

# New Syntaxes for RDF

David Beckett  
Institute for Learning and Research Technology (ILRT)  
University of Bristol  
8-10 Berkeley Square, Bristol  
BS8 1HH, UK  
dave.beckett@bristol.ac.uk

## ABSTRACT

This paper reviews syntaxes for RDF as defined in *RDF Model and Syntax W3C Recommendation* including RDF/XML as updated by the *RDF/XML Syntax Specification (Revised)* and describes the problems that remain after the revising. These include not clearly showing the RDF triple model and not working very well with newer XML technology such as XSLT and *W3C XML Schema (WXS)*.

The paper then constructs requirements for new syntaxes in the two main uses – as a transfer syntax as an end user syntax. It summarises existing approaches and discusses using XML or non-XML formats and then describes two new syntaxes, an outline XML one and a new textual RDF syntax *N-Triples Plus* based on the *N-Triples* test case syntax.

## Categories and Subject Descriptors

D.3.1 [Formal Definitions and Theory]: Syntax; I.7.2 [Document Preparation]: Markup languages

## General Terms

Resource Description Framework, Extensible Markup Language

## Keywords

RDF, XML

## 1. INTRODUCTION TO RDF/XML

RDF was first defined by the W3C's *RDF Model and Syntax W3C Recommendation* [21] (M&S) in February 1999. This included a recommended syntax called RDF/XML that was designed for a variety of goals by the RDF working group including enabling it to be embedded in HTML (before XHTML existed) in order to describe web pages, with a frame-style syntax and using XML QNames in order to shorten the long URIs that RDF uses for its terms. Namespaces in XML [7] specification was developed in parallel with RDF, and RDF was one of the first W3C specifications to use it.

Figure 1 shows some RDF/XML from M&S for the sentence *Ora Lassila is the creator of the resource <http://www.w3.org/Home/Lassila>*.

The format begins with an outer `rdf:RDF` XML document element. Contained within is an `rdf:Description` element for a “frame-style” block of properties, all about the resource with the URI `http://www.w3.org/Home/Lassila`. The element

Copyright is held by the author/owner(s).  
WWW2004, May 17–22, 2004, New York, NY USA.  
ACM xxx.xxx.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://description.org/schema/">
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

Figure 1: Example RDF/XML from the RDF Model and Syntax Specification (1999)

`s:Creator` encodes the property with the value “Ora Lassila”. This element name gives a URI reference from the namespace name (URI) for “s” which in this case is `http://description.org/schema/` concatenated with the element’s local name `Creator` giving the URI `http://description.org/schema/Creator`.

When a triple has a URI object, an `rdf:resource` attribute is used on an empty property element with the URI as the attribute value. An RDF literal can also have an XML language, given with an `xml:lang` attribute and can have an XML content when the `parseType="Literal"` attribute is used on the property element.

It was a goal to allow embedding in HTML such that common web browsers would ignore them; this can be done when there is no visible element content (CDATA). RDF/XML handled this case by defining alternate forms including writing properties with literal content as XML attributes in what was called the *Basic Abbreviated Syntax*.

There are several other abbreviations both to make the resulting RDF/XML more compact and to allow the omission of description blocks. Several common RDF vocabulary terms had special support such as the `rdf:type` property and the reification vocabulary. RDF containers – ordered, unordered or an alternative of list of resources – have a syntax form to provide easy generation of the container membership properties.

There were three syntax forms for distributed description of triples. The `aboutEach` and `aboutEachPrefix` attributes allowed the triples to be given about multiple resources in a container (the former) or about all resources with a URI of a certain prefix (the latter). The `bagID` attribute allowed descriptions of the collection of triples given in one of the frame-style descriptions using RDF reification.

The RDF/XML syntax was defined by an extended BNF in a formal grammar along with descriptive text in several sections of the document. The use of namespaced elements and attributes meant that using a DTD to define it was not possible and this was before modern XML schema language standardisation work was started

so there was no *W3C XML Schema (WXS)*[16] or Relax NG[10] etc. available.

