Publishing RDF from Relational Database Based on D2R Improvement

YING CHEN[1, 2], XIAOMING ZHAO[1, 2], SHIQING ZHANG[2]
[1]Department of Computer Science; [2]Institute of Image Processing and Pattern Recognition
Taizhou University
No.605 Dongfang Avenue, Linhai, Taizou, Zhejiang
CHINA
ychen222@tzc.edu.cn

Abstract: - As a key technology to implement Semantic Web, linked data have gradually been an academic and industrial concern. Linked data represents a practice of technologies on the web and linked structure data. The goal of linked data is to enable people to share structured data on the web as easily as they can share documents today. On the Web of data structured with linked data, users can jump from one dataset to another. Compared with the Web of document which enables one to jump from one document to another, data Web provides much more links and has semantics. Mapping relational databases to RDF is a fundamental problem for the development of linked data. This paper firstly makes an improvement with R2RML to open source software named D2R which can publish linked data, and then takes an example of the data concerning a college enrollment in Sichuan of China in 2011, demonstrating that we can publish the data in RDB into RDF as linked data through the new approach more efficient and make it browsable.

Key-Words: - Semantic web; relational database; RDF; D2R; R2RML; linked data

1 Introduction

The term Linked Data was coined by Tim Berners-Lee in his Linked Data [1] Web architecture note. The term refers to a style of publishing and interlinking structured data on the Web. The basic assumption behind Linked Data is that the value and usefulness of data increases the more it is interlinked with other data.

Chris Bizer and other practitioners submitted an application for Linked Open Data Project to W3C SWEO (Semantic Web Education and Outreach), with two page document [2] giving an overview about the Linking Open Data project. The goal of the W3C SWEO Linking Open Data community project [3] is to extend the Web with a data commons by publishing various open datasets as RDF (Resource Description Framework) on the Web and by setting RDF links between data items from different data sources.

Over the past few years, an increasing number of web sites have started to publish structured data on the Web according to the Linked Data principles. This trend has led to the extension of the Web with a global data space—the Web of Data [5].

The basic tenets of Linked Data are to:

(i) Use the RDF data model to publish structured data on the Web.

(ii) Use RDF links to interlink data from different data sources.

Up to September 2011, 31,634,213,779 RDF Triples and 503,998,829 RDF (Out-) Links have been published. The data domain has been widely extended to 295 datasets, such as government data (13 billion triples), Geographic Data (6 billion triples), Life Sciences (3 billion triples), Publications and Media (4.6 billion triples), and User Generate Content [4].

With the advent of Enterprise 2.0, not only the Public Web, but also the enterprises have an urgent need to link its data to data on the Web in order to have it widely applied. However, the current data, especially the data from the enterprises, most of which exist in relational database. So mapping relational databases to RDF is a fundamental problem.

D2R (Database to RDF) Server [6-7] is a tool which has been widely used for publishing relational databases on the Semantic Web. It enables RDF and HTML browsers to navigate the content of the database, and allows querying the database using the SPARQL (SPARQL Protocol and RDF Query Language) query language. However a certain amount of inference (datatypes, basic RDFS, owl:sameAs) would be desirable and feasible with the current architecture [8].

R2RML (Relational databases to RDF Mapping Language) is a W3C recommended language [9] for expressing customized mappings from RDB (Relational Databases) to RDF datasets. R2RML enables different types of mapping implementations and then generates RDF dumps.

This paper gives an introduction on how to make an improvement on D2R with R2RML on extending an existing popular framework for mapping relational data to RDF, proposes algorithm designs and implementation technology of mapping relational databases to RDF mapping file based on R2RML, while providing an RDF view that is always consistent with the underlying database to make it browsable.

2 Related work

2.1 Linked data

Wikipedia defines linked data as “a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs (Uniform Resource Identifier) and RDF.”

To put it simply, Linked data, is the best practice to use Web and create semantic interlinks between different data sources. Different sources can come from different system within an organization or different system from different organizations. They are probably linked to each other even though they are completely different in terms of contents, storage locations and storage modes. For example, Books on Amazon could be related with people on MySpace, because the author of the book might have registered on MySpace. In short, the biggest feature of Linked data is that it can interlink different data.

These features were introduced by Tim Berners-Lee in his Web architecture note Linked Data [7] and have become known as the Linked Data principles. These principles are the following:

(i) Resource. Before publishing data of a certain field, we must make sure what we are going to publish [10]. Anything could be named as resource, as long as it is of great use and has the great chance to be cited.

