NG4J – Named Graphs API for Jena

Chris Bizer
Freie Universität Berlin
Berlin, Germany
chris@bizer.de

Richard Cyganiak
Freie Universität Berlin
Berlin, Germany
richard@cyganiak.de

E. Rowland Watkins
University of Southampton
Southampton, UK
erw01r@ecs.soton.ac.uk

ABSTRACT

Named Graphs are a simple extension to the RDF abstract syntax and enable statements to be made about RDF graphs. This is beneficial for a variety of application areas like data syndication, provenance tracking, ontology versioning, signing RDF, and for specifying access rights to parts of RDF repositories. The Named Graph abstract syntax has been adopted by the W3C Data Access Working Group with a slight modification as the basis for SPARQL. In this poster we present NG4J – Named Graph API for Jena, an implementation of Named Graphs on top of the Jena Semantic Web Framework. NG4J provides APIs for manipulating sets of Named Graphs, TriX and TriG readers and writers for serializing multiple graphs in one document and an implementation of the TriQL query language. The SWP API provides methods for adding meta-information like propositional attitudes to graphs and for signing graphsets using digital signatures.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures, D.3.2 [**Programming Languages**]: Language Classifications – Java, I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods – *Representation languages.*

General Terms

Security, Languages, Theory.

Keywords

Named Graphs, TriX, TriG, TriQL, Digital Signatures, RDF, Semantic Web.

1.INTRODUCTION

The Semantic Web can be seen as a collection of RDF graphs. The RDF recommendations explain the meaning of any one graph, and how to merge a set of graphs into one, but do not provide suitable mechanisms for talking about graphs or relations between graphs. The ability to do so is required or beneficial for many applications, such as

- content syndication and provenance tracking,
- specifying access rights to or replacing and updating parts of an RDF repository,
- attaching meta-information about intellectual property rights or privacy preferences to published information,
- ontology versioning and evolution,
- signing published information,
- capturing logical relationships between graphs.

Demos and Posters of the 2nd European Semantic Web Conference (ESWC 2005), Heraklion, Greece, 29. May – 1. June 2005

Named Graphs [10] have been developed by members of the Semantic Web interest group as a simple extension to the RDF abstract syntax and as a foundation for these scenarios. A Named Graph is an RDF graph which is labelled by a URIref [5].

Current working drafts of the SPARQL query language [8] in development at the W3C Data Access Working Group are based on a similar notion. They define an *RDF Dataset* as a set of named graphs plus one distinct unlabelled 'background graph'.

NG4J – Named Graphs API for Jena [2] is an extension to the Jena Semantic Web Framework [9]. It provides support for named graphs to the popular Java-based toolkit [6], and thereby enables and simplifies many of the applications mentioned above.

By retrofitting Jena with an extended data model while staying compatible with the existing Jena API, we leverage the work of the Jena team and provide a migration path for existing applications based on Jena.

2. MANIPULATING GRAPH SETS

The basic information container in NG4J is the *NamedGraphSet.* It provides several ways to access graph sets and their members:

Graph View: NG4J's graph-centric API provides basic methods for manipulating graph sets. The members of an NG4J NamedGraphSet are Jena Graphs and can be manipulated using standard Jena functions for adding, removing and finding triples.

Quad View: For some applications, a quad-centric view is more practical. NG4J supports such a view. The first item in an NG4J quad is a graph selector; the other three constitute an RDF triple.

Provenance-enabled Jena Model View: Jena is a two-layered framework: a *Model* layer provides comprehensive functions for application developers; a low-level *Graph* layer allows straightforward implementation of custom data stores and inference engines and is less convenient for application development. NG4J works at the Graph layer but is able to present a Model view on a merge of all the graphs in a NamedGraphSet. This simplifies migration of existing Jena applications. When using this Jena Model view, the application developer can still take advantage of the underlying graph set: Provenance-enabled Jena Statements can be queried for their source graph name, and loading RDF documents into the Model will cleanly replace all statements previously loaded from the same URL.

3.SERIALIZATION AND PERSISTENCE

RDF serialization syntaxes like RDF/XML, N3 or Turtle provide a document-based storage mechanism for individual RDF graphs which can then be transmitted over a network. NG4J maintains compatibility with these syntaxes by serializing a graph set into a set of separate RDF documents. Sometimes it is desirable to encode several graphs within a single document, e.g. to exchange or archive SPARQL repositories or to publish RDF information together with provenance meta-information. For these purposes, NG4J provides parsers and serializers for two graph set serialization formats:

TriX [7] is a simple XML-based serialization syntax for both classic RDF and sets of Named Graphs. Unlike RDF/XML, TriX is a straightforward translation of the abstract syntax into XML, which allows for effective use of generic XML technologies such as XSLT and XQuery. An XSLT-based syntactic extension mechanism allows for more concise, more readable or otherwise different variations of the format.

TriG [3] is a compact and readable alternative to TriX. It is a variation of Turtle [1] with a notation that groups triples into multiple graphs.

Many Semantic Web applications have a need for persistent storage. NG4J provides a simple relational database-backed store. The store uses a denormalized data model for performance reasons. A set of Named Graphs is stored in a single quad table, while a separate graph name table allows for empty graphs. Currently, we support only MySQL.