## 2. REVISED RDF/XML

In 2001, The W3C RDF Core working group (RDF Core) started updating the RDF specifications including revising the XML syntax in terms of design and its specification. *RDF/XML Syntax Specification (Revised)*[1] now redefines and explain the XML syntax separate from RDF concepts and semantics. The revised syntax removed the distributed referents – aboutEach, aboutEachPrefix and bagID which had little use in the community, did not all have corresponding concepts in the RDF model and were difficult to use, especially in combination. This results in RDF/XML syntax being more clearly a triple-encoding format rather than with a mixture of quantification with unclear scope. The revision also added support for a set of new requirements: datatyped literals, explicit blank node identifiers and a resource collection syntax. The latter has been used by the Web Ontology Language (OWL)[23] for describing closed sets of terms. There were also some other minor changes and clarifications. An example of the revised syntax showing the collections support is given in Figure 2

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://example.org/stuff/1.0/"
  xml:base="http://example.org/fruit/">

  <ex:Basket rdf:about="http://example.org/item1">
    <ex:hasFruit rdf:parseType="Collection">
      <rdf:Description rdf:about="peach"/>
      <rdf:Description rdf:about="apple"/>
      <rdf:Description rdf:about="pear"/>
    </ex:hasFruit>
  </ex:Basket>
</rdf:RDF>
```

Figure 2: An RDF/XML revised example using collections

The new syntax specification is supported with machine readable testcases giving mappings from RDF/XML to RDF triples. This is written in a new test case format called *N-Triples* defined in the *RDF Test Cases*[18] working draft. N-Triples also enabled discussions of the abstract syntax separate from XML detail issues and was then used further in the semantics draft. This format was not intended as a new user syntax and is entirely regular, providing no alternative forms. N-Triples is discussed further in Section 6.

## 3. REMAINING PROBLEMS OF RDF/XML

Comments on the RDF M&S and later work were received from the community as feedback and recorded on the *RDF Core Working Group Issue List*<sup>1</sup>. Not all of these were possible to address during the syntax revisioning without inventing a new syntax which was out of scope for the working group. The major remaining problems are as follows:

1. One cannot tell an RDF node element/property element by simple inspection of the element in question without knowing the “striping” (after Brickley[8]).
2. The frame-style approach does not clearly match the triples in the RDF graph.
3. There are excessive choices in writing RDF/XML.

<sup>1</sup><http://www.w3.org/2000/03/rdf-tracking/>

4. Elements, attributes and attribute values are used for the same purposes, for example, encoding a URI reference.
5. The way that XML QNames are used does not constrain the elements and attributes that can appear in RDF/XML.
6. The unconstrained syntax cannot be described completely with XML schema languages such as DTDs and WXS.
7. It does not allow using  `xsi:type`  for specifying W3C XML Schema datatypes.
8. The syntax is not easy to use with XML technologies such as XSLT, XQuery and other XML tools.
9. It is impossible to embed in XHTML while retaining DTD validation (also true with any other XML syntax).
10. It is hard to emit human-readable RDF/XML from an RDF graph due to the range of choices (after Carroll[9]).
11. It cannot describe collections of literals.
12. Not all property URIs can be encoded.
13. Various aesthetic criticisms have been levelled at the syntax such as being “ugly”.

## 4. RDF NEW SYNTAXES REQUIREMENTS

There are two general classes of syntax that have been identified from the existing development of RDF/XML and discussion with other communities:

1. A canonical syntax that clearly represents RDF triples.
2. A syntax that is intended to be easy to author and read.

These have such different targets that they may not be met by a single syntax since the former tends to suggest minimal use of user-friendly forms and the latter would tend to have “syntactic sugar” to enable both common and complex RDF triple structures to be written concisely. A single syntax may work poorly at both jobs and remain inappropriate for both which is not much of an improvement over the current state. It is not even clear that a single end user syntax can satisfy the different needs of end user communities. These may benefit from their own XML form mapping to a canonical one via XSLT, if the target XML was suitable for that.

The requirements for a future syntax come from the problem reports on the existing syntax, experience from issues that emerged during the revision of RDF/XML, comments on the new syntax working drafts and also recorded issues on RDF Core’s postponed issue list. These were mostly postponed due to it being out of scope of the group’s charter. The following sections contain the requirements grouped into approximate categories.

### 4.1 Critical requirements

These requirements come from the lessons learnt from the current syntax and feedback and must be satisfied. The problems enumerated in Section 3 are given where associated with a requirement. Any new RDF syntax must:

- Be able to encode all legal RDF graphs. (Problems 11, 12)
- Clearly map to and from the triples abstract syntax. (Problems 1, 2, 10)
- Use a minimal number of alternate forms. (Problem 3)

## 4.2 XML design requirements

These came as advice from the XML community and W3C XML working groups on how to modernise the XML to current best practice and make it easier to work with using other XML technologies and tools. An RDF syntax expressed in XML should:

- Use a small set of XML tags. (Problems 5, 6)
- Not mix the use of elements and attributes for the same purpose.
- Be a “modern” XML syntax – such as using XML QNames in attribute values.<sup>2</sup> (Problems 4, 13)
