RDF Triples in XML

Jeremy J. Carroll
Hewlett-Packard Labs
Bristol, BS34 12QZ
UK
jjc@hpl.hp.com

Patrick Stickler
Nokia
Hatanpäänkatu 1
33900 Tampere
Finland
Patrick.Stickler@nokia.com

ABSTRACT

RDF/XML does not layer RDF on top of XML in a useful way. We use a simple direct representation of the RDF abstract syntax in XML. We add the ability to name graphs, noting that in practice this is already widely used. We use XSLT as a general syntactic extensibility mechanism to provide human friendly macros for our syntax. This provides a simple serialization solving a persistent problem in the Semantic Web.

Keywords

Semantic Web, RDF, XML

1. INTRODUCTION

This poster presents a new minimalist XML syntax for RDF. It is called TriX , and is described in more detail in [3].

1.1 The Requirements

The requirements we choose to address are: the format serializes the RDF graph; the format is compatible with XML tools; and as few other features are included as possible.

One of the problems with RDF syntax is an excess of requirements from different communities creating a political problem that may get solved with a technical hack.

We have two additional features: naming of graphs and syntactic extensibility. Graph naming is an extension to the RDF abstract syntax. They do not reflect the needs of any specific community, but meet general requirements of many RDF users.

1.2 Examples

Example 1: Here is a TriX document:

```
<trix xmlns="http://www.w3.org/2004/03/trix/trix-1/">
<graph>
   <uri>http://example.org/Graph1</uri>
   <triple>
      <uri>http://example.org/Bob</uri>
      <uri>http://example.org/wife</uri>
      <uri>http://example.org/Mary</uri>
   </triple>
   <triple>
      <uri>http://example.org/Mary</uri>
      <uri>http://example.org/age</uri>
      <typedLiteral datatype=
    "http://www.w3.org/2001/XMLSchema#integer"
      >32</typedLiteral>
   </triple>
</graph>
</trix>
```

Syntactic extensions to the minimalist core, require a processing instruction. **Example 2** is the same graph expressed using qnames and XSD type support:

```
<?xml-stylesheet type="text/xml" href=
 "http://www.w3.org/2004/03/trix/ext/all.xsl" ?>

<trix xmlns:eg="http://example.org/"
   xmlns="http://www.w3.org/2004/03/trix/trix-1/" >
<graph>
   <qname> eg:Graph1 </qname>
   <triple>
      <qname> eg:Bob </qname>
      <qname> eg:wife </qname>
      <qname> eg:Mary </qname>
   </triple>
   <triple>
      <qname> eg:Mary </qname>
      <qname> eg:age </qname>
      <integer> 32 </integer>
   </triple>
</graph>
</trix>
```

2. RDF/XML REVISED, BUT NOT FIXED

A historical survey of XML syntaxes for RDF can be found in [3]. This takes as from Guha's MCF [5] through to the RDF/XML Syntax (Revised) [1].

While many syntactic problems have been fixed, some of the 'postponed issues' [7] indicate the extent of the original mess: e.g. 'RDF embedded in XHTML and other XML documents is hard [i.e. impossible] to validate.' and 'it is not possible to define [. . .] a subset [of RDF/XML] that [. . .] can represent all [. . .] RDF graphs [and] can be described by an DTD or an XML Schema'.

In brief, RDF/XML does not layer RDF on top of XML in a useful way.

Meanwhile, there are other unresolved syntactic issues, involving qnames, collections, literals as subjects, blank nodes as predicates, reification and quoting. Hence, a further round of work on RDF/XML is likely to be a continuation of legacy hell, with additional requirements pulling in different directions, and old requirements not getting dropped.

However, many communities find that while RDF/XML has many features they do not like, certain key features, such as the use of qnames, or the collection syntax, are highly attractive and keep them enagaged.

3. TriX SYNTAX

The core of TriX is the `triple` element, which contains three children, the subject, predicate and object of the triple. Each of these children is either a `uri` element, an `id` element (for blank nodes), a `plainLiteral` or a `typedLiteral` element.

WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

The root element of the document is a `graphset` element, which has zero or more `graphs` as its child elements.

TRIX is described by a DTD, shown in table 1 and by an XML Schema. This format is very close to the RDF abstract syntax [6], the only deviation being the ability to name graphs.

```
<!-- TriX: RDF Triples in XML -->
<!ELEMENT trix         (graph*)>
<!ATTLIST trix         xmlns    CDATA
          #FIXED "http://www.w3.org/2004/03/trix/trix-1/">
<!ELEMENT graph        (uri, triple*)>
<!ELEMENT triple       ((id|uri), uri,
        (id|uri|plainLiteral|typedLiteral))>
<!ELEMENT id           (#PCDATA)>
<!ELEMENT uri          (#PCDATA)>
<!ELEMENT plainLiteral (#PCDATA)>
<!ATTLIST plainLiteral xml:lang CDATA #IMPLIED>
<!ELEMENT typedLiteral (#PCDATA)>
<!ATTLIST typedLiteral datatype CDATA #REQUIRED>
```

Table 1: TRIX DTD

4. NAMING GRAPHS

TRIX provides for global graph naming by the use of a `uri` element before the triples of a graph.