(ii) Resource identification. Any resource would be identified with HTTP URI (Uniform Resource Identifiers), with the purpose that the data could be accessed through HTTP protocol to make it accessible and interlinked based on Web.

(iii) Resource description. There are many kinds of descriptions about resource, such as HTML, XML, RDF and JPEG. The file on file Web is in the form of HTML, and data on the data Web is in the form of RDF. RDF describes the resource as a triple, subject, predicate and object.

For example:

Miss Yang teaches English.

(Subject) (Predicate) (Object)

The subject is to describe the resource. The predicate indicates certain property of the subject, such as name and date of birth, or certain relations between individuals, such as employment, acquaintances, professor, etc. The object represents the value of the property or relation. Both the Subject and the Predicate need to be represented with HTTP URI. The object can use HTTP URI to identify another resource, or use character string to represent text. The subject can be considered as class resource, the predicate as property resource of class resource, and the object class resource or text resource. The triple can be divided into two categories: text triple and non-text triple, and the latter can be regarded as the interlinking between class resources.

Linked data have many advantages, but in order to play their key roles, they have to meet the fundamental principles of Linked data. However, in practice, a great number of existing data do not meet these principles, so the practioners of linked data have developed a range of practical tools to help complete the converting from traditional data to Linked data.

2.2 D2R

D2R is one of the popular tools, which can publish the relational database into Linked data [11-14]. D2R mainly consists of D2R Serve, D2R Engine and D2R Mapping. D2R Serve is a HTTP Serve, which provides access interface for RDF data query and more options for top-level RDF browser, SPARQL query client and traditional HTML browser.

D2RQ Engine [15] can use D2RQ Mapping to convert the data in relational database into RDF format. D2RQ Engine does not convert the relational database into real RDF data, but use

D2RQ Mapping file to map it into virtual RDF format. When users visit the relational database, the file can convert query language SPRQL of RDF data into query language SQL of RDB data, and then convert the query result of SQL into RDF triple or SPARQL query result. D2RQ Engine is built on the interface of Jena, which is a java platform to create Semantic Web application and provides programming environment based on RDF and SPARQL. D2RQ Mapping is mainly used to define the Mapping rules which convert the relational database into RDF format.

Figure 1 illustrates the server architecture of D2R [6].

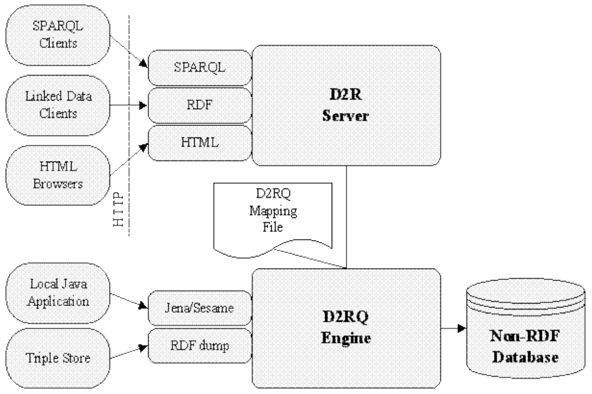


Fig.1 D2R Server Architecture diagram

The Architecture consists of:

(i) D2R Server. An HTTP server that provides a Linked Data view, a HTML view for debugging and a SPARQL Protocol endpoint over the database.

(ii) D2RQ Engine. A plug-in for the Jena Semantic Web toolkit, which uses the mappings to rewrite Jena API calls to SQL queries against the database and passes query results up to the higher layers of the frameworks.

2.3 Other Approaches

The vast majority of the structured data of our age is stored in relational databases. In order to link and integrate this data on the Web, it is of paramount importance to make relational data available according to the RDF data model and associated serializations [16]. Various tools and projects have been launched aiming at facilitating the lifting of mapping RDB data to RDF to reach semantically structured and interlinked data.

We can differentiate between tools that either expose their data as RDF, or expose a SPARQL endpoint for interactive querying of the data. Some typically examples are listed as below:

Triplify [17] is based on mapping HTTP - URI requests onto relational database queries expressed in SQL with some additions. Triplify provides small, light-weight plugins for database-backed Web applications and transforms the resulting relations into RDF statements, publishes the data on the Web in various RDF serialization, in particular as Linked Data.

