Temporary RDF graphs in the Linked Data and REST style

Dominik Tomaszuk[1]
[1]University of Bialystok, Institute of Computer Science
dtomaszuk@ii.uwb.edu.pl

Abstract. This paper describes a simple method of linking to external ad hoc RDF graph. We propose a mechanism based on the Hypertext Transfer Protocol standard Representational State Transfer software architecture. This proposal also comply Linked Data method of publishing Resource Description Framework. It can be used to express select and create operations across various sources.

Keywords. Semantic Web, Linked Data, Representational State Transfer (REST), Resource Description Framework (RDF)

1. Introduction and motivation

The rules of Linked Data [1] can provide a very simple guide for selecting and modifying data on the graph store. In RESTful web services [2] requests and responses are built around the transfer of representations of resources. We attempt to define the proper methods to find and modify the RDF data in graph store with web service means. In this paper a new architectural style access to graph stores based on Linked Data and Representational State Transfer (REST) is presented. This proposal can be used without dedicated applications, it can be executed in web browsers.

The paper is constructed as follows. In Section 2, we propose a flexible solution for generate temporary Resource Description Framework (RDF) graph. Section 3 is devoted to related work. The paper ends with conclusions.

2. Introducing temporary graphs

In this Section we define syntax and semantics of temporary graphs. We also introduce algorithm to generate this external ad hoc RDF graph. We also propose formal grammar for our approach.

Following [3], let I be the set of all Internationalized Resource Identifier (IRI) references, B an infinite set of blank nodes, L the set RDF plain literals, and D the set of all RDF typed literals. I, B, L and D are pairwise disjoint. Let $O = I \cup B \cup L \cup D$ and $C = I \cup B$. An RDF triple T is a triple in $C \times I \times O$. If $T = (s, p, o)$ is RDF triple, s is called the subject, p the predicate and o the object.

2.1. Abstract syntax

A temporary graph TG is a tuple (u, N, v), where u is an IRI reference pointer to source, N is a named graph, and v is a mapping from patterns to RDF graph. A named graph N is a pair (n, G), where n is IRI and called name and G is RDF graph, which is a set of RDF triples T. A pattern P from v is a tuple from $(C \cup V) \times (I \cup V) \times (O \cup V)$ where V is variable, infinite and disjoint from

I, *B*, *L* and *D*.

For instance let *AliceProfile* be a *TG*, then:

AliceProfile = (http://example.org, {}, m),
with m(http://example.org) =
{(-person, rdf:type, foaf:Person)
(-person, foaf:name, -name)}

2.2. Semantics

A mapping m from V to T is a partial function $m : V \rightarrow T$. For a pattern $p \in P$ we denote by $m(p)$ the triple obtained by replacing the variables in p according to m. Let R and S be sets of mappings. We define the natural join binary operator between R and S as:

$R \bowtie S = \{m_1 \cup m_2 : m_1 \in R \wedge m_2 \in S\}$

We also use unary relational algebra operations: projection (π) and selection (σ) as:

$\pi_{a1, \ldots, an}(R) = \{m[a_1, \ldots, a_n] : m \in R\}$

$\sigma_{a\theta b}(R) = \{m : m \in R, m(a)\ \theta\ m(b)\}$

For instance let *(-person, rdf:type, foaf:Person)* then it translates to:

$\pi_{\text{-person}}(\sigma_{p=\text{rdf:type}}(T))$

When we have set of two or more triples we should join them:

$\pi_{\text{-person}}(\sigma_{p=\text{rdf:type}}(T)) \bowtie \pi_{\text{-person, -name}}(\sigma_{p=\text{foaf:name}}(T))$

2.3. Generating temporary graphs

A variable is prefixed by "-" and the "-" is not part of the variable name. Triple patterns should begin after graph name. Multiple triple patterns are separated by "/". Elements of triple pattern are separated by "|". All prefixes of abbreviated IRIs should be associated with well-known IRI [4]. If prefixes are not in well-known IRI, it is literal. Typed literal is suffixed by "+" and IRI to data type. Blank node is prefixed by "_". A triple pattern *TP* in IRI is a pattern in $(S \cup V \cup P) \times (I \cup V \cup P) \times (O \cup V \cup P)$, where P is a prefix name.

For example *http://example.org/-person|rdf:type|foaf:Person/-person|foaf:name|-name/* with GET method generating triples from that have FOAF [5] name[1] in predicate.