The requirement for graph naming, is not from one community within the Semantic Web, but a requirement that goes across the board. It is needed for metadata repositories, and for ontological systems. Graph naming occurs in Semantic Web programming environments and query languages. Nearly all users of the Semantic Web name their graphs, the base syntax should provide explicit support. Current solutions are muddled and *ad hoc*. A standardized approach will be highly beneficial.

example 3 shows a named graph, with provenance information in a separate graph.

```
<trix
   xmlns="http://www.w3.org/2004/03/trix/trix-1/">

<graph>
   <uri>http://example.org/Graph2</uri>
   <triple>
      <uri>http://example.org/aBook</uri>
      <uri>http://purl.org/dc/elements/1.1/title</uri>
      <typedLiteral datatype=
  "http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral"
      >&lt;ex:title xmlns:ex="http://example.org/"&gt;
          A Good Book
       &lt;/ex:title&gt;</typedLiteral>
   </triple>
</graph>
<graph>
   <uri>http://example.org/Graph3</uri>
   <triple>
      <uri>http://example.org/Graph2</uri>
      <uri>http://example.org/source</uri>
      <uri>http://example.org/book-description.rdf</uri>
   </triple>
</graph>

</trix>
```

The use of named graphs for provenance, and how to indicate the intended assertional status, which may differ between the two graphs in the example, is explored in depth in [2].

5. EXTENSIBILITY

Requirements related to ease of writing and reading an XML syntax for RDF tend, in general, to conflict with the core requirements of giving a transparent representation of the graph in a way that can easily be processed with XML tools. This is because the RDF graph tends to be too fine-grained and detailed for direct human consumption, and user-friendly syntaxes need to use 'macros' of some sort. In RDF/XML macros are provided for typed nodes, property URIs, property attributes, three parseTypes, striping, reification and container membership. These macros then create problems for XML tools.

The answer we suggest is to use XSLT as a general purpose and interoperable extensibility mechanism. This can use the stylesheet processing instruction, [4]. Each community can then define and use whatever syntactic extensions they wish, declaring the extensions they are using at the top of the data files. Example 2 shows how it can be used, where the referenced transform expands qnames into URIs and expands typed literals. Further transforms might permit the use of `xml:base`, or implement `parseType="Literal"` or `parseType="Collection"`. In fact, using XSLT2 the whole of RDF/XML could be implemented as a transform into TRIX providing backward compatibility.

With such a web based approach to syntactic extensibility anyone can define their own extensions. Those that are useful will be used; those that are not, will not. This will form an evolutionary system for designing useful XML serializations for RDF.

6. CONCLUSIONS

The problem of how to serialize RDF in XML has produced many proposals. Most, particularly RDF/XML, obscure the nature of the RDF graph, hence making the problem seem difficult. Despite the revision of RDF/XML, discussions continue.

With little difficulty, we have produced a thought-out and simple proposal. We suggest that it is time that the Semantic Web community choose a simple serialization such as ours, and stopped wasting time with this problem.

The use of XSLT as an extensibility mechanism permits the inevitably rather unreadable machine-friendly syntax to be represented in a more human-friendly fashion. It also permits backward compatibility with RDF/XML.

Naming graphs is a necessary part of the Semantic Web, and should be included in the core syntax. Further work on the semantics and abstract syntax of graph naming is available [2].

7. REFERENCES

[1] D. Beckett. RDF/XML Syntax Specification (Revised). W3C Rec., 2004.

[2] J. J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named Graphs, Provenance and Trust. Technical Report HPL-2004-57, HP Labs, 2004.

[3] J. J. Carroll and P. Stickler. TriX: RDF Triples in XML. Technical Report HPL-2004-56, HP Labs, 2004.

[4] J. Clark. Associating Style Sheets with XML documents Version 1.0. W3C Rec., 1999.

[5] R. Guha and T. Bray. Meta Content Framework Using XML . `http://www.w3.org/TR/NOTE-MCF-XML-970624/`, 1997.

[6] G. Klyne and J. Carroll. RDF Concepts and Abstract Syntax. W3C Rec., 2004.

[7] B. McBride. RDF Issue Tracking. `http://www.w3.org/2000/03/rdf-tracking/`, 2003.