OpenLink's Virtuoso RDF Views [18] allows the mapping of relational data into RDF. RDF Views are integrated into the Virtuoso query execution engine, consequently allowing SPARQL query over native RDF and relational data.

Ultrawrap [19] is concerned with wrapping relational databases and using the existing SQL system to execute SPARQL queries on the relationally stored data without replicating the data to triplestores.

SparqlMap [20] is a SPARQL-to-SQL rewriter with the rationale of enabling SPARQL querying on existing relational databases by rewriting a SPARQL query to exactly one corresponding SQL query based on mapping definitions expressed in R2RML.

Sparqlify [21] is a SPARQL-SQL rewriter that enables one to define RDF views on relational databases and query them with SPARQL. It is currently in alpha state and powers the Linked-Data Interface of the LinkedGeoData Server.

Revelytix Spyder [22] is a software tool used to expose relational data stores as if they were RDF or a SPARL 1.1 end point. No data is extracted and duplicated in a triple store. A Spyder exposes a relational data store as if it were stored as RDF, making that data available to any application issuing a SPARQL query.

From our point of view, a major reason for the lack of deployment of these tools and approaches lies in the complexity of generating mappings. While R2RML which can greatly enhance the efficiency of mapping is just fit the issue.

So we propose an approach that improves D2R with R2RML for publishing RDF from Relational database.

3 D2R improvement with R2RML

By using D2R, the relational database can be converted and accessed in two ways:

(i) Method I. Visiting the data in relational database in two steps after they have been converted into virtual RDF data. The first step is to generate

Mapping file, and the second step is to use Mapping, converting the relational database to be accessed. Users can visit the relational database through D2R serve or use API of Jena/ Seasame in Java application to access the data.

(ii) Method II. The second method is to convert the data in relational database into real RDF file to make it accessible to RDF Store.

In general, the first method is more widely adopted.

The scale of data in database is large and data are not stable, so when they are converted into virtual RDF data, we can generate mapping file with an R2RML processor which has potential to integrate numbers of heterogeneous databases and makes the space of the data being less complicated and easy to update rather than D2RQ Engine.

3.1 R2RML processor architecture

We improve the "D2RQ Engine" in Figure 1 be instead of an "R2RML processor". The input of R2RML processor including R2RML mapping file and SQL connection of relevant database, while the output is N3 formatted mapping file.

The architecture of our R2RML processor shows in Figure 2.

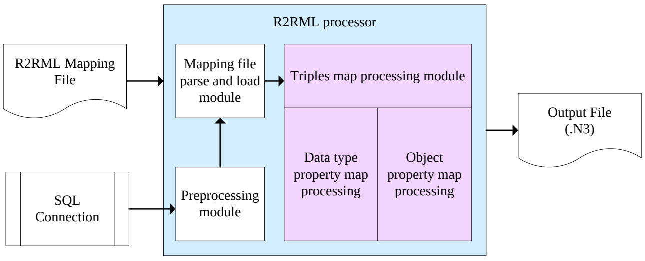

R2RML processor
R2RML Mapping File
Mapping file parse and load module
Triples map processing module
Output File (.N3)
Data type property map processing
Object property map processing
SQL Connection
Preprocessing module

Fig.2 Architecture of R2RML processor prototype

Functional specifications of the prototype show as below:

(i) Preprocessing module. By a given database connection parameters, Preprocessing module acquires the SQL Connection;

(ii) Mapping file parse and load module. Mapping file parse and load module obtains the legal R2RML mapping file, and loads the mapping file into the Jena model, acquires all triples map elements in the model;

(iii) Triples map processing module. Triples map processing module includes data type property mapping processing and object property mapping processing, accomplishes the processing of all triples map elements in order to get RDF triples data of the mapping, outputs the RDF triples data as N3 format file.

3.2 Algorithm implementation

Basic thinking of the algorithm can be simply described as:

(i) To parse the mapping file and obtain all map elements;

(ii) To analysis each map element, obtain the correspondence relationship between sub-element and relational table with its fields;

(iii) To acquire tuples of logic table by given database connection, map field value of tuples to RDF Term in accordance with the correspondence relationship, and combines as RDF triples.

The mapping algorithm has the following steps:

(i) By configuring the parameters to acquire the SQL connection for the RDB, load the R2RML mapping file (MF);

(ii) To parse the R2RML mapping file MF, obtain all triples map elements TripleMap;

(iii) To parse all TripleMap elements, acquire tuples of logic table by given database connection SQL connection;

(iv) Map processing in turn according to type of ObjectMap in PredicateObjectMap, map tuples in logic table in turn to a set of RDF triples data triple; add the data triple to the output columns List;

(v) Output all elements in List to output file OUT.