- Permit W3C XML Schema datatypes in the instance data using `xsi:type`. (Problem 7)
- Make it easy to generate and manipulate with XSLT, XQuery and XPath. (Problem 8)

## 4.3 Syntax conveniences requirements

Any RDF syntax intended for hand-production by end users should provide:

- A short form for complex things such containers and collections.
- A form for collections of literals. (Problem 11)
- A way to embed in XHTML while retaining validation. (Problem 9)
- A more convenient way to express reification.

## 4.4 Extended RDF model requirements

These are not immediate requirements but the lack of an easy way to do these as modifications to RDF/XML limited RDF Core from making changes such as these to the RDF model. It would be forward-looking if a syntax provided support for an extended RDF that allowed:

- Literal subjects.
- Blank nodes as property labels.
- The explicit delineation of subgraphs (sometimes called *contexts*) and associated provenance
- The expression of formulae, rules, and other concepts which form higher layers of “the semantic web picture”

## 4.5 Conflicting requirements

The parts of RDF/XML that made embedding in non-validated HTML possible are also those that make up the excessive number of alternate forms (for example, all property attributes of RDF/XML could be removed and the syntax would be able to represent all the same graphs as at present). This means that a design for embedding in this way would clash with a minimal design. However, in this case, a design for embedding in XHTML would require DTD or WXS validation via using XHTML Modularization so an approach similar to RDF/XML would not be possible. More detailed discussion of these problems will be given in Section 6.

<sup>2</sup>Although this isn’t friendly to all XML technologies such as XML Canonicalization, XSLT – the namespace bloat problem.

## 5. EXISTING PROPOSALS

There have been several proposals for new syntaxes for RDF, both aimed as canonical syntaxes, end-user syntaxes and a combination thereof. These have included proposals to add or remove functionality to RDF/XML or HTML to make embedding RDF more convenient, entirely new XML syntaxes, using existing XML technologies to define a transfer encoding and also non-XML proposals aimed at making things easier to write. Approaches using DTD with RDF/XML have been possible but only when the terms in use in the application are limited to a constrained set[2].

It is clear that RDF/XML has already too many options in the ways to encode RDF graphs (although some people have proposed more). So a true subset of RDF/XML could be used as a recommended, minimal form. This is the approach used by Adobe’s XMP<sup>3</sup> which encodes a profile of RDF/XML inside several formats (PDF, TIFF, JPG, PNG, HTML and others) to describe the content. Seven items were removed or changed from RDF/XML – `rdf:RDF` was mandated and `rdf:parseType="Literal"`, top-level containers, `rdf:ID`, `rdf:bagID`, `rdf:aboutEach` and `rdf:aboutEachprefix` were forbidden. This smaller profile has been called “RDF/XML-7” and has been successfully deployed with many Adobe products. This subset remains compatible with the revisions since the last three of the seven XMP forbid were removed from the syntax.

In [4] Berners-Lee considered another subset of RDF/XML but without the node/property element striping, the key part of its formation. This led to a rather complex set of additions in order to declare the current subject of the triple. It has not been updated in light of later RDF/XML developments and does not seem a fruitful approach to pursue.

XML has a linking technology XLink[15] and a way to point to parts of XML documents (XPointer) that could be used to encode a graph similar to RDF. This was recognised early on in the design of these technologies – whereas RDF has links built in, XML has linking added outside the core. Daniel[14] described a possible mapping from XML using XLink to RDF triples. This has been most recently considered as part of ongoing work of the W3C Technical Architecture Group (TAG) who been considering the kind of document that might live behind an XML namespace URI. This document potentially could link to several other resources such as style sheets, schemas and RDF descriptions. The current best proposal RDDDL[5] by Borden and Bray is defined as a profile of XHTML 1.0 Basic adding two attributes, but can be considered straightforwardly as RDF triples relating the namespace URI to other resources. It is not a proposal for a general RDF syntax.

Berners-Lee’s *Notation 3 (N3)*[3] (2000-) is a “an academic exercise in language designed for a human-readable and scribbleable[sic] language”. The N3 language and its primary implementation *CWM* describe a research language that includes functionality outside the RDF model. The syntax defines a text format using a BNF-like grammar that uses a lot of punctuation to abbreviate the RDF. Each RDF triple can be given as a set of three terms explicitly or abbreviated in a variety of forms using a form that operates like XML QNames in RDF/XML. Declarations are allowed starting with @ using @*prefix* to give namespace URIs a short prefix. There are also parts that go beyond RDF to enable gather a set of statements, adding variables and scoping them to a set. The language is closely tied to the *CWM* code which makes it very useful for semantic web experiments in logic, rules and beyond RDF triples. These tend to make it not completely suitable as a language to meet problems and requirements for a new syntax.

<sup>3</sup><http://www.adobe.com/products/xmp/>

RDF Core designed N-Triples (described in *RDF Test Cases*[18]) as a true subset of N3, with no abbreviated forms allowed. This restriction and the resulting regularity and simplicity meant that it was a format that was easy to generate and understand as well as being remaining usable by existing N3 tools. It has proved very practical to use in dealing with RDF test case descriptions. There are both advantages and disadvantages of using non-XML formats which are discussed in more detail in the next Section 6.

