Automatically Generating RDF Instances from Relational Data Sources

Kyawt Kyawt San, Khin Nweni Tun
University of Computer Studies, Yangon
kyawtkyawts@gmail.com, knntun@gmail.com

Abstract

The development of Semantic Web mainly depends on enrichment of Semantic Web metadata. The semantic web represents metadata as a relation of triples using the graph based data model, ***Resource Description Framework*** (RDF) *data model. An automatic and simple strategy is needed to deal with a process of mapping relational database to Resource Description Framework data format in order to provide an opportunity for exploration, experimentation and representation of relational data into Web data. The goal of this paper is to make existing relational database content available for Semantic Web applications. This paper fulfills the needs of this metadata by dealing with a successful and automatic production of Semantic Web metadata approach. To demonstrate the practical applicability of our approach, a prototype which is a system for generating RDF document from relational database instances, has been implemented. We make Semantic Web metadata production as simple as HTML publishing.*

Keywords: Semantic Web, Relational Database, RDF

1. Introduction

The vision of the Semantic Web is to extend principles of the Web from documents to data. Data should be accessed using the general Web architecture using, e.g., URI-s (Uniform Resource Identifier); data should be related to one another just as documents (or portions of documents). In addition, the Semantic Web extends the existing web in which it requires a human operator, using computer systems to perform the tasks required to find, search and aggregate its information. It's impossible for a computer to do these tasks without human guidance because Web pages are specifically designed for human readers. The Semantic Web aims to change it by presenting Web page data in such a way that it is understood by computers, enabling machines to do the searching, aggregating and combining of the Web's information — without a human operator.

There are many ways in which one can contribute to creating the Semantic Web. We would like to publish some legacy relational data in RDF.

In order to make this huge amount of relational data available for the Web of Data, a connection must be established between RDBs and a format suitable for the Web of Data. A large body of research work has been focused on mapping the vast quantities of data from RDB to the Resource Description Framework data format. The development of the current web of documents into a semantic web requires the inclusion of large quantities of data stored in relational databases (RDB). Therefore, the study of difference between Semantic Web applications using RDF and relational database is necessary. There are different approaches to convert relational data to RDF. The relational data can be accessed semantically either by generating RDF triples corresponding to original data or by keeping the data in the DB, where it can be managed better and generated RDF on demand.

This paper proposes an approach of mapping relational database instances to RDF representation format. We implement a system for publishing RDF document. Our approach generates not only mapping relational data to RDF but also generates RDF document.

The organization of this paper is as follows: Section 2 focuses on the related work of directly mapping relational data to RDF. Section 3 presents the Semantic Web technologies. Section 4 describes the processing steps for generation RDF instances from relational database and describes the prototype of our implementation results. Section 5 concludes the paper.

2. Related Work

Different researches are investigated in RDB migrations focusing on different domains. The existing technical methods and system prototype are still having the following shortages or defects: they are semi-automatic or manual and require much user interaction. Transformed structures are so simple: e.g. primary keys assumed to be single-column, foreign keys assumed to be single column and relationship assumed only to be 1:1.

In recent years, with the growing importance and benefits provided by Web semantic, there has been a lot of effort on migrating RDBs into the relatively newer technologies (XML/RDF/OWL). Krishna [7] proposed a methodology for representing an ER diagram in RDF, this method maps relational data to an RDF format with the extensive use of user-defined URIref vocabularies. They made the semantics as expressed by the database more explicit. Farouk et al [6] presented an approach for converting DB to RDF with additional defined rules. It focused on adding extra knowledge (user-defined rules) during mapping process. Martin et al [11] proposed the two-layer mapping model of a database schema to an ontology structure for dynamic RDF metadata production. Its main aim is to simplify the work of semantic web presentation developers. Lei et al [3] utilized the relational view mechanism to publish data stored in relational databases with their view-based Triplify approach.

3. Semantic Web Technologies

To represent the semantic web, the following technologies will be used:

1. A global naming scheme(URIs)
2. A standard syntax for describing data(RDF)
3. A standard means of describing the properties of that data(RDF Schema)
4. A standard means of describing relationships between data items(Ontology defined with the OWL Web Ontology Language)

3.1 Uniform Resource Identifier

URIs is used to identify a resource uniquely. As shown in Figure1, a URI can be either a URL, which stands for Unique Resource Location, or URN, that is Unique Resource Name or both. A URL will give indication about the location of a resource. A URN will give information about the name of a resource and therefore we would know exactly what (or who) it is.