The algorithm of generating temporary graphs:

input: IRI with TP and G
output: temporary RDF document
if Prefixes $\notin \varnothing$ *then*
  foreach x $\in$ *Prefixes do*
    change x to IRI
numberOfStatement = 0
isCorrectSubject = false

1 Assuming that the foaf:name and foaf:Person are associated /.well-known/ with FOAF vocabluary.

```
setOfProjection = ∅
extractGraph(g ∈ G)
foreach t ∈ T do
  if match(t, TP) and numberOfStatements < count(TP) then
    if isIRI(t.subject) then
      isCorrectSubject = true
    add to setOfProjection
    numberOfStatements =  numberOfStatements + 1
if isCorrectSubject = true then
  foreach r ∈ setOfProjection
    generate(r)
```

3. Related work

We can distinguish four main categories of ad hoc generating or importing RDF graph: based on XML [6, 7], based on SPARQL [8, 9], based on OWL [10] and views [11, 12].

Abiteboul et al. propose Active XML documents [6]. It may contain intensional subtrees defined through XQuery. While this approach may be sufficient for RDF/XML document, it is not satisfactory for other serializations. Another proposal, which allow import RDF triples, is XML Inclusions [7]. It is a mechanism for merging XML documents, by include documents or parts thereof. Unfortunately, it also can be dedicated only for RDF serializations based on XML.

Schenk et al. propose Networked Graphs [8]. It allows user do define RDF graphs by using SPARQL CONSTRUCT clause [9] and named graph model. While this approach may be sufficient for RDF graph stores, it is not satisfactory for other web services, which do not support SPARQL queries and named graphs. Additionally, this proposal do not use Linked Data and REST style.

Yet another approach is provided in [10]. *owl:imports* is a mechanism, which includes a whole OWL document to another one. Disadvantage of this approach is that it can be used to import parts of file and it support only static inclusions.

[11] and [12] concentrate to declarative view mechanisms. This solutions allow to define classes and properties based on graph patterns. Unfortunately, these approach are not oriented towards reuse and can not interlinked each other.

4. Conclusions

The problem of how to adjust generating temporal graphs to RDF sources has produced many proposals. Most of them are hard to use without dedicated tools, hence making the problem seem difficult.

We have produced a simple and thought-out proposal. We believe that our idea is an interesting approach, because it is independent and uses common standards.

Acknowledgements

The author would like to thank W3C WebID Incubator Group. Professor Henryk Rybinski's comments and support were invaluable.

References

1. Bizer C., Heath T., Berners-Lee T.: Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems (2009)
2. Fielding R. T., Taylor R. N.: Principled Design of the Modern Web Architecture, Transactions on Internet Technology (TOIT), Volume 2 Issue 2 (2002)
3. Klyne G., Carroll J. J.: Resource Description Framework (RDF): Concepts and Abstract Syntax. World Wide Web Consortium (2004)
4. Nottingham M., Hammer-Lahav E.: Defining Well-Known Uniform Resource Identifiers (URIs). Internet Engineering Task Force (2010)
5. Brickley D., Miller L.: FOAF Vocabulary Specification 0.98. FOAF Project (2010)
6. Abiteboul S., Benjelloun O., Milo T.: Positive active XML. PODS '04 Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (2004)
7. Marsh J., Orchard D., Veillard D.: XML Inclusions (XInclude) Version 1.0 (Second Edition). World Wide Web Consortium (2006)
8. Schenk S., Staab S.: Networked graphs: a declarative mechanism for SPARQL rules, SPARQL views and RDF data integration on the web. WWW '08 Proceeding of the 17th international conference on World Wide Web (2008)
9. Prud'hommeaux E, Seaborne A.: SPARQL Query Language for RDF. World Wide Web Consortium (2008)
10. Motik B., Patel-Schneider P. F., Parsia B.: OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. World Wide Web Consortium (2009)
11. Magkanaraki A., Tannen V., Vassilis Christophides V., Plexousakis D.: Viewing the Semantic Web through RVL Lenses. Lecture Notes in Computer Science (2003)
12. Volz R., Oberle D., Studer R.: Implementing Views for Light-Weight Web Ontologies. Seventh International Database Engineering and Applications Symposium (2003)
13. Crocker D., Overell P.: Augmented BNF for Syntax Specifications: ABNF. Internet Engineering Task Force (2008)

Appendix A. Grammar

In this section we present grammar of our proposal. We define it in Augmented Backus–Naur Form (ABNF) [13]:

*query = scheme "://" authority "/" pattern "/" *(pattern "/") ["?g=" graph]*
scheme = "http" / "https"
authority = host [":" port]
*host = *VCHAR*
*port = *DIGIT*
pattern = element "|" element "|" element
*element = *VCHAR*
*graph = *VCHAR*