Case Study on Linked Data and SPARQL Usage for Web Application Development

Josef Petrák, Jan Zemánek, and Vojtěch Svátek

Lab. of Intelligent Systems & Department of Information and Knowledge Engineering,
University of Economics, Prague me@jspetrak.name, jan.zemanek@gmail.com,
svatek@vse.cz

Abstract. Usual web applications have a relational databases with an application–specific schema in the backend. To alter such a database schema we must run various queries onto the database, and to exchange them we have to choose another application–specific data format. An alternative is RDF as a web standard allowing to create reusable data schemas and interchangeable data. For a real case study we selected an academic community website. In its development we used various common schemas and stored data in an RDF store. Furthermore we published data as Linked Data and experimented with connecting external RDF repositories to enrich our data.

Keywords: SQL, RDF, RDFa, XHTML, SPARQL, Semantic Web Linked Data, Linking Open Data, LoD, CRUD database applications, Web application development

1 Introduction

Most current websites are human-readable only. Even if they are dynamically generated, all semantics of presented data is lost as they are published as (X)HTML pages. People typically understand data such as titles, author names, dates, etc. properly, but computers treat the pages as a set of formatting tags, and do not understand the meaning of presented information. Even if a person understands the content of `<p class="author">` tag as a name of the author, a computer understands it just as a paragraph styled with a stylesheet class of the name "author".

Resource Description Framework was introduced to represent data in a computer-readable form. They are handled as a graph of triples in the form subject–predicate–object, where any resource and any term is identified by a unique URI. It does not dictate the syntax. Except the referential RDF/XML syntax [3] there are many more such as Notation 3 [7], RDF/JSON or, finally, RDFa [2], which is a set of attributes to encode RDF metadata into any XML format. RDF does not define the data schema, but for various domains there are available independent specifications of terms (called ontologies, schemas, vocabularies or taxonomies). Such schemas are de facto standards to express information in various domains: FOAF for people and their relationships, SIOC for interlinked communities, RDF Calendar [5] for calendar events, Dublin Core for library metadata, DOAP for software project description etc. Using such standard schemas increases the usability of our data.

2 Well-Defined Linked Data

Every object–called resource in RDF–has a global identifier in the form of a URI. There are two kinds of resources: information and non–information ones. *Information resources* have a representation on the Web. They include web pages, digital images, audio files, etc. *Non–information resources* represent real-world concepts such as books, people, events, vocabulary terms and many other abstract concepts.

The design of Linked Data [8] provides recommendations on how to publish both kinds of resources, and it is also an initiative on publishing data from various closed databases in RDF. DBPedia[1] is an example of such a Linked Data database of information coming from Wikipedia. The most current release—DBpedia 3.4—describes 2.9 million entities including people, places, organizations, etc. Each of them is one non–information resource with a unique URI and its information representation. It is recommended to reuse resources from such knowledge bases instead of creating new individuals.

3 RDF and SPARQL in Use

Selecting RDF to store data allows to easily make changes in the data schema. Information is stored in the form of triples, so adding one new attribute is an easy operation of creating a new triple. In relational databases this usually requires to alter a whole table and to add one new column with some default value for all information already stored in that table.

In our experimental application on research group website[2] most of resources are non-information ones. When creating or updating information, web–based forms are used to input data, and other scripts generate XHTML+RDFa and RDF/XML information representations for these resources, each having a different URI. It is possible to make content dereferencing of resource URI: according to content negotiation one of representations (RDF or XHTML) is returned [6]. To create, read or update triples [9] about an information resource, SPARQL [1] queries are executed by one of available RDF application frameworks such as ARC2[3] or Sesame.[4]

The first example shown describes a simple calendar of events. iCalendar[5] is a standard for describing calendar data accepted by current calendar and planning

[1] http://dbpedia.org/

[2] http://keg.vse.cz/

[3] http://arc.semsol.org/

[4] http://openrdf.org/

[5] RFC 5545 Internet Calendaring and Scheduling Core Object Specification (iCalendar)

tools. For describing such information in RDF there is an RDF Calendar schema using the same concepts. When using this schema we can serve directly the RDF Calendar data for an RDF–aware tool, or transform data into a non–RDF syntax called iCal and share data without any loss of information. An example RDF/N3 description of an event is given below. It has a type, a summary, and a start and end date.

```
<http://keg.vse.cz/resource/event/2007-10-22-270153889>
  a ical:Vevent ;
  ical:summary "KEG hosted the meeting of the EU MedIEQ project." ;
  ical:dtstart "2007-10-22" ;
  ical:dtend "2007-10-23" .
```

