

D2R MAP – A Database to RDF Mapping Language

Christian Bizer
Freie Universität Berlin
Institut für Produktion,
Wirtschaftsinformatik und OR
Garystr. 21, D-14195 Berlin, Germany
+49 30 838 54057

bizer@wiwiss.fu-berlin.de

ABSTRACT

The vision of the Semantic Web is to give data on the web a well-defined meaning by representing it in RDF and linking it to commonly accepted ontologies. Most formatted data today is stored in relational databases. To be able to use this data on the Semantic Web, we need a flexible but easy to use mechanism to map relational data into RDF. The poster presents D2R MAP, a declarative language to describe mappings between relational database schemata and OWL ontologies.

Categories and Subject Descriptors

H.2.3 [Database Management]: Languages

General Terms

Standardization, Languages

Keywords

Semantic Web, relational databases, RDF data model, mapping

1. INTRODUCTION

The Semantic Web is an extension of the current Web, in which data is given a well-defined meaning by representing it in RDF and linking it to commonly accepted ontologies. This semantic enrichment allows data to be shared, exchanged or integrated from different sources and enables applications to use data in different contexts [1].

Most formatted data today is stored in relational databases. To be able to use this data in a semantic context, it has to be mapped into RDF, the data format of the Semantic Web.

The data model behind RDF is a directed labelled graph, which consists of nodes and labelled directed arcs linking pairs of nodes [2]. To export data from an RDBMS into RDF, the relational database model has to be mapped to the graph-based RDF data model.

2. D2R LANGUAGE FEATURES

D2R MAP is a declarative, XML-based language to describe such mappings. The main goal of the language design is to allow flexible mappings of complex relational structures without having to change the existing database schema. This flexibility is achieved by employing SQL statements directly in the mapping rules. The resulting record sets are grouped afterwards and the data is mapped to the created instances. This approach allows the handling of binary and higher degree relationships, multivalued

class properties, complex conditions and highly normalized table structures, where instance data is spread over several tables.

The mapping process performed by the D2R processor has four logical steps as shown in Figure 1. For each class or group of similar classes a record set is selected from the database. Second, the record set is grouped according to the `groupBy` columns of the specific `ClassMap`. Then the class instances are created and assigned a URI or a blank node identifier. Finally, the instance properties are created using datatype and object property bridges.

The division between step three and four allows references to blank nodes within the model and to instances dynamically created in the mapping process.

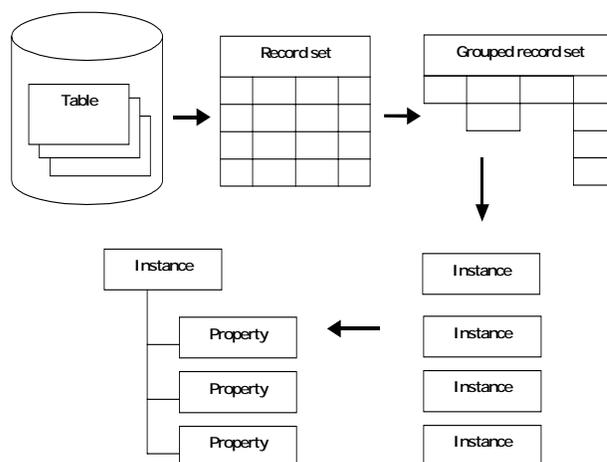


Figure 1. The D2R mapping process.

The second goal is to keep D2R MAP as simple as possible. Apart from elements to describe the database connection and the namespaces used, the actual mappings are expressed with just three elements. For each class or group of similar classes in the ontology a `ClassMap` element is used. Each `ClassMap` has an `sql` attribute and a `groupBy` attribute. To create instance URIs, patterns and value substitution tables can be used. The instance properties are constructed with `DataTypePropertyBridge` elements for literal properties, which can be typed using XML datatypes and `xml:lang` attributes. Datatype properties can be converted similarly using patterns and value substitution tables. References to external resources or instances within the model are created with an `ObjectPropertyBridge` element. To refer to the instances created on the fly, a `referredClass` together with a `referredGroupBy` attribute is used. Multiple values of a single property can be put in `rdf:Bag`, `rdf:Alt` or `rdf:Seq` containers, using the `useContainer`

attribute together with a `DataTypePropertyBridge` or `ObjectPropertyBridge` element.

3. EXAMPLE

The following example illustrates the use of a D2R MAP to export data about authors and their publications from a database into RDF. Because authors usually have more than one publication and publications can be written by multiple authors, the information would typically be stored in three database tables: One for the authors, one for their publications and a third one for the n:m relationship between authors and publications. A D2R MAP transformation of these tables to the classes `ex:Author` and `ex:Book` could look as follows:

```
<Map>
  <DBConnection jdbcDSN="bookDB" />
  <ProcessorMessage
    outputFormat="RDF/XML-ABBREV" />
  <Namespace prefix="ex"
    namespace="http://example.org#" />
  <ClassMap type="ex:Book"
    sql="SELECT isbn, title FROM books;"
    groupBy="isbn"
    uriPattern="ex:book@@isbn@">
    <DatatypePropertyBridge
      property="ex:title"
      column="title" xml:lang="en" />
  </ClassMap>
  <ClassMap type="ex:Author"
    sql="SELECT authors.aid, name, URL,
      isbn FROM authors, bookauthor
      WHERE authors.aid = bookauthor.aid;"
    groupBy="authors.aid">
    <DatatypePropertyBridge
      property="ex:fullname"
      column="name" />
    <ObjectPropertyBridge
      property="ex:homepage"
      column="URL" />
    <ObjectPropertyBridge
      property="ex:author_of"
      referredClass="ex:Book"
      referredGroupBy="isbn"
      useContainer="rdf:Bag" />
  </ClassMap>
</Map>
```

The first three subelements define the database connection, the desired output format and an example namespace. The first `ClassMap` element describes the mapping for the `ex:Book` class. The instance URIs are created using a pattern. The `xml:lang` attribute is set for the title property.

The second `ClassMap` describes the creation of `ex:Author` instances and links the authors to their publications using an `rdf:bag` container for the `ex:author_of` property. Because the `ex:Author` class map contains no URI construction schema, instances are identified as blank nodes. The following example instance is created with the map above:

```
<ex:Author rdf:nodeID='A465'>
  <ex:fullname>Chris Bizer</ex:fullname>
  <ex:homepage rdf:resource=
    'http://www.bizer.de' />
```

```
<ex:author_of>
  <rdf:Bag>
    <rdf:li rdf:resource=
      'http://example.org#book321230273' />
    <rdf:li rdf:resource=
      'http://example.org#book884237273' />
  </rdf:Bag>
</ex:author_of>
</ex:Author>
```

This example shows only some features of D2R MAP. The complete language specification and further examples are found at <http://www.wiwiss.fu-berlin.de/suhl/bizer/d2rmap/D2Rmap.htm>.

A D2R processor prototype is publicly available under GNU LGPL license. The processor is implemented in Java and is based on the Jena API [3]. It exports data as RDF, N3, N-TRIPLES and Jena models. It is compliant with all relational databases offering JDBC or ODBC access. The processor can be used in a servlet environment to dynamically publish XHTML pages containing RDF, as a database connector in applications working with Jena models or as a command line tool.

4. RELATED WORK

Other mapping approaches which have influenced the design of D2R MAP, are developed by the AIFB Institute, University of Karlsruhe, Germany [4] and by Boeing, Philadelphia, USA [5]. It is planned to extend D2R MAP with conditional mappings and more sophisticated value transformation abilities. These extensions could be based on RuleML [6], RDFT [7] or further language constructs borrowed from XSLT.

REFERENCES

- [1] James Hendler, Tim Berners-Lee, Eric Miller. Integrating Applications on the Semantic Web. *Journal of the Institute of Electrical Engineers of Japan*, Vol 122(10), October 2002, p.676-680.
- [2] Graham Klyne, Jeremy Carroll (eds.). *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C Working Draft (work in progress). November 2002, <http://www.w3.org/TR/2002/WD-rdf-concepts-20021108/>.
- [3] Brian McBride. *Jena: Implementing the RDF Model and Syntax Specification*. Technical report, Hewlett Packard Laboratories (Bristol, 2000). <http://www.hpl.hp.com/semweb/index.html>.
- [4] Nenad Stojanovic, Ljiljana Stojanovic, Raphael Volz. A reverse engineering approach for migrating data-intensive web sites to the Semantic Web. *IIP-2002 (Montreal, 2002)*. <http://www.aifb.uni-karlsruhe.de/WBS/nst/docs/papers/IIPv31finalv1.pdf>.
- [5] Tom Barrett et al. *RDF Representation of Metadata for Semantic Integration of Corporate Information Resources*. WWW2002 (Hawaii, 2002). http://www.cs.rutgers.edu/~shklar/www11/final_submissions/paper3.pdf.
- [6] RuleML. <http://www.dfki.uni-kl.de/ruleml/>.
- [7] Borys Omelayenko. *RDFT: A Mapping Meta-Ontology for Business Integration*. ECAI-2002 (Lyon, 2002) <http://www.cs.vu.nl/~borys/papers/rdft4ktsw02.pdf>.