Efficient RDF Schema Mapping and Triples Generation Based on ETL Tool

Jiao Li, Guojian Xian
Agricultural Information Institute of CAAS
Current methods to generate RDF(Resource Description Framework) data

1. RDF data extraction from Relational Database (RDB)
   - mainstream, RDB-to-RDF/RDB2RDF

2. other format (CSV, Excel, JSON and XML files) to RDF

https://www.w3.org/2001/sw/wiki/Category:RDF_Generator
Current methods to RDB-to-RDF

- **Ontology matching**: Concepts and relations are extracted from relational schema or data by using data mining, and then mapped to a temporal established ontology or specific database schema.

- **Mapping Language**: This involves cases of low similarity between database and target RDF graph, as exampled by R2RML, which enables users express the desired transformation by following chosen structure or vocabulary.

- **Query Engine-based**: Transformation process is based on the SPARQL query of search engines with capability in supporting large collection of concurrent queries.
## General Tools for RDB2RDF

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
<th>Input</th>
<th>Output Format</th>
</tr>
</thead>
</table>
| D2RQ          | a system for accessing relational databases as virtual, read-only RDF graphs. It offers RDF-based access to the content of relational databases without having to replicate it into an RDF store. Using D2RQ you can:  
  • query a non-RDF database using SPARQL  
  • access the content of the database as Linked Data over the Web  
  • create custom dumps of the database in RDF formats for loading into an RDF store  
  • access information in a non-RDF database using the Apache Jena API | Oracle, MySQL, PostgreSQL, SQL Server, HSQLDB, Interbase/Firebird | RDF            |
| Triplify      | a small PHP plugin for Web applications, which reveals the semantic structures encoded in relational databases by making database content available as RDF, JSON or Linked Data | Relational Database | RDF, JSON, Linked data |
| R2RML Parser  | export relational database contents as RDF graphs, based on an R2RML mapping document. Contains an R2RML mapping document for the DSpace institutional repository solution | Relational Database, MySQL, PostgreSQL, Oracle | Turtle, N-Triples, RDF/XML, Notations3 |
But, these tools can not fully included:

- support most non-RDF data formats and output formats
- offer a packaged and multifunctional RDF data process method without programing
- integrated use with the triple stores

So we tried to:

- merge RDF generation with ETL(Extract-Transform-Load)
- redevelop the prominent ETL tool to an RDF ETL framework in a semantic-based way
- provide a user-friendly, open to use and intuitive interface
Our solution for RDF generation and management

RDF ETL plugin: RDFZier

New developed plugin:

• based on Kettle (a leading open-source ETL application on the market) in an ETL environment
• RDF 4J
• support multiple mainstream non-RDF format inputs AND ETL of multi-source heterogeneous data
• offer one-stop templates without coding
• efficient paralleling process that can provide multithreaded operations
• store multiple types of outputs into a selected RDF endpoint (triple store) or file system
General View

Query the chosen field information with SQL language.
Format supported

Input:
- Relational database (MySql, SqlServer), NoSQL, Data Stream/Text file (csv, Excel, json, XML)...

Output format:
- Turtle, JSON-LD, N-triples, RDF/XML, NQuads, TriG, RDF/JSON, TriX, RDF Binary
Parameters defined in RDFZier

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Namespace</td>
<td>Prefix</td>
</tr>
<tr>
<td></td>
<td>collections of names identified by URI references</td>
</tr>
<tr>
<td></td>
<td>different prefixes depending on the required namespaces</td>
</tr>
<tr>
<td>Subject URI</td>
<td>HTTPURI template for the Subject/Resource, a placeholder (sid) would be used</td>
</tr>
<tr>
<td></td>
<td>and replaced by UniqueKey</td>
</tr>
<tr>
<td>Class Types</td>
<td>the classes to which the resource belongs, supporting multi-class types</td>
</tr>
<tr>
<td></td>
<td>(split by semicolon), such as skos:Concepts; foaf:Person</td>
</tr>
<tr>
<td>UniqueKey</td>
<td>the unique and stable primary key of resource, part of the Subject URI</td>
</tr>
<tr>
<td>Fields Mapping Parameters</td>
<td>a list of field map from selected data source to target RDF schema,</td>
</tr>
<tr>
<td></td>
<td>including the input Stream Field, Predicates, Object URIs, Multi-Values</td>
</tr>
<tr>
<td></td>
<td>Separator, Data Type, Lang Tag</td>
</tr>
<tr>
<td>Dataset Metadata</td>
<td>Meta Subject URI</td>
</tr>
<tr>
<td></td>
<td>URI pattern of generated dataset</td>
</tr>
<tr>
<td></td>
<td>Meta Class Types</td>
</tr>
<tr>
<td></td>
<td>the classes to which the resource belongs</td>
</tr>
<tr>
<td>Parameters</td>
<td>a list of descriptions of generated dataset, including PropertyType,</td>
</tr>
<tr>
<td></td>
<td>Predicates, Object Values, Data Type, Lang Tag</td>
</tr>
<tr>
<td>Output Setting</td>
<td>File system setting</td>
</tr>
<tr>
<td></td>
<td>option for file system storage, including Filename and RDF format</td>
</tr>
<tr>
<td></td>
<td>RDF store setting</td>
</tr>
<tr>
<td></td>
<td>option for RDF store, including triple store name, server URL, Repository</td>
</tr>
<tr>
<td></td>
<td>ID, Username (if any), Password, Graph URI</td>
</tr>
</tbody>
</table>
Output setting

Save to File:  local system
Save to Store:
- virtuoso
- GraphDB
- Blazegraph
- MarkLogic
Example of use

- one-stop RDF generation from RDB
- direct mapping
- field mapping rules or a semantic schema is must
Triple store--Virtuoso

**SPARQL Query**

```
select *
{<http://linked.aginfra.cn/sci/kg/journal_article/H.13918063> ?p ?o}
```
Future View

- Multi-format Data Conversion and Loading (between different serialization formats or Endpoints)
- Remote RDF Data Migration
- RDF Graph Update (by using SPARQL 1.1 update)
Thank you!

Questions/Comments?

lijiao@caas.cn
xianguojian@caas.cn