RESTful writable APIs for the web of Linked Data using relational storage solutions

Antonio Garrote
María N. Moreno García
Outline

1. Introduction
2. SPARQL to SQL mapping using R2RML
3. RESTful API
4. Conclusions
Web APIs

- Mobile App.
- JS. App
- Native App.

OAuth

- JSON

HTTP API

- MVC
  - Dynamic Web App.
  - Key Value Store
  - Relational DB
Linked data APIs?

• RDF data model vs JSON objects
• Object IDs vs URIs
• Properties identified by URIs vs plain strings
• URIs linking data in different services
Transition path

- Translation of SPARQL update queries into SQL queries
- RDF graphs encoded in JSON objects
- RESTful protocol to manipulate RDF graphs
SPARQL to SQL translation

- R2RML starting point

Diagram:

- RDF quads
- TableMappings
- Relational Data
- R2RML Document
SPARQL to SQL translation

- inverse transformation
SPARQL to SQL translation

R2RML Mapping

Constant Mappings

Variable Mappings

(Table, Subject, Property, Object, Graph)

Quad Matchers
SPARQL to SQL translation

(Table, Subject, Property, Object, Graph)

Compatible?

Constant Terms

Variables +BNode IDs

SPARQL Quad Pattern
SPARQL to SQL translation

SELECT

Quad Matcher

Quad Matcher

Quad Matcher

σ π

σ π

σ π

∪

∪

∪

SPARQL Relational Algebra
SPARQL to SQL translation

Insert

Quad Matcher

Quad Matcher

Quad Matcher

Min. Insertion Cost metric

Insert / Update SQL DM Query
SPARQL to SQL translation

- Different compatibility functions: data types, language tags in literals
- R2RML extensions to generate IDs from URIs
- Limitations: auto increment columns and database constraints
RESTful API

• Starting point: “SPARQL 1.1 Uniform HTTP Protocol for Managing RDF graphs”

• Granularity: RDF named graph

• HTTP uniform interface semantics
RESTful API

- Declarative definition of APIs (RDFS vocabulary)
  - URI templates
  - Mapping of templates to SPARQL endpoints
RESTful API

- Minting of resource URIs

HTTP POST Request

\[
[ 
\langle p1 \rangle \langle o1 \rangle ; \\
\langle p2 \rangle \langle o2 \rangle ; \\
... 
\] 
\]

RDF

HTTP GET Response

\[
<\text{uri}> 
\langle p1 \rangle \langle o1 \rangle ; \\
\langle p2 \rangle \langle o2 \rangle ; \\
... 
\]

RDF

200

graph_uri

201

New resource URI

graph_uri#self
RESTful API

• Extensions (linked-data-api):
  - JSON-LD as the primary media type
  - Method overloading
  - JSONP support
  - Format URL parameter
  - Pagination parameters
Sample Application

- Library
- Prototype
  - JavaScript client
  - RESTful API
  - FOAF+SSL auth
  - Relational backend

http://github.com/antoniogarrote/clj-r2rml

http://antoniogarrote.com/cvbuilder
Conclusions

- Linked data and sem. web technologies can help us to build better web APIs
- Huge opportunity to increase the amount of linked data available in the web
- Real benefits for end users
- Easy transition path for web developers must be provided