We provide the Algorithm as follow:

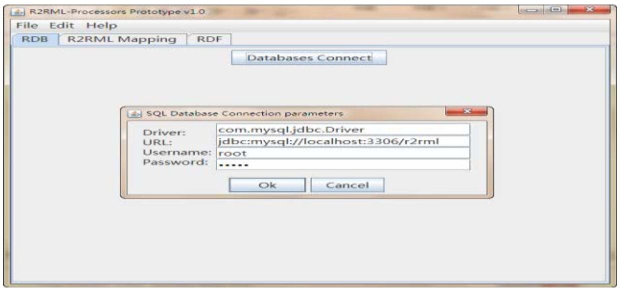
```
Algorithm. SQLconnection2N3
Input: database SQL connection, R2RML mapping file mf
Output: output file OUT
Steps:
1. List $\leftarrow \varnothing$ ;
2. model $\leftarrow$ loading and parsing R2RML mapping file mf to Jena model;
3. tripleMaps $\leftarrow$ get TripleMap from the model;
4. FOR each tripleMap in tripleMaps DO
5. List $\leftarrow$ doTripleMap of each tripleMap (SQL connection, tripleMap);
6. END FOR;
7. write elements in List to output file OUT;

For Algorithm. doTripleMap
Input: SQL connection, tripleMap
Output: List
```

```
Steps:
1. List ← ∅ ;
2. logicalTable ← get LogicalTable from tripleMap;
3. subjectMap ← get SubjectMap from tripleMap;
4. rows ← SQL query of the logicalTable using SQL connection;
5. FOR each row in rows DO
    6. subject ← subject(row, subjectMap);
    7. graphsInSMap ← graph (row, subjectMap);
    8. classes ← classes(row, subjectMap);
    9. FOR each class in classes DO
        10. triple ← (subject, rdf:type, class);
        11. IF (graphs = ∅ ) add(triple, defaultGraph) to List;
        12. ELSE add (triple, graphsInSMap) to List;
    13. END FOR;
    14. predicateObjectMaps ← get PredicateObjectMap from tripleMap;
    15. FOR each predicateObjectMap in predicateObjectMaps DO
        16. List ← doPMapNR of each predicateObjectMap (subject, graphsInSMap, predicateObjectMap, row);
    17. END FOR;
18. END FOR;
19. For each referencing PredicateObjectMap in predicateObjectMaps DO
    20. List ← doPMapWR of each referencing PredicateObjectMap (SQL connection, predicateObjectMap, subjectMap, tripleMap);
21. END FOR;
22. Return List;
```

3.3 Generate mapping file

According to our Algorithm, we develop an R2RML Processor Prototype based on J2SE (Java 2 Platform Standard Edition); select an example of the admission information of Sichuan province in 2011 from the frequently used database to describe the operation.

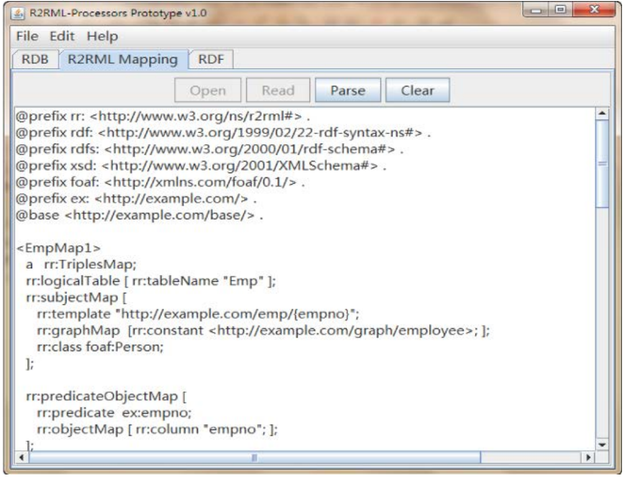
R2RML Processor Prototype has 3 labels: RDB, R2RML Mapping and RDF, each of the labels has its interface and achieves the corresponding functions. Interface of label RDB contains a button "Databases Connect", by clicking which can pop up a database connection parameters configuration window includes Drive, URL, Username and Password, to complete preprocessing module functions. See Figure 3.