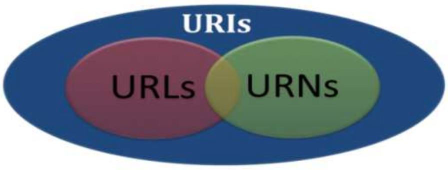

Figure 1: The URIs, containing URLs, URNs and their intersection

It is necessary to resolve any duplication, either by producing URIs based on fully qualified names of schema elements, or by producing them randomly. Every resource is identified by a Uniform Resource Identifier (URI). In the case of a Web page, the URI can be the Unified Resource Locator (URL) of the page. The URI does not necessarily enable the access via the Web to the resource; it simply has to unambiguously identify the resource. The use of Uniform Resource Identifiers (URI) for entities along with the ability to link them together using predicates enables RDF to effectively integrate data from multiple sources.

3.2 Resource Description Framework

The Resource Description Framework (RDF) provides a means for adding metadata annotations to Web resources. RDF is a semantic data model and an attempt to address the aforementioned semantic limitations of XML. It views web data as a set of resources that may also be related to each other, uniquely identified by its Unique Resource Identifier (URI). Information about web entities is expressed through RDF statements.

The basic element of RDF is the triple: a resource (the subject) is linked to another resource (the object) through an arc labeled with a third resource (the predicate). The object of a statement can be another resource, identified by a URI, a literal or a data type value.

For instance, the information that W3C is the owner of the web page http://www.w3.org/RDF, can be expressed through the statement

< http://www.w3.org/RDF, "owner", http://www.w3.org >

Table 1: RDF Statement

Subject (Resource)	< http://www.w3.org/RDF>
Predicate (Property)	Owner
Object (Value)	<http://www.w3.org>

3.3 RDF Schema

While RDF allows associating any property with any web resources, the extended language RDF-Schema is used to define schemas of web resources. RDF Schema is a simple ontology definition language that allows users to define the vocabulary needed to describe the resources in the domain with meta-data. To define the ontology RDFS uses the RDF triples format. In RDFS, user can define classes, properties, and relationships to model the concepts in the domain.

3.4 Web Ontology Language

In general, Ontology provides a mechanism to capture information about the objects and the relationships that hold between them in some domain of interest. OWL ***Web Ontology Language*** was developed to provide a syntax that can be understood directly by computers. OWL ontology is also an RDF graph, which is in turn a set of RDF triples. There are actually three versions of OWL: OWL Lite, OWL DL, and OWL Full.

In order to achieve the goals, the most important is to be able to define and describe the relations among data (i.e., resources) on the Web. Among these Semantic Web technologies, we use URIs to identify a name or a resource on the Internet, RDF is used for creation of meta data about an entity and RDFS is to define the vocabulary needed to describe the resources in the domain with metadata.

4. RDF Generation Process

The RDF generation is a process by which relational data is transformed into RDF triples. In this paper, we propose the process of generating RDF document as shown in Figure 2. To apply transformation process, we take relational data and schema as input and produce an RDF document and publish on the web. To generate RDF document, relative URIs are used against a base URIs to form RDF document. In order to fulfill the process, we only need to extract relational data and transform to RDF by using the transformation rules. These rules are explained detail in following section.

4.1 Transformation Rules

In this section, we would like to present the transforming relational data to RDF by direct transformation rules.

Relation Name ⟷ Subject
Field Name ⟷ Predicate
Value ⟷ Object

From these transformation rules, relational URIs, attribute URIs and tuple URIs are generated and are concatenated to the base URI. Generating suitable URIs for the RDF “resources” is one of the key issues. An essential component of RDF graphs is URIs. It should be generated for relations, attributes and tuples. All URIs are generated by appending to a base URI.

To apply transformation rules, we use the example relational database as shown in Table 2 and show the partial result.

The direct mapping to RDF is done by applying the rules proposed in [5] and simple transformation rules:

Table 2: Sample Relational Database

Relation Name	**Primary Key**	**Foreign Key**
Book(Book ID, Title, AuthorID, PublicationDate, Price, Publisher	BookID	AuthorID
Author (AuthorID, Name, Address)	AuthorID	
Publisher(PID, PName, Address)	PID	
User (UID, Name, Address)	UID	
UserBook (UID, BookID, Date)	UID, BookID,Date	UID,BookID

(i) Relation URI

Relation URI is an URI formed from the concatenation of the base URI, table name, primary key column name and primary key value.

RelationURI(X, Y) $\leftarrow$ Rel(X), $Concat_2$ (http://www.Semanticweb.com/, X, Y)
e.g., http://SemanticWeb.com/library/book/ID=1

(ii)Attribute URI

Attribute URI is an URI formed from the concatenation of the base URI, table name and the column name.

