Yes	Yes	mapping rules, Examples and prototype framework	Yes

#### V. CONCLUSION AND FUTURE WORK

By adopting our approach, domain-related experts can be directed-automatically to engage in mapping of different sources of RDB Schema and data to RDF(S)-OWL ontologies.

The main advantages of this approach are adopting the methods clearly, easily, and closely to the software programmers. Moreover, we studied different cases of an RDB, which decreased the loss of information and avoided ambiguity where rules are applied. For the future work, we confess there is a domain for extending this work by extracting new mapping rules and querying RDBs on the semantic web using an ontology generated by our rules.

#### REFERENCES

- [1] R. Cyganiak, D. Wood, and M. Lanthaler. (2014). RDF 1.1 concepts and abstract syntax. Available: <http://www.w3.org/TR/rdf11-concepts/>
- [2] D. Brickley, R. Guha, and B. McBride. (2004). RDF Vocabulary Description Language 1.0: RDF Schema. Available: <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>
- [3] B. Motik, B. C. Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. (2009). OWL 2 Web Ontology Language: Profiles. Available: <http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/>
- [4] K. C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang, "Structured databases on the web: Observations and implications," ACM SIGMOD Record, vol. 33, pp. 61-70, 2004.
- [5] W. Hu and Y. Qu, "Discovering simple mappings between relational database schemas and ontologies," in ISWC'07/ASWC'07 Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference, Busan, Korea, 2007, pp. 225-238.
- [6] L. Zhang and J. Li, "Automatic Generation of Ontology Based on Database," Journal of Computational Information Systems, vol. 7, pp. 1148-1154, 2011.
- [7] M. Arenas, A. Bertails, E. Prud, and J. Sequeda. (2012). A Direct Mapping of Relational Data to RDF. Available: <http://www.w3.org/TR/rdb-direct-mapping/>
- [8] H. Mohamed, Y. Jincai, and J. Qian, "Towards Integration Rules of Mapping from Relational Databases to Semantic Web Ontology," in Web Information Systems and Mining (WISM), 2010 International Conference on, Sanya, China, 2010, pp. 335-339.
- [9] A. Bertails and E. G. Prud'hommeaux, "Interpreting relational databases in the RDF domain," in Proceedings of the sixth international conference on Knowledge capture, Banff, AB, Canada, 2011, pp. 129-136.
- [10] I. Astrova, "Reverse engineering of relational databases to ontologies," in Proceeding of 1st European Semantic Web Symposium(ESWS), Heraklion, Grete, Greece, 2004, pp. 327-341.
- [11] S. Zhou, G. Meng, H. Ling, and H. Zhang, "Tool for Translating Relational Databases Schema into Ontology for Semantic Web," in Education Technology and Computer Science (ETCS), 2010 Second International Workshop on, Wuhan, 2010, pp. 198-201.
- [12] M. Li, X.-Y. Du, and S. Wang, "Learning ontology from relational database," in Proceedings of 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 2005, pp. 3410-3415.
- [13] G. Shen, Z. Huang, X. Zhu, and X. Zhao, "Research on the Rules of Mapping from Relational Model to OWL," in Proceedings of the OWLED\*06 Workshop on OWL: Experiences and Directions, Athens, Georgia-USA, 2006.
- [14] C. Hu, H. Li, X. Zhang, and C. Zhao, "Research and Implementation of Domain-Specific Ontology Building from Relational Database," in ChinaGrid Annual Conference, 2008. ChinaGrid'08. The Third, Dunhuang, Gansu, 2008, pp. 289-293.
- [15] E. Marx, P. Salas, K. Breitman, J. Viterbo, and M. A. Casanova, "RDB2RDF: A relational to RDF plug-in for Eclipse," Software: Practice and Experience, vol. 43, pp. 435-447, 2013.
- [16] E. Prud'hommeaux and R. Lee. (2004). W3C RDF validation service. Available: <http://www.w3.org/RDF/Validator/>
- [17] Aapch Jena. (2015). A free and open source Java framework for building Semantic Web and Linked Data applications. Available: <http://jena.apache.org/>