R2RML-Processors Prototype v1.0
File Edit Help
RDB
R2RML Mapping
RDF
Databases Connect
SQL Database Connection parameters
Ok
Cancel

Driver: com.mysql.jdbc.Driver
URL: jdbc:mysql://localhost:3306/r2rml
Username: root
Password: •••••

Fig.3 Configure the database connection parameters

Interface of label R2RML Mapping contains 4 buttons. By clicking Open to choose corresponding R2RML mapping file, clicking Read to read the mapping file and display in the text area(See Figure 4), and clicking Parse to process mapping and generate N3 file to output. Here we save the N3 file as "cy2.n3".



```
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix ex: <http://example.com/> .
@base <http://example.com/base/> .

<EmpMap1>
  a  rr:TriplesMap;
  rr:logicalTable [ rr:tableName "Emp" ];
  rr:subjectMap [
    rr:template "http://example.com/emp/{empno}";
    rr:graphMap  [rr:constant <http://example.com/graph/employee>; ];
    rr:class foaf:Person;
  ];

  rr:predicateObjectMap [
    rr:predicate  ex:empno;
    rr:objectMap [ rr:column "empno"; ];
  ];
```

Fig.4 Load R2RML mapping file and display

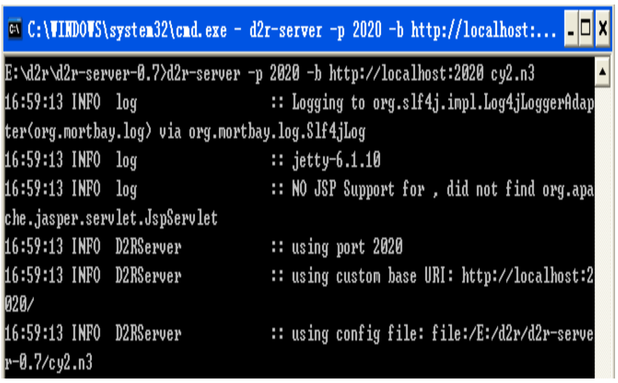
4 D2R improvement based mapping from RDB to RDF

After the Mapping file has been generated, we hope to convert and access the data in the relational database through mapping file. We use the data processing and accessing method defaulted by D2R server to browse Linked data.

This method is simple. We do not have to write code and what we do is to enter into the file path where D2R is under the command line and execute the following command to run D2R Server:

d2r-server [-p port] [-b serverBaseURI] [--fast] mappingFileName

"mappingFilName" uses cy2.n3, the Mapping file generated in the previous part. See Figure 5.



```
C:\WINDOWS\system32\cmd.exe - d2r-server -p 2020 -b http://localhost:...
E:\d2r\d2r-server-0.7>d2r-server -p 2020 -b http://localhost:2020 cy2.n3
16:59:13 INFO  log              :: Logging to org.slf4j.impl.Log4jLoggerAdap
ter(org.mortbay.log) via org.mortbay.log.Slf4jLog
16:59:13 INFO  log              :: jetty-6.1.10
16:59:13 INFO  log              :: NO JSP Support for , did not find org.apa
che.jasper.servlet.JspServlet
16:59:13 INFO  D2RServer        :: using port 2020
16:59:13 INFO  D2RServer        :: using custom base URI: http://localhost:2
020/
16:59:13 INFO  D2RServer        :: using config file: file:/E:/d2r/d2r-serve
r-0.7/cy2.n3
```