4.QUERYING GRAPH SETS

Accessing RDF information through an API in a general-purpose programming language can result in repetitive, hard to maintain code. Typical access operations can be expressed more concisely in an RDF query language.

NG4J implements TriQL [4], a query language that extracts information from sets of Named Graphs, and is based on RDQL. TriQL uses graph patterns to query sets of Named Graphs. A graph pattern consists of an optional graph name and a set of triple patterns. Queries themselves are executed against all graphs in a graph set. Variable graph names help limit results to graphs that have certain properties or stand in a certain relationship to one another.

```
SELECT ?mbox, ?age
WHERE
   ex:aliceFoaf (?alice foaf:knows ?whom .
                 ?whom foaf:mbox ?mbox .
                 ?whom foaf:PersonalProfileDocument ?ppd)
           ?ppd (?w foaf:mbox ?mbox .
                 ?w foaf:age ?age)
```

The example finds people known by Alice who have published their age in their PersonalProfileDocument.

The TriQL query engine in NG4J evaluates queries against in-memory or database-backed sets of Named Graphs.

The SPARQL query language will likely also give us the ability to query across named graphs. TriQL predates SPARQL and we expect that SPARQL will supersede TriQL once SPARQL has become a final W3C recommendation.

5.PUBLISHING AND SIGNING GRAPHS

Information publication on the Semantic Web is one area which could benefit from Named Graphs. The Semantic Web Publishing (SWP) vocabulary provides terms that assert meta-information about graphs. Properties include an information provider's propositional attitude towards published information (e.g. are they the source of the information or did they merely quote from a third party?) and details of digital signatures used to sign Named Graphs [5]. NG4J implements convenience methods for using the SWP vocabulary. The API allows users to sign graphs and to verify signatures without the need for detailed knowledge about how digests and signatures are formed. The following example shows a TriG serialization of a signed graph set:

```
ex:graph1 {
    ex:Bob ex:foo ex:abc.
}
ex:graph2 {
    ex:Bob ex:foo ex:xyz.
}
<urn:uuid:ae526…> {
    ex:graph1 swp:digestMethod swp:JjcRdfC14N-sha1.
    ex:graph1 swp:digest "YjRhNzIy…"^^xsd:base64Binary.
    ex:graph1 swp:assertedBy <urn:uuid:ae526…>.
    ex:graph2 swp:digestMethod swp:JjcRdfC14N-sha1.
    ex:graph2 swp:digest "NmM2NWRk…"^^xsd:base64Binary.
    ex:graph2 swp:quotedBy <urn:uuid:ae526…>.
    <urn:uuid:ae526…> swp:authority ex:Alice.
    <urn:uuid:ae526…> swp:signatureMethod
        swp:JjcRdfC14N-rsa-sha1.
    <urn:uuid:ae526…> swp:signature
        "E2aylVAB…"^^xsd:base64Binary.
}
```

The current SWP implementation supports X.509 certificates and private keys in PKCS #12 keystores, although the SWP vocabulary supports PGP keys as well.

6.CONCLUSION

The Named Graphs abstract syntax, a simple extension of the RDF abstract syntax, has the potential to enhance the development of Semantic Web applications in several directions. Named Graphs together with the SWP vocabulary provide a formally defined foundation for a future Semantic Web trust layer. We hope that NG4J, as an implementation of Named Graphs for the Jena framework, encourages experimentation in areas of advanced Semantic Web development, such as provenance tracking, versioning, access control, privacy and trust.

NG4J is available under BSD license and can be downloaded from http://www.wiwiss.fu-berlin.de/suhl/bizer/ng4j/ .

7.REFERENCES

[1] Beckett, D. *Turtle – Terse RDF Triple Language* http://www.ilrt.bris.ac.uk/discovery/2004/01/turtle/

[2] Bizer, C., and Cyganiak, R. *NG4J – Named Graphs API for Jena.* http://www.wiwiss.fu-berlin.de/suhl/bizer/ng4j/

[3] Bizer, C. *The TriG Syntax.* http://www.wiwiss.fu-berlin.de/suhl/bizer/TriG/

[4] Bizer, C. *TriQL – A Query Language for Named Graphs.* http://www.wiwiss.fu-berlin.de/suhl/bizer/TriQL/

[5] Carroll, J., Bizer, C., Hayes, P., and Stickler, P. *Named Graphs, Provenance and Trust*. Fourteenth International World Wide Web Conference (WWW2005), 2005.

[6] Carroll, J., Dickinson, I., Dollin, C., Reynolds, D., Seaborne, A., Wilkinson, K. *Jena – Implementing the Semantic Web Recommendations*. In: Proceedings of the Thirteenth International World Wide Web Conference (WWW2004), 2004.

[7] Carroll, J., and Stickler, P. *RDF Triples in XML.* In: Extreme Markup Languages 2004 Proceedings, 2004.

[8] Prud'hommeaux, E., and Seaborne, A. *SPARQL Query Language for RDF*. W3C Working Draft. http://www.w3.org/TR/2005/WD-rdf-sparql-query-20050217/

[9] *Jena – A Semantic Web Framework for Java.* http://jena.sourceforge.net/

[10] *Named Graphs.* W3C Semantic Web Interest Group Website. http://www.w3.org/2004/03/trix/