Position paper: Named Graphs in Linked Data

Joshua Shinavier
RPI Tetherless World Constellation
Troy, New York
josh@fortytwo.net

ABSTRACT

Named graphs are likely to be at the forefront of any future revisions of the RDF data model. The ability to assert metadata about a graph is essential to provenance tracking, access control, and overall data management in the Semantic Web domain. If named graphs were to be incorporated into RDF, it follows that Linked Data would support named graphs at a basic level. This paper investigates potential publishing schemes for Linked Data which anticipate a higher degree of integration of named graphs with RDF, minimally extending Linked Data best practices in a backward compatible manner.

Keywords

Linked Data, Named Graphs, Semantic Web

1. INTRODUCTION

Named graphs are an extension of the RDF data model whereby RDF graphs are given a global identifier in the form of a URI. Although named graphs are not a part of the current version of RDF, it is widely recognized that they are necessary in practice. By enabling the expression of metadata about graphs, named graphs are essential to real-world applications which organize and manage RDF data.

What if named graphs were in fact included in the core syntax of RDF? What effect would this have on Linked Data? On the one hand, if it were possible to express arbitrary graphs and graph metadata in any Semantic Web document, the document-centric provenance schemes of current Linked Data clients would break down. On the other hand, it would be much more straightforward to publish data sets with diverse provenance – for example, collaborative knowledge bases and user-driven data from the Social Web – as Linked Data while preserving the provenance trail.

This paper explores this possibility, providing a simple extension to Linked Data best practices which enables the publication of arbitrary named graph data sets as Linked Data, conferring the advantages of dereferenceable HTTP URIs on graphs. The technique is described in terms of an existing named graph data model and existing formats, as these provide the most predictable route to a Semantic Web with built-in support for named graphs. The technique is compared and contrasted with an alternative technique which is less expressive but does not require nonstandard formats. Both techniques meet the following requirements:

- able to publish data with mixed assertional status. For example, third-party reviews of a product may be provided alongside authoritative information about the product, perhaps in the same document.
- backward compatible with deployed Linked Data, in that a client designed to consume Linked Data with named graphs can also consume Linked Data which does not make use of the named graph data model, nor of named graph formats
- includes existing Linked Data best practices

In addition, a fully conforming Linked Data publishing service is presented.

2. LINKED DATA

Linked Data is the practice of publishing RDF data on the Web in such a way that one can explore a Giant Global Graph[1] of human knowledge in a structured format. Linked Data is published according to four high-level principles:[2]

- URIs provide globally unique names for things
- HTTP URIs make it possible to *dereference*, or look up, those names on the Web
- when an HTTP URI is dereferenced, information is provided in a standard format
- links between URIs enable information discovery

In accordance with the third principle, Linked Data is generally provided in the RDF/XML [2] format, which is a W3C

[1] `http://dig.csail.mit.edu/breadcrumbs/node/215`
[2] `http://www.w3.org/DesignIssues/LinkedData.html`

recommendation. The other formats discussed in this paper are nonstandard.

At a lower level, Linked Data includes best practices [6] concerning vocabularies, content negotiation, minting of URIs, and so on.

3. NAMED GRAPHS

An *RDF graph* [12] is simply a set of RDF triples. Such graphs are a built-in feature of RDF. In addition, the notion of *named graphs* has arisen in various contexts in response to a need to refer to a set of triples as a whole. Named graphs are often used internally by tools which consume and manage RDF data.

For the purpose of this paper, named graphs are those defined by Carroll et al. [7]. This model of named graphs is a slight variation on the RDF data model, in which it is possible to refer to an RDF graph by means of a URI reference. This makes it possible to assert RDF statements about RDF graphs, with applications in security and access control, provenance tracking, versioning, and the expression of logical relationships among graphs.

In contrast to RDF graphs, a named graph is identified only by name, and is not necessarily a unique set of statements: two named graphs, with different URIs, may have equal or equivalent RDF graphs.

Named graphs are backward compatible with the RDF and OWL recommendations, and they are explicitly supported in the SPARQL query language [13].