URIs for attributes (n>= 1)

$AttrURI_n$ (X1. . . Xn, Y, Z) $\leftarrow$ Rel(Y),Attr(X_1, Y), . . . Attr(X_n, Y), Concat $_{2+2n}$ (http://www.SemanticWeb.com/, Y, "#", X1, ",", X_2, ",", . . . , ",", X_n, Z)

e.g., <http://SemanticWeb.com/library/book#ID>
<http://SemanticWeb.com/library/book#BookTitle>
<http://SemanticWeb.com/library/book#AuthorId>
<http://SemanticWeb.com/library/book#PublicDate>
<http://SemanticWeb.com/library/book#Price>
<http://SemanticWeb.com/library/book#PublisherId>

(iii) Tuple URI

Tuple URI is the values of RDF literals formed from the lexical form of the column value.

URIs for tuples ($n \geq 1$):

TupleID(X, Y, Z) $\leftarrow$ Rel(Y),PK_n (X1, . . . , X_n, Y), Value(V_1, X_1, X, Y), . . . ,Value(V_n, X_n, X, Y), $Concat_{2+4n}$(http://www.SemanticWeb.com/,Y,"#", X_1, "=", V_1, ",",X_2, "=", V_2, . . . , ",", X_n, "=", V_n, Z)

We show some of the tuple triples only.

<http://SemanticWeb.com/library/book>
<http://SemanticWeb.com/library/book#ID > 1

<http://SemanticWeb.com/library/book>
<http://SemanticWeb.com/library/book#AuthorID>
< http://SemanticWeb.com/library/author/ID=1>

Each row is turned into a series of triples with a common subject. We get the final RDF triples by one processing step. Once information is in RDF form, it becomes easy to process it.

4.2 Generating RDF Document

A prototype has been developed to show the proposed transformation system of RDF triples. The mapping process is implemented using java and mysql. Using the relational database shown in Table 2, the converting to RDF process is done.

To implement RDF generation of our proposed system, we first export relational data. Then relational database content is transformed into RDF using the presented transformation rules. In our approach, all types of relationships between tables are considered. Blank nodes problems are also solved by assigning URI references to blank nodes.

Figure 2 shows a process of publishing RDF document generated from relational data source. In order to complete the process, we first export relational data content. Then, assigning URI to relational data content is done by concatenating the relative URI to the base URI in accordance with the transformation rules. Each tuple becomes a series of triples with a common subject. In our approach, we overcome the converting problems of composite primary keys and foreign keys. All types of relationships between tables are also considered. Resulting RDF can be stored in static RDF documents or in a native RDF database. During the mapping process, a corresponding RDF instances are automatically generated by the system applying transformation rules. Generated RDF documents are validated against RDF Validator and Converter and obtain the RDF triples. RDF code for entire example database is too long and we show some of the triples only. The fragment of the RDF code is written in NTriples format as follows.

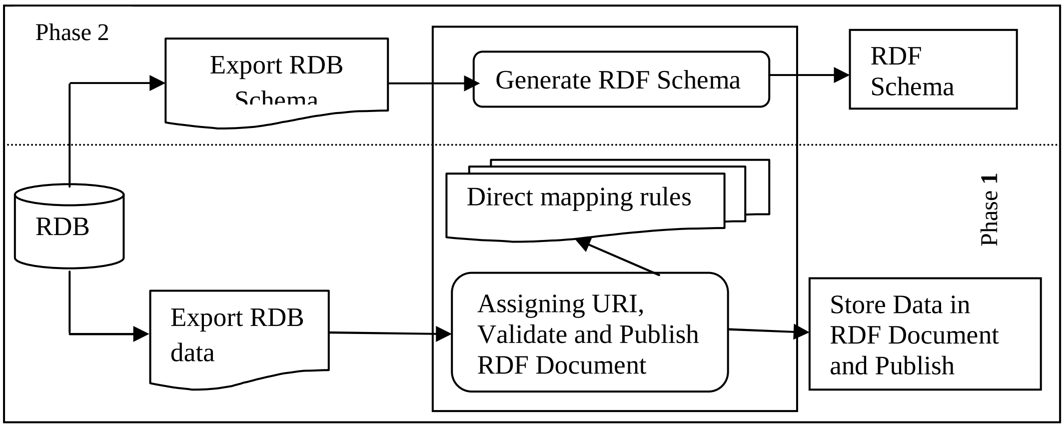

Figure 2: A Process for Automatic Generation of RDF Document

(1) <http://SemanticWeb.com/library/book/ID=1> <http://SemanticWeb.com/library/book#Id> 1.

(2) <http://SemanticWeb.com/library/book/ID=1> <http://SemanticWeb.com/library/book#BookTitle> "Focus on IELTS".

(3) <http://SemanticWeb.com/library/book/ID=1> <http://SemanticWeb.com/library/book#AuthorId> <http://SemanticWeb.com/library/author/ID=1>.

(4) <http://SemanticWeb.com/library/book/ID=1> <http://SemanticWeb.com/library/book#PublicDate> 2005.

(5) <http://SemanticWeb.com/library/book/ID=1> <http://SemanticWeb.com/library/book#Price> 2000.

(6) <http://SemanticWeb.com/library/book/ID=1>< http://SemanticWeb.com/library/book#PublisherId> <http://SemanticWeb.com/library/publisher/ID=1>.

(7) <http://SemanticWeb.com/library/book/ID=2> <http://SemanticWeb.com/library/book#Id> 2.

(8) <http://SemanticWeb.com/library/book/ID=2> <http://SemanticWeb.com/library/book#BookTitle> "Step up to IELTS".

(9) <http://SemanticWeb.com/library/book/ID=2> <http://SemanticWeb.com/library/book#AuthorId> <http://SemanticWeb.com/library/author/ID=2>.

(10) <http://SemanticWeb.com/library/book/ID=2> <http://SemanticWeb.com/library/book#PublicDate> 2004.

(11) <http://SemanticWeb.com/library/book/ID=2> <http://SemanticWeb.com/library/book#Price> 1500.

(12) <http://SemanticWeb.com/library/book/ID=2> < http://SemanticWeb.com/library/book#PublisherId> <http://SemanticWeb.com/library/publisher/ID=2>.

Table 3: Comparison of existing methods and the implemented system

Name	Mode	Data Synchronization	Use GUI
METAmorphoses	Manual	Static	Yes
Lei et al	Manual	Dynamic	Yes
Proposed System	Auto	Dynamic	Yes

Table 3 illustrates the comparison between existing methods and the proposed method. We show some of the comparisons only due to space limited. Compared with some other existing methods, our system generates semantic meta data automatically and provides simple programming interface. In addition, all types of relationships between tables are considered.

5. Conclusion

Our experimental result demonstrates that our approach performs well and processes automatic generation. It can be obviously seen that mass generation of Semantic Web meta data is needed. Our system can be used where data is stored in a relational database and there is a need for generating RDF automatically without domain expert.

References

[1] “A Direct Mapping of Relational Data to RDF “, W3C Working Draft 29 May 2012, http://www.w3.org/TR/2012/WD-rdb-direct-mapping-20120529/.

[2]C.Huajun, W.Yimin, W.Heng,et al, “ Towards a Semantic Web of Relational Databases: a Practical Semantic Toolkit and an In-Use Case from Traditional Chinese Medicine “,http://ccnt.zju.edu.cn/projects/dartgrid.

[3]C.Lei, Y.Nansheng, “ Publishing Linked Data from Relational Databases Using Traditional Views “, IEEE, 2010.

[4]D.Beckett, J.Grant, “SWAD-Europe Deliverable 10.2: Mapping Semantic Web Data with RDBMSes, http://www.w3.org/2001/sw/Europe/reports/scalable_rdbms_mapping_report.
[5]Kyawt Kyawt San, Khin Nweni Tun, “Formalization of Direct Mapping for Relational Data to Semantic Web Representation”, ICCA 2012.
[6]M.Farouk , M.Ishizuka, “Converting DB to RDF with Additional Defined Rules”, 8thInternationalConferenceonWebInformationSystemsandTechnologies, 2012.
[7]M.Krishna, “Retaining Semantics in Relational Databases by Mapping them to RDF”, Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology,2006.
[8]Miranker, D., Lecture Notes in Computer Science, Volume 5181/2008 Database and Expert Systems Applications-(2008).
[9]N.Gherabi, K.Addakiri, “Mapping relational database into OWL Structure with data semantic preservation “, (IJCSIS) International Journal of Computer Science and Information Security, Vol.10, No.1, 2012.
[10]N.Lopes, “An overview of RDB2RDF techniques and tools”, DERI Reading Group Presentation, 2009.
[11]S.H.Tirmizi, J.Sequeda, D.Miranker, “ Translating SQL Applications to the Semantic Web”, Springer-Verlag Berlin Heidelberg 2008.
[12]S.Martin, J.Ivan, “The Database to RDF Mapping Model for an Easy Semantic Extending of Dynamic Web Sites “, 2005.