A more recent strawman proposal for a new XML format was Bray's *RPV*[6] "designed to be entirely unambiguous and highly human-readable." It takes a strong resource-centred approach describing a particular resource with the properties and values parts of the RDF triple very clearly written, using a small number of elements and attributes, with very short names which makes it compact if rather terse. It was restricted in the triples that could be written in the graph, for example providing no blank node or datatype literals support and inventing a new base URI mechanism, parallel to XML Base[22] but applying to individual triple parts. This allows all property URIs to be made available (unlike RDF/XML) and abbreviates the long URIs using the relative URI reference. Each *r*, *p* or *v* attribute has a different base URI which can be confusing. In typical applications, only the properties tend to benefit most from relative URIs; subject and objects of triples can be relative but typically need more general URIs.

Triple[25] is "a layered and modular rule language" defined as non XML syntax for RDF, along with an encoding in RDF/XML for TRIPLE<sub>0</sub> used for logic and inference. The Haystack project[24] created the Adenine programming language for describing and manipulating RDF data inside the system. These two languages were intended as domain-specific RDF syntaxes with built-in processing rather than being for more general purposes.

## 6. XML AND NON-XML SYNTAXES

As already introduced, N-Triples and N3 are existing RDF syntaxes that have been deployed successfully as a test case language and a format that is very compact and powerful for semantic web research. Designing a new syntax and not using XML has costs as well as benefits in terms of perceived simplicity that need to be drawn out. XML is generally required by W3C policy for serialisations of web formats except where it is excruciatingly painful. The few non-XML common web formats include CSS which is text in order to be embeddable in HTML and XQuery, although an XML version of the latter is being developed after the text one.

A text format will typically be MIME type *text/something* such as *text/plain*. If it is sent without an encoding, the receiving software is required to treat it as US-ASCII. The protocol or application layer may provide the content encoding via another mechanism (such as an HTTP Content-Encoding header or an *http-equiv* meta tag in HTML). This means text formats lose one of XML's big wins – built-in Unicode and dealing with the internationalization of text. The CSS language is one widely used text web format which has had to solve this, and in CSS2 it gained an *@charset* directive to allow specifying the encoding. N3 was changed from being an US-ASCII format to UTF-8 encoded so that some native encoding of characters are possible, albeit with a restriction to what might be a non-preferred encoding.

Although a text based format might be easy to read and write for people, it does mean writing new tools that deal with the lexical analysis, grammar (and if used, Unicode decoding and encoding). These are the aspects that are already implemented by many well-tested, mature and widely available XML tools and APIs which would have to be discarded for a textual approach.

However, these formats do give (in the least abbreviated form, N-

Triples) a very clear description of the RDF triples and can make the long URIs disappear from user view, when the XML QName-style abbreviations are used. Both the RDF Core and Web Ontology working groups use N-Triples with QName-style abbreviations in their documents as ways to describe the RDF triples. This gives the advantages of improved clarity and reducing the verbosity of full URIs that can decrease comprehension.

New syntaxes written in XML also have a cost, in terms of choosing which XML abstraction to base upon. The revised RDF/XML syntax uses the *XML Infoset*[13] which is the basis of WXS's PSVI and others. Earlier XML technology was designed on SGML, DTDs and the DOM however more recently a new data model the *XQuery 1.0 and XPath 2.0 Data Model*[17] has been designed which looks like the current best-of-breed.

The *SOAP Encoding* (Section 3, [19]) allows the encoding of directed labelled graphs, although it is not yet clear if all RDF graphs could be transferred via this method apart from using it to transport embedding RDF/XML in a naive form. In particular it may be that there is no way to encode blank nodes or RDF datatypes – however whether this is possible is still an ongoing research issue.

## 7. NEW SYNTAX APPROACHES

A new syntax should be closely based on the RDF graph via the terminology in *RDF Concepts and Abstract Syntax*[20] so that it is complete, and also take into account the requirements given earlier (Section 4). In particular the critical requirements (Section 4.1) will be met if it closely aligns with the abstract syntax.

### 7.1 A profile of RDF/XML

A profile of RDF/XML could be made to try to meet the requirements, similar to XMP as previously discussed in section 5. This profile would firstly have to remove most of the abbreviations in order to be minimal. The critical requirement to encode all legal RDF graphs in RDF/XML requires allowing any URI for a predicate. This could be done by adding a new *rdf:predicate* element taking an attribute to give the URI. The resulting syntax would not meet the critical requirement very well to clearly map to the RDF triples unless all node/property element striping was removed, with only 1 level was allowed. The resulting syntax would be something like the example shown in Figure 3 for two triples.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xml:base="http://example.org/">
  <rdf:Description rdf:about="item">
    <dc:title>The Item</dc:title>
  </rdf:Description>
  <rdf:Description rdf:about="item">
    <rdf:predicate rdf:uri="42">abc</rdf:predicate>
  </rdf:Description>
</rdf:RDF>
```