3.1 Why not use reification?

It's tempting to use the RDF reification vocabulary as an alternative to named graphs; after all, so-called reified triples are a core feature of the Semantic Web. Unfortunately, the semantics of reified triples [10] are weaker than one might expect. A reified triple merely describes the relationship between a triple and its components, not to be confused with the quotation of a statement which can be accepted or rejected by a consumer. Even if reification were semantically stronger, the sheer number of triples required to represent a single reified statement would probably discourage most application developers from using it.

3.2 Serializing named graphs

There are several serialization formats for RDF which support named graphs to a greater or lesser extent.

3.2.1 TriX

TriX [8] is an XML-based format which was created as a simpler and generally more interoperable alternative to RDF/XML. Moreover, TriX adds the ability to name RDF graphs, either with global graph names in the form of URI references or with file scoped names in the form of blank node identifiers. It is worth noting that in the TriX proposal, a graph name identifies an RDF graph, or rather an equivalence class of RDF graphs, directly. This contrasts with the later named graphs proposal, in which it is possible for two distinct named graphs to share the same RDF graph. However, the close association of TriX with named graphs suggests that the original naming mechanism of TriX has been superseded by the intensional mechanism of named graphs. The ability to declare a graph as *asserted* or not asserted is built into the TriX syntax.

3.2.2 TriG

TriG [5] is an extension of the compact and human-friendly Turtle [1] RDF format which, like TriX, adds explicit support for named graphs. For the sake of simplicity, all examples in this paper use the TriG format, even when the data is understood to be published in RDF/XML. The triples in an RDF/XML document are considered to inhabit the default graph in the corresponding TriG serialization. TriG includes optional syntax elements for compatibility with the even more expressive Notation3 [4] format.

3.2.3 N-Quads

N-Quads [9] is an extension of the simple, line-delimited N-Triples RDF format, adding an *optional context value* for triples. While N-Quads is expressive enough for the serialization of named graphs, the meaning of context values is not defined: they may be URI references, blank node identifiers, or literals, and are not necessarily graph names.

4. NAMED GRAPHS IN LINKED DATA

From their inception, named graphs have been advanced as a tool for Semantic Web publishing. However, their inception predates the rise of Linked Data on the Web. While it may be true that named graphs are more commonly used for internal data management than they are for exchanging provenance metadata among applications, by using URIs – commonly HTTP URIs – to identify graphs, named graphs elevate provenance metadata to the level of things which can referred to, talked about, linked in the Semantic Web. An HTTP URI which cannot be dereferenced is not cool.[3]

A key question, then, is whether named graphs are *information resources* or *non-information resources*: the Semantic Web demands that we distinguish between the two. According to the Architecture of the World Wide Web [11], an information resource is one whose "essential characteristics can be conveyed in a message". Named graphs seem to fit that description, as they can be serialized using any of the formats described above.

It is common practice, in applications which consume Linked Data, to use the HTTP URI of a Semantic Web document as a graph name for the RDF triples encoded in the document. For example, the Tabulator [3] Linked Data browser and editor distinguishes between a concept layer and a document layer of navigation, maintaining detailed metadata about the documents describing Semantic Web concepts and presenting this metadata to the user as provenance information.

As noted in the TriX proposal, the OWL Ontology element and the OWL import mechanisms implicitly use document URLs as graph names, while it is noted that "this gives Named Graphs backward compatibility with RDF/XML, but has the disadvantage that retrieval URL, document name and graph name are mixed up."

[3] `http://www.w3.org/TR/cooluris/`

The position of this paper is that the status of named graph URIs on the Semantic Web is not yet universally agreed upon, and that it is worthwhile to explore both alternatives in the context of Linked Data. In the following, we will pose a problem of data publication in the presence of named graph metadata. One of the solutions will treat graphs as information resources and merely use the named graphs data model, while the other will treat graphs as non-information resources and make use of extended formats and best practices. The best choice in the long term is left as an open question.