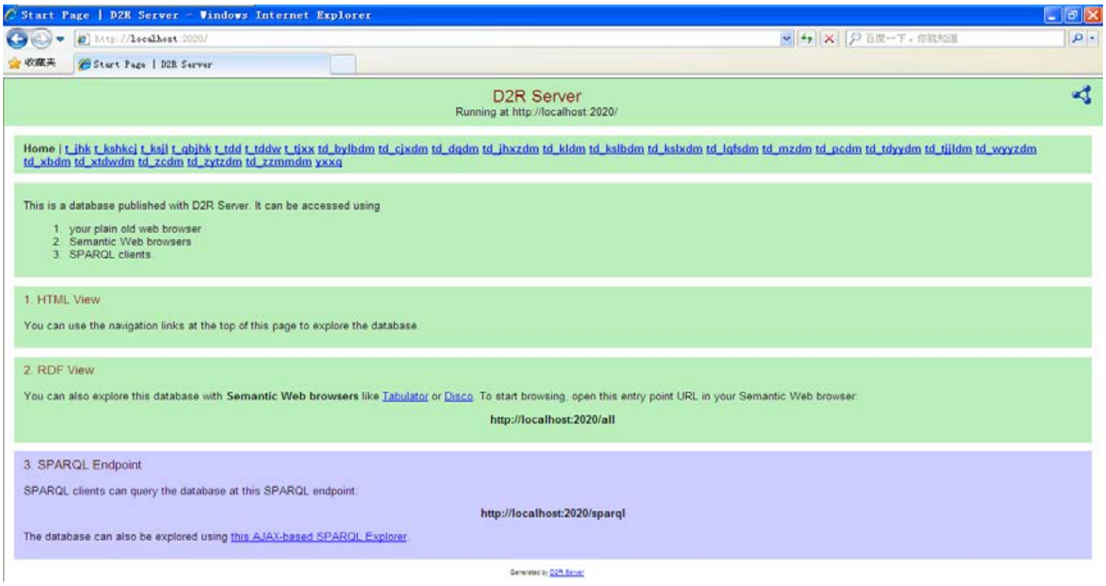
Fig.5 Using n3 mapping file

When we visit http://localhost:2020 in the Web browser, we can use default HTML browser, DF browser or SPARQL query end to access our data. See Figure 6.

We can select anyone of the 26 schema, such as “td_xbdm” in this paper, and we will get the following results. See Figure 7.

If we click “td_xbdm #”, we will get the detailed property and relation of “td_xbdm”. See figure 8.

When browsing the details of “td_xbdm#”, we can click vocab “td_xbdm” and find its source “db:td_xbdm”. See Figure 9.



Start Page | D2R Server - Windows Internet Explorer

http://localhost:2020/

D2R Server
Running at http://localhost:2020/

Home | t_jhk t_kshkcj t_ksjl t_qbjhk t_tdd t_tddw t_tjxx td_bylbdm td_cjxdm td_dqdm td_jhxzdm td_kldm td_kslbdm td_kslxdm td_lqfsdm td_mzdm td_pcdm td_tdyydm td_tjjldm td_wyyzdm td_xbdm td_xtdwdm td_zcdm td_zytzdm td_zzmmdm yxxq

This is a database published with D2R Server. It can be accessed using

1. your plain old web browser
2. Semantic Web browsers
3. SPARQL clients

1. HTML View

You can use the navigation links at the top of this page to explore the database.

2. RDF View

You can also explore this database with **Semantic Web browsers** like Tabulator or Disco. To start browsing, open this entry point URL in your Semantic Web browser:

http://localhost:2020/all

3. SPARQL Endpoint

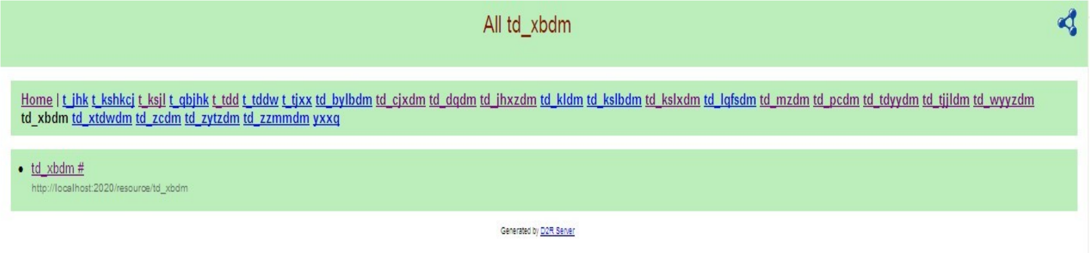
SPARQL clients can query the database at this SPARQL endpoint:

http://localhost:2020/sparql

The database can also be explored using this AJAX-based SPARQL Explorer

Generated by D2R Server

Fig.6 D2R Server’s run interface page



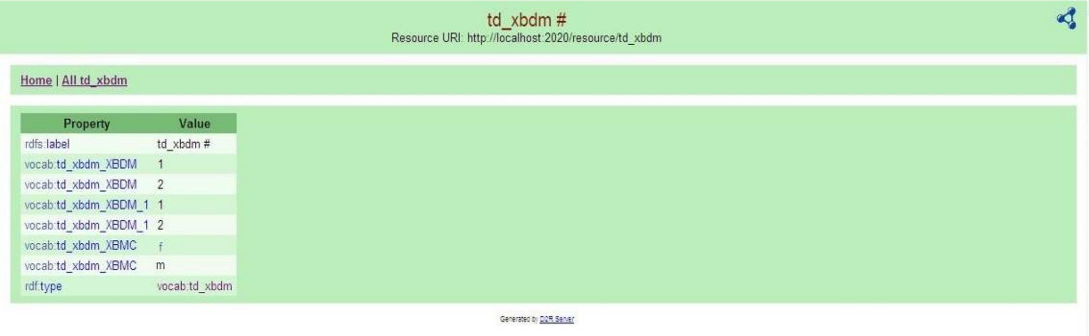
All td_xbdm

Home | t_jhk t_kshkcj t_ksjl t_qbjhk t_tdd t_tddw t_tjxx td_bylbdm td_cjxdm td_dqdm td_jhxzdm td_kldm td_kslbdm td_kslxdm td_lqfsdm td_mzdm td_pcdm td_tdyydm td_tjjldm td_wyyzdm td_xbdm td_xtdwdm td_zcdm td_zytzdm td_zzmmdm yxxq

- td_xbdm #
  http://localhost:2020/resource/td_xbdm

Generated by D2R Server

Fig.7 Result of clicking “td_xbdm” in HTML page



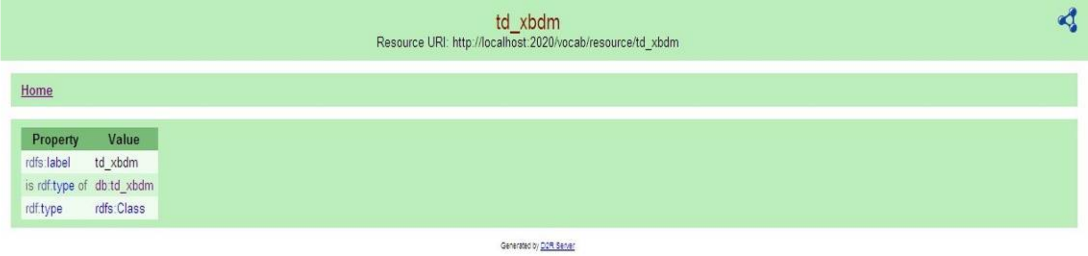
td_xbdm #
Resource URI: http://localhost:2020/resource/td_xbdm

Home | All td_xbdm

Property	Value
rdfs:label	td_xbdm #
vocab:td_xbdm_XBDM	1
vocab:td_xbdm_XBDM	2
vocab:td_xbdm_XBDM_1	1
vocab:td_xbdm_XBDM_1	2
vocab:td_xbdm_XBMC	f
vocab:td_xbdm_XBMC	m
rdf:type	vocab:td_xbdm

Generated by D2R Server

Fig.8 Result of clicking “td_xbdm #” in HTML page



td_xbdm
Resource URI: http://localhost:2020/vocab/resource/td_xbdm

Home

Property	Value
rdfs:label	td_xbdm
is rdf:type of	db:td_xbdm
rdf:type	rdfs:Class

Generated by D2R Server

Fig.9 Resource of td_xbdm

Likewise, we can visit the other 26 schema in the browser, or we can also explore this database with Semantic Web browsers like Tabulator or Disco.

5 Conclusion

The paper gives an introduction of how to publish the data of relational database into RDF as Linked data, with a focus on how to improve D2R with R2RML and emulate the D2RQ Server with some customizations. As discussed above, D2RQ has its own proprietary map-ping language and until this moment there is not any production version that supports R2RML. To fill this gap, we developed a customized version of improve D2R to support the main terms of R2RML for expressing customized mappings from relational databases to RDF datasets, translation procedures and the method to create association attribute. And it supports “Relational Views” which allows the publication of mappings when user chooses to generate the kind of Views.

The R2RML language can use different types of logic table, SQL base table/view or R2RML view, to specify RDB2RDF mapping. Intuitively, different types of logical table and different mappings defined on logical tables will have different effects on the performance of the mapping algorithm. Besides, if the users wish to connect their own data source to other source on the Web, their relations have to be built and storage the relations in real RDF database. When accessing RDF data, users not only have to query about the virtual RDF database converted from D2R, but also has to query the real RDF database to make sure all the internal and external relations have been queried.

After the creation of the R2RML mapping, probably the user needs to publish it to make some SPARQL queries over the RDF data. The limitation here is that D2RQ does not support “R2RML Views”, so future work should focus on this aspect and also to improve the algorithm in order to facilitate efficiency.