Figure 3: A profile of RDF/XML with a predicate URI

This would have some advantages in being close to the existing syntax but does add yet another form. It does not change enough to deal with any of the other XML design requirements such as using a minimal set of tags (unless the typed node and property element productions were deleted) and as a small change, also does not address any of the conveniences or extended model requirements.

### 7.2 New XML Syntaxes

A new XML syntax that looks like the abstract syntax will tend to seem like an XML-ized version of N-Triples, if it is minimal.

This is sufficient but does not meet the additional XML requirements (Section 4.2) that suggest using some more modern XML design ideas e.g. QNames. At present RDF/XML uses QNames only as the element and attribute names however newer XML work such as WXS use and allow them as attribute values to identify concepts that are identified by a (namespace name, local name) pair. RDF does not use such identifiers, so QNames could only be made to define or refer to URI references, blank node identifiers or literals. This suggests continuing the RDF/XML approach of concatenating the (namespace name, local name) to give a URI. However, QNames used in this fashion cannot encode all URI references so cannot be used as the sole way to encode identifiers for RDF graphs, and thus there must be a way to give any URI. This tends to suggest having either both QName-style and longer URI-style approaches. However, allowing QNames in element content (or attribute values) causes problems such as invisibility from XML processors, XML Namespace scoping and with XML Canonicalization. Mixing QNames with URIs in similar fields can cause interoperability problems since the syntax of both are very similar – `ex:prop` is a syntactically legal QName and URI with URI scheme `ex`.

XML entities are another alternative for abbreviating URIs into shorter forms but they are tied very closely to DTDs and are also not possible to validate with the current WXS. Due to these shortcomings, there are several current discussions in the XML community on ways to use a profile of XML without entities. This suggests that the use of entities in new formats should be avoided[11].

To minimise the vocabulary used for an XML syntax, the elements and attributes must be fixed, with the varying parts of the triples either in element or attribute content (CDATA, or defined by other WXS datatype). Given the requirement to encode all RDF, this means that the distinction between URIs, blank nodes and literals needs to be made either by additional elements or attributes. The additional element for each part of the triple will tend to give a rather verbose appearance as shown by the example in Figure 4 although the `<literal>` element could be omitted here, with the loss of regularity.

```
<triple>
  <node><uri>http://www.w3.org/Home/Lassila</uri></node>
  <node><qname>s:Creator</qname></node>
  <node><literal>Ora Lassila</literal></node>
</triple>
```

**Figure 4: A regular RDF XML syntax in element-normal form**

This *element-normal form* does not read as very modern, so it might be better to replace the `node` element with subject, predicate and object in particular to enforce current RDF model requirements on where URIs, blank nodes and literals can be used (at least for now – it could be removed later to allow RDF extensions).

The main alternative to an all-element approach is to use XML attributes to indicate the triple part such as that shown in Figure 5.

```
<triple>
  <subject uri="http://www.w3.org/Home/Lassila" />
  <predicate ref="s:Creator" />
  <object>Ora Lassila</object>
</triple>
```

**Figure 5: An RDF XML syntax with attributes indicating types**

This looks more modern, like the kind of XML seen in WXS although the attribute names might be slightly different. It is now that introducing the `xsi:type` commonly used for indicating the content is WXS datatypes would fit in well. QNames, URIs and blank

nodes would be all needed which requires both defining and referring attributes for all of these. The main syntax shortcuts that are very common and could be added are for the `rdf:type` property and the collection and container forms.

An additional `type` attribute could be given on the `triple` element to signify the type URI, or as an element inside the element to replace the `predicate` and `object` elements. However both of these would remove the clear triple view; in particular you would get two triples from the attribute form. Examples of these possible typed node forms are shown in Figure 6

```
<triple type='http://example.org/types/Thing'>
  <subject uri="http://example.org/thing1" />
  <predicate ref="ex:prop1" />
  <object>abc123</object>
</triple>
<triple>
  <subject uri="http://example.org/thing22" />
  <type ref="ex:Thing" />
</triple>
```

**Figure 6: An RDF XML syntax with node types**

The container and collections are patterns that respectively generate properties or more complex sets of nodes. These might benefit from support, particularly the latter which is very long to write out longhand and used a lot in OWL, so an additional `collection` element with contained subjects could be added in a form something like that shown in Figure 7 (also showing use of an `xsi:type`)

```
<collection>
  <subject uri="http://example.org/resource" />
  <subject ref="ex:anotherResource" />
  <subject xsi:type="xsd:decimal">10</node>
</collection>
```

**Figure 7: An RDF XML syntax with an RDF collection of nodes**

This new XML syntax design meets all of the critical requirements for a new transfer syntax and most of the other requirements. It has some allowance for common usage patterns in providing a few abbreviations but its best feature is that it would work better with the newer XML technology. The QNames and URIs mixture in attribute values may still be a problem (for both tools and users) and worth simplifying.

The requirement of embedding in XHTML would partially work for any XML syntax, but the restrictions of validation with that make it hard to use without making it work with existing validators, and thus writing a new XHTML Modularization module. These are, however, general problems of embedding any XML format in XHTML while preserving validation.

### 7.3 New non-XML Syntaxes

Any new text-based syntax should be probably be something very similar to the above outline XML designs, with influence from N-Triples and N3 given that they have been found relatively easy to explain (at least in the most regular triple form). The latter has more punctuation than for either a minimal or entirely user-friendly language so would have to be cut down dramatically, but the most commonly used ideas given above have analogues in N3 (QNames, prefixes, datatypes, collections).

As already discussed in section 6, careful updates for internationalisation support such as declaring of charset and enabling the use of local characters in URIs and literals might have to be added.

A textual format could be easily embedded in XHTML by misusing the `<script>` element, which remains a rather distasteful

choice. This would only work easily for users if the use of characters used for structure in XML were avoided. N-Triples and N3 both use < and > for URIs so this does not look practical, leaving external linking as the remaining choice.

If a such a text syntax and an XML one were being designed, it would be a great benefit if they were of a similar level of complexity and preferably, providing as far as possible equivalent mappings to the same model. This has been successfully achieved with the Relax NG XML schema language and it's text equivalent Relax NG Compact.

## 8. N-TRIPLES PLUS

This section describes a proposal for a new textual, non-XML syntax for RDF based on *N-Triples* with some additions from *Notation 3 (N3)*[3]. Many people and groups end up with very similar ad-hoc syntaxes for RDF after a little time playing with writing pseudo-RDF in text, so it clearly matches something that is natural to use. Both the RDF Core and Web Ontology working groups use N-Triples with QName-style abbreviations in their documents as ways to describe the RDF triples, which is equivalent to what is described below, with some predefined namespace prefixes.

The approach taken was to add to N-Triples, a well defined test case syntax for RDF, rather than taking N3, an evolving research language and cutting pieces out. This allows the minimal set of useful additions to be made without adding either beyond-RDF concepts such as N3's {} or syntax forms that are not widely used or understood.

The changes made to N-Triples are as follows:

1. Arbitrary whitespace can be used to separate tokens.
2. The content-encoding is changed from ASCII to UTF-8.
3. @prefix is added to allow using short prefixes for URIs.
4. Namespace-qualified names are allowed for URIs similar to QNames in XML[7]
5. , added to give lists of objects for some subject, predicate.
6. ; added to give lists of predicate, object pairs for some subject.
7. [ ] added to introduce a blank node.
8. a added to abbreviate the very common rdf:type URI.

Figure 8 shows an example of N-Triples Plus using these new features representing the same triples as those created by the RDF/XML Example 7 of *RDF/XML Syntax Specification (Revised)*[12] in section 2.6.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ex: <http://example.org/stuff/1.0/> .

<http://www.w3.org/TR/rdf-syntax-grammar>
dc:title "RDF/XML Syntax Specification (Revised)" ;
ex:editor [
  ex:fullname "Dave Beckett";
  ex:homePage <http://purl.org/net/dajobe/>
] .
```

Figure 8: N-Triples Plus syntax example

The EBNF for N-Triples Plus is given in Appendix A but does not describe the triple generation, which is relatively straightforward. The N-Triples Plus , (**objectList** list) provides a sequence

of triple object nodes and the ; (**predicateObjectList** list) provides a sequence of triple (predicate, object) pairs. These sequences are then used in the **triples** and **blank** productions with a subject **verb** or **blank** to give the three parts of the RDF triples.

The **qname** production allows abbreviation of the URIs like in RDF/XML, the only different is when used against the default namespace (:) when terms like :abc are needed. This definition isn't exactly the same as either XML or N3, since the overlap is not very clear. It presently just adds \_ over the N-Triples name definition. An alternative would be to import the NCNAME definition from *Namespaces in XML*[7] with possibly some exclusions such as '-' and '.' that N3 uses for other syntax. However that would add a dependency on XML that is not currently present in N-Triples Plus. An alternative would be to import the definitions and write them directly in terms of the Unicode character ranges.

An additional syntax form for collections could be added using the existing N3 list syntax (...) that creates RDF collections from the contained ordered sequence of nodes. It would mean adding a **list** term to the alternatives of the **blank** production plus adding a description of the rather complex set of triples that are generated. This would give something like the abbreviation shown in Figure 9.

```
:a :b ( node1 node2 ) .
```

is short for

```
:a :b
[ rdf:first node1;
  rdf:rest [ rdf:first node2;
             rdf:rest rdf:nil ]
] .
```

Figure 9: N-Triples Plus collections example

Other possible extensions would be to add an @base uri to set the base URI in the same fashion as xml:base in XML, and @language to set the default literal language for the following terms in the document.

N-Triples Plus was implemented from scratch using standard lexer and parser generator tools (flex and bison) in a few hours, along with an existing N-Triples parser to handle the encoding rules for URIs and strings.

## 9. CONCLUSION

This paper has discussed the existing RDF/XML syntax and outlined some of its problems, requirements for new syntaxes and from that reviewed existing proposals. Two new syntaxes were described as possible new syntaxes for RDF, an outline of a simple XML syntax and a text syntax N-Triples Plus which both mostly address the critical requirements of syntaxes for RDF.

It has been shown that it is not trivial to make a clearly better syntax, that one syntax will not suit all purposes, and that there are both benefits and costs of pursuing multiple ways to write the same thing, especially when they are written with different audiences in mind.

## 10. ACKNOWLEDGEMENTS

This paper reports on work done under the Semantic Web Advanced Development Europe (SWAD-Europe) project <http://www.w3.org/2001/sw/Europe/> funded by the EU IST-7 programme IST-2001-34732.

## 11. REFERENCES

- [1] D. Beckett. *RDF/XML Syntax Specification (Revised)*. World Wide Web Consortium (W3C), October 2003. W3C Working Draft (work in progress), <http://www.w3.org/TR/rdf-syntax-grammar/>.
- [2] D. Beckett, E. Miller, and D. Brickley. *Expressing Simple Dublin Core in RDF/XML*. Dublin Core Metadata Initiative, July 2002. DCMII Recommendation, <http://dublincore.org/documents/dcmes-xml/>.
- [3] T. Berners-Lee. Notation 3. Technical report, World Wide Web Consortium (W3C), 1998. Design Note, <http://www.w3.org/DesignIssues/Notation3>.
- [4] T. Berners-Lee. A strawman unstriped syntax for RDF in XML. Technical report, World Wide Web Consortium (W3C), 1999. Design Note, <http://www.w3.org/DesignIssues/Syntax>.
- [5] J. Borden and T. Bray. Resource directory description language (RDDL). Technical report, The Open Healthcare Group and Antarctica Systems, June 2003. <http://www.tbray.org/tag/rddl/rddl3.html>.
- [6] T. Bray. The RPV (resource/property/value) syntax for RDF. Technical report, Antarctica Systems, January 2003. <http://www.textuality.com/xml/RPV.html>.
- [7] T. Bray, D. Hollander, and A. Layman. *Namespaces in XML*. World Wide Web Consortium (W3C), January 1999. W3C Recommendation, <http://www.w3.org/TR/REC-xml-names>.
- [8] D. Brickley. *Understanding the Striped RDF/XML Syntax*. World Wide Web Consortium (W3C), October 2001. <http://www.w3.org/2001/10/stripes/>.
- [9] J. J. Carroll. Unparsing RDF/XML. In *Proceedings of the eleventh international conference on World Wide Web*, pages 454–461. ACM Press, 2002.
- [10] J. Clark and M. Makota. *RELAX NG Specification*. OASIS, December 2001. Committee Specification, <http://relaxng.org/spec-20011203.html>.
- [11] K. Clark. The long, long arm of SGML. *XML.com*, November 2003. <http://www.xml.com/pub/a/2003/11/05/deviant.html>.
- [12] W. Conen, R. Klapsing, and E. Kppen. RDF M&S revisited: From reification to nesting, from containers to lists, from dialect to pure XML. In *Proceedings of the first Semantic Web Working Symposium*. Stanford University, July/August 2001.
- [13] J. Cowan and R. Tobin. *XML Information Set*. World Wide Web Consortium (W3C), October 2001. W3C Recommendation, <http://www.w3.org/TR/xml-infoset/>.
- [14] R. Daniel Jr. Harvesting RDF statements from XLinks. Technical report, World Wide Web Consortium (W3C), September 2000. W3C Note, <http://www.w3.org/TR/xlink2rdf/>.
- [15] S. DeRose, E. Maler, and D. Orchard. *XML Linking Language (XLink)*. World Wide Web Consortium (W3C), June 2001. W3C Recommendation, <http://www.w3.org/TR/xlink/>.
- [16] D. Fallside. *XML Schema 1.0*. World Wide Web Consortium (W3C), May 2001. W3C Recommendation, <http://www.w3.org/TR/xmlschema-0/>.
- [17] M. Fernández, A. Malhotra, J. Marsh, M. Nagy, and N. Walsh. *XQuery 1.0 and XPath 2.0 Data Model*. World Wide Web Consortium (W3C), November 2002. W3C Working Draft, <http://www.w3.org/TR/2002/WD-query-datamodel-20021115/>.
- [18] J. Grant and D. Beckett. *RDF Test Cases*. World Wide Web Consortium (W3C), October 2003. W3C Working Draft, work in progress, <http://www.w3.org/TR/rdf-tevstcases/>.
- [19] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, and H. Nielsen. *SOAP Version 1.2 Part 2: Adjuncts*. World Wide Web Consortium (W3C), June 