5. A PROBLEM AND TWO SOLUTIONS

Suppose that we, as a data publisher, would like to publish the following named graphs as Linked Data:[4]

```
ns1:graph {
    ns0:rdfnext_3af5 rdfs:seeAlso
        <http://www.w3.org/2009/12/rdf-ws/> .
}

ns2:graph {
    ns0:rdfnext_bb13 ov:category
        <http://dbpedia.org/resource/Semantic_Web> .
}

{
    ns0:rdfnext owl:sameAs ns0:rdfnext_3af5,
        ns0:rdfnext_bb13 .

    ns1:graph a rdfg:Graph .

    ns1:post a sioc:Post ;
        sioc:has_creator ns3:eva ;
        sioc:embeds_knowledge ns1:graph .

    ns2:graph a rdfg:Graph .

    ns2:post a sioc:Post ;
        sioc:has_creator ns3:leon ;
        sioc:embeds_knowledge ns2:graph .
}
```

In this example, bloggers Eva and Leon have each made a statement about the #rdfnext resource. It is not our intention to assert the truth of these statements, but only to relay them to the information consumer in the hope that they are useful, annotating their named graphs with provenance information to help the consumer decide whether or not to accept them. The default named graph, on the other hand, contains first-hand information which we have asserted about the data sources. In particular, we have connected each of the third-party graphs with its respective blog post by means of the `sioc:embeds_knowledge` property.

In order to publish the above as Linked Data, we have a number of options. One of these is simply to throw away the graph names and the associated provenance information in order to express the information in RDF/XML:

```
{
    ns0:rdfnext_3af5 rdfs:seeAlso
        <http://www.w3.org/2009/12/rdf-ws/> .

    ns0:rdfnext_bb13 ov:category
        <http://dbpedia.org/resource/Semantic_Web> .

    ns0:rdfnext owl:sameAs ns0:rdfnext_3af5,
        ns0:rdfnext_bb13 .
}
```

Clearly, this is not acceptable if the provenance information is important. The following provides a better option.

5.1 Solution #1: identify graphs with documents

Rather than dropping the provenance metadata, we will serve information with different provenance in different documents. It is for this purpose that the resource identified by `ns0:rdfnext` has been given two “alternative” URIs in addition to the single “hub” URI. Third party information about the resource is served against the alternative URIs in two separate documents: one for each graph. In the document identified by `ns1:graph`:

```
{
    ns0:rdfnext_3af5 rdfs:seeAlso
        <http://www.w3.org/2009/12/rdf-ws/> .
}
```

and in the document identified by `ns2:graph`:

```
{
    ns0:rdfnext_bb13 ov:category
        <http://dbpedia.org/resource/Semantic_Web> .
}
```

Provenance metadata about the graphs must be asserted externally to the documents themselves. It is served against the hub URI of each resource which is the dereferenceable subject or object of a statement in the graphs:

```
{
    ns0:rdfnext owl:sameAs ns0:rdfnext_3af5,
        ns0:rdfnext_bb13 .

    ns1:graph a rdfg:Graph .
    ns1:post sioc:embeds_knowledge ns1:graph .

    ns2:graph a rdfg:Graph .
    ns2:post sioc:embeds_knowledge ns2:graph .
 }
```

Finally, first-hand information about the blog posts is published with links to the non-asserted graphs:

```
{
```

[4]All URIs for “internal” resources are presented as QNames, while namespace prefixes are omitted. The use of owl:sameAs reflects a common pattern.

```
        ns1:post a sioc:Post ;
            sioc:has_creator ns3:eva ;
            sioc:embeds_knowledge ns1:graph .
    }
```

and

```
    {
        ns2:post a sioc:Post ;
            sioc:has_creator ns3:leon ;
            sioc:embeds_knowledge ns2:graph .
    }
```

If we restrict ourselves to publishing this data set in RDF/XML, then the above is probably the best solution. Nonetheless, it has some conceptual and practical drawbacks:

- It involves significant data duplication, in that all statements in which the graph appears as subject or object (in the example, the `sioc:embeds_knowledge` statements and the `rdf:type` statements of the graphs) must be duplicated in the associated description of every resource which is described in the graph. This is in addition to the usual URI rewriting and other transformations which are typically required to publish monolithic data sets as Linked Data.
- It requires named graph resources to be described in documents for which the graph URI is neither the dereferenceable subject nor object. This is not merely an attempt to avoid the semantic difficulties of graphs which describe themselves: serving provenance metadata against the URIs of the graphs themselves would result in the injection of statements not present in the original graphs, potentially invalidating the same metadata.
- Graphs about graphs quickly become unmanageable. For example, if a statement about `ns1:graph` were to be included in a third graph `ns4:graph`, the information in `ns4:graph` would not be discoverable in general, unless its metadata were replicated in every document in which `ns1:graph` is mentioned.

5.2 Solution #2: explicit named graphs

The above solution took advantage of the data model of named graphs without making use of named graph formats. In the following, we will do both.

The solution below has the advantage that the published Linked Data is exactly the original named graph data, provided that the URIs are in the right URI space. It also affords unlimited expressivity, in terms of graphs which describe other graphs.

If we use hash URIs for each of the namespaces, publishing the data becomes particularly simple and efficient for client and server. First-hand metadata about resources, as well as third-party metadata, are served against the hub URI:

```
    ns1:graph {
        ns0:rdfnext_3af5 rdfs:seeAlso
            <http://www.w3.org/2009/12/rdf-ws/> .
    }

    ns2:graph {
        ns0:rdfnext_bb13 ov:category
            <http://dbpedia.org/resource/Semantic_Web> .
    }

    {
        ns0:rdfnext owl:sameAs ns0:rdfnext_3af5,
            ns0:rdfnext_bb13 .
    }
```

Since there's a one-to-one correspondence between graphs and blog posts in this example, we can even describe them in the same documents (otherwise, we would have one for each):

```
    {
        ns1:graph a rdfg:Graph .

        ns1:post a sioc:Post ;
            sioc:has_creator ns3:eva ;
            sioc:embeds_knowledge ns1:graph .
    }
```

and:

```
    {
        ns2:graph a rdfg:Graph .

        ns2:post a sioc:Post ;
            sioc:has_creator ns3:leon ;
            sioc:embeds_knowledge ns2:graph .
    }
```

The immediate disadvantages of this approach are that it requires non-standard extensions, and that its treatment of named graphs as non-information resources is debatable. It also raises the question of whether any given named graph serialized in a document is *complete*: named graphs "[fix] the graph according to the name in a rigid, non-extensible way", although this is perhaps an arbitrary design decision. If a graph truly cannot be named in more than one document, then some graphs need to be split into two: one graph for the forward links of a resource, and another for backlinks. On the other hand, if we treat graphs with the open world assumption, never assuming that the full extent of a graph is known, then there is no need for metadata duplication.

6. EXTENDING BEST PRACTICES

Solution #2 above illustrates three additional rules for publishing provenance-aware Linked Data on the Web, which we will now make explicit:

- provide the data in a quads format (TriX, TriG, or N-Quads)
- coin dereferenceable URIs for named graphs and serve named graph metadata against the URI of its graph

- assert authoritative metadata about a resource in the default graph of its associated description

Otherwise, follow current Linked Data best practices. The third principle builds on the named graph notion of asserted vs. non-asserted graphs. The named graphs proposal includes a rich vocabulary for expressing *propositional attitudes* such as asserting or denying a graph, commenting upon it, and so on. However, most of the information on the Semantic Web has an implicit, default attitude: it is asserted to be true. As a simple way of migrating from existing RDF/XML Linked Data to Linked Data in named graph formats, it is reasonable to assume that everything not named by a graph URI (that is, in the default graph) is asserted to be true by the data publisher. Whether a client accepts or rejects the data depends on its trust in the publisher.