Acknowledgements:
This Project is supported by National Natural Science Foundation of China under Grant No.61272261 and No.61203257, Scientific Research Fund of Zhejiang Provincial Education Department under Grant No.Y201223744, Incubation Fund of Taizhou University under Grant No.PY2012015.

References:

[1] T. Berners-Lee, Linked data, 2006. *http://www.w3.org/DesignIssues/LinkedData.html (last modified on 18 June 2009)*

[2] L.O. Data, W3c sweo community project, *http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData (last modified on 13 May 2013)*

[3] C. Bizer, T. Heath, D. Ayers, Y. Raimond, Interlinking open data on the web, *Int. Conf. Demonstrations Track, 4th European Semantic Web Conference*, 2007.

[4] C. Bizer, A. Jentzsch, R. Cyganiak, *State of the lod cloud*, Version 0.3 (September 2011).

[5] C. Bizer, T. Heath, T. Berners-Lee, Linked data-the story so far, *International Journal on Semantic Web and Information Systems (IJSWIS)*, Vol.5, No.3, 2009, pp. 1-22.

[6] C. Bizer, R. Cyganiak, D2r server, Version 0.7 *http://www4. wiwiss. fu-berlin. de/bizer/d2r-server.*

[7] C. Bizer, R. Cyganiak, D2r server-publishing relational databases on the semantic web, *Int. Conf. 5th international Semantic Web conference*, 2006.

[8] C. Bizer, R. Cyganiak, D2rq-lessons learned, *Int. Conf. W3C Workshop on RDF Access to Relational Databases*, 2007.

[9] S. Das, S. Sundara, R. Cyganiak, R2rml: Rdb to rdf mapping language, w3c recommendation.

In 2012. *http://www.w3.org/TR/r2rml/ (last modified on 27 September 2012)*

[10] C. Bizer, R. Cyganiak, T. Heath, How to publish linked data on the web, *Tutorial in the 7th International Semantic Web Conference, Karlsruhe, Germany*, 2008.

[11] H. Bai, B. Liang, Semantic Pattern Mapping Between RDBMS and Linked Data Based on Open Source Software. *New Tecnology of Library and Information Service*, Vol.7, No.8, 2011, pp. 1-7.

[12] C. Bizer, D2r map-a database to rdf mapping language, *Int. Conf. The twelfth international World Wide Web Conference, WWW2003*, 2003.

[13] S.S. Sahoo, W. Halb, S. Hellmann, K. Idehen, T. Thibodeau Jr, S. Auer, J. Sequeda, A. Ezzat, A survey of current approaches for mapping of relational databases to rdf, *W3C RDB2RDF Incubator Group Report*, 2009.

[14] Y. Lv, Z. Ma Transformation of relational model to rdf model, *Int. Conf. Systems, Man and Cybernetics*, 2008.

[15] V. Eisenberg, Y. Kanza, D2rq/update: Updating relational data via virtual rdf, *Int. Conf. Proceedings of the 21st international conference companion on World Wide Web*, 2012.

[16] M.S. Marshall, R. Boyce, H.F. Deus, J. Zhao, E.L. Willighagen, M. Samwald, E. Pichler, J. Hajagos, E. Prud'hommeaux, S. Stephens, Emerging practices for mapping and linking life sciences data using rdf—a case series, *Web Semantics: Science, Services and Agents on the World Wide Web*, No.14, 2012, pp. 2-13.

[17] S. Auer, S. Dietzold, J. Lehmann, S. Hellmann, D. Aumueller Triplify: Light-weight linked data publication from relational databases, Int. Conf. *Proceedings of the 18th international conference on World wide web*, 2009.

[18] C. Blakeley, Virtuoso rdf views-getting started guide, *OpenLink Software*, 2007.

[19] J.F. Sequeda, D.P. Miranker Ultrawrap: Sparql execution on relational data, *Technical Report TR-12-10, University of Texas at Austin, Department of Computer Sciences*, 2012.

[20] J. Unbehauen, C. Stadler, S. Auer, Accessing relational data on the web with sparqlmap. *In Semantic technology, Springer*, 2013, pp. 65-80.

[21] AKSW, Sparqlify:Overview. *https://github.com/AKSW/Sparqlify (last modified on 13 June 2013)*

[22] A. Miller, D. McNeil, *Revelytix rdb mapping language specification*, 2010.