The second example is about a member database and how we can get more information about those people from an external RDF database just using the knowledge of non–information (i.e. person) resource identifier and his or her aliases. A short example of RDF/N3 description of a person is given below. The FOAF schema is heavily accepted to describe people, their contacts and relationships.

```
<http://keg.vse.cz/resource/person/jan-zemanek> a foaf:Person ;
  foaf:name "Jan Zemanek" ;
  foaf:mbox <mailto:jan.zemanek@gmail.com> ;
  owl:sameAs <http://data.semanticweb.org/person/jan-zemanek> ;
  owl:sameAs <http://dblp.l3s.de/d2r/resource/authors/Jan_Zem%C3%A1nek> .
```

Also schemas to describe RDF schemas are defined in RDF. One of them is OWL [4], which proposes a property called `owl:sameAs`, very useful to handle various resource identifying the same object. To keep all information linked it is necessary to create such aliases. We have created two aliases pointing to the identifier in an external data stores[6] to be able to query those stores and to link objects in these databases. Using one SPARQL query and this alias we can get the list of publications published by this person.

```
SELECT ?title, ?url
WHERE {
  ?publication dc:creator
      <http://data.semanticweb.org/person/jan-zemanek> .
  ?publication dc:title ?title .
  ?publication swrc:url ?url .
}
```

The query will result into a list of publication titles and URIs where these publications are available. While we have two different RDF stores we can run two different remote queries [10] or query SPARQL endpoints and put the results from both together.

[6] http://data.semanticweb.org/ or http://dblp.l3s.de/d2r/

4 Future Work and Conclusions

Having data expressed in a RDF graph allows to interlink them. We will also use RDF and SPARQL for data integration within a bookmarking application based on a relational database. A selected schema to publish the tagging folksonomy will help us define keywords and interlink lists of publications and presentations with ”related” bookmarks depending on such keywords. From other examples it is also possible to link geospatial locations for events, related documents and people to the event; we can also integrate data from other external RDF–based web applications. A combination of RDF and SPARQL allows to develop a dynamically generated website with the same effort as with SQL and databases. It also allows to better reuse existing data schemas and make changes in the data structure, integrate applications, exchange the data with other applications on the web, and discover new data and understand them by user agents.

Acknowledgements *This work was partially supported by the CSF project no.P202/10/0761, ”Web Semantization”.*

References

1. E. Prud’hommeaux, A. Seaborne (ed.). SPARQL Query Language for RDF. http://www.w3.org/TR/rdf-sparql-query/, 2008.
2. B. Adida, M. Birbeck, S. McCarron, S. Pemberton. RDFa in XHTML: Syntax and Processing – A collection of attributes and processing rules for extending XHTML to support RDF http://www.w3.org/TR/rdfa-syntax/, 2008.
3. D. Beckett, B. McBride (ed.). RDF/XML Syntax Specification (Revised) http://www.w3.org/TR/rdf-syntax-grammar/, 2004.
4. M. K. Smith, C. Welty, D. L. McGuinness (ed.). OWL Web Ontology Language Guide http://www.w3.org/TR/owl-guide/, 2004.
5. D. Connolly, L. Miller (ed.). RDF Calendar – an application of the Resource Description Framework to iCalendar Data. http://www.w3.org/TR/rdfcal/, 2005.
6. C. Bizer, R. Cyganiak, T. Heath. How to Publish Linked Data on the Web. http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/, tutorial, 2008.
7. T. Berners–Lee. Notation 3. http://www.w3.org/DesignIssues/Notation3, in Design Issues, 1998.
8. T. Berners–Lee. Linked Data. http://www.w3.org/DesignIssues/LinkedData.html, in Design Issues, 2006.
9. D. J. Lewis. Building Semantic Web CRUD operations using PHP. http://www.ibm.com/developerworks/opensource/library/os-php-crud/index.html, 2008.
10. Simon Schenk, Josef Petrák. Sesame RDF Repository Extensions for Remote Querying. http://znalosti2008.fiit.stuba.sk/download/articles/znalosti2008-Schenk.pdf, in proceedings of Znalosti 2008.