6.1 Consuming Named Graphs in Linked Data

The main purpose of this paper is to discuss alternatives for publishing Linked Data with named graphs. However, something must also be said about consuming that data. Current Linked Data clients are not prepared to handle documents containing more than one graph. For that reason, data published under scheme #1 would be much easier for existing clients to consume; the only problem is in knowing which documents to trust. This requires an understanding of the named graphs data model and of the provenance vocabulary used to publish the data. Semantic Web documents with explicit named graphs are more expressive but additionally require the client to distinguish between graphs, which are relevant to provenance, and documents, which are relevant to caching and network operations. Otherwise, many issues of managing provenance of Linked Data from the Web are the same as those of integrating generic named graph data sets.

7. AN IMPLEMENTATION

The ideas discussed in this paper have been implemented in the TwitLogic [14] semantic data aggregator. TwitLogic is a real-time service which maintains a connection to Twitter's streaming API,[5] extracting structured information provided by Twitter users and immediately publishing it as Linked Data under an open license. Author attribution metadata is maintained for every statement extracted from the Twitter stream, making the provenance of the data set thoroughly diverse. TwitLogic currently supports both of the alternative publishing schemes described above: one in the RDF/XML, Turtle and N-Triples formats, and the other in the TriG and TriX formats.

8. CONCLUSION AND FUTURE WORK

This paper has discussed the possibility of built-in support for named graphs in Linked Data and explored two data publishing patterns which take advantage of it. The choice of an appropriate pattern depends on the degree to which named graphs may describe each other, and on the almost philosophical question of whether a named graph is an information resource. This paper is intended to stimulate discussion. However, a more thorough and more formal treatment of graph naming in Linked Data would be worthwhile.

While no Linked Data client tool exists to take advantage of these patterns, such support may be added to the LinkedDataSail[6] client, depending on the outcome of any discussion this work generates. Additionally, the data publishing component of TwitLogic is suitable to be made into a reusable tool.

9. ACKNOWLEDGEMENTS

This work has been supported by the Tetherless World Constellation.[7] Thanks to Tom Heath for helpful comments on an early draft of this paper.

10. REFERENCES

[1] D. Beckett and T. Berners-Lee. Turtle-terse RDF triple language. *W3C Submission, Jan*, 2008.

[2] D. Beckett and B. McBride. RDF/XML syntax specification (revised). *W3C Recommendation*, 10, 2004.

[3] T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets. Tabulator: Exploring and analyzing Linked Data on the Semantic Web. In *In Procedings of the 3rd International Semantic Web User Interaction Workshop (SWUI06)*, 2006.

[4] T. Berners-Lee and D. Connolly. Notation 3 (N3) A readable RDF syntax. *http://www.w3.org/DesignIssues/Notation3*, 1998.

[5] C. Bizer and R. Cyganiak. The TriG syntax. *http://sites.wiwiss.fu-berlin.de/suhl/bizer/TriG/Spec/TriG-20070730*, 2007.

[6] C. Bizer, R. Cyganiak, and T. Heath. How to publish Linked Data on the Web. http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/.

[7] J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named graphs. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(4):247–267, 2005.

[8] J. Carroll and P. Stickler. RDF triples in XML. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, page 413. ACM, 2004.

[9] R. Cyganiak, A. Harth, and A. Hogan. N-quads: Extending n-triples with context. *http://sw.deri.org/2008/07/n-quads/*, 2008.

[10] P. Hayes. RDF Semantics. W3C Recommendation, 10 February 2004. *World Wide Web Consortium*, 2004.

[11] I. Jacobs and N. Walsh. Architecture of the World Wide Web. *W3C Recommendation*, December 2004.

[12] G. Klyne, J. Carroll, and B. McBride. Resource description framework (RDF): Concepts and abstract syntax. *W3C recommendation*, 10, 2004.

[13] A. Seaborne and E. Prud'hommeaux. SPARQL query language for RDF. *W3C recommendation, W3C (January 2008) http://www. w3. org/TR/2008/REC-rdf-sparql-query-20080115*, 2008.

[14] J. Shinavier. Real-time #SemanticWeb in <= 140 chars. In *Linked Data on the Web (LDOW2010)*, 2010.

[5] http://apiwiki.twitter.com/Streaming-API-Documentation

[6] http://github.com/joshsh/linked-data-sail

[7] http://tw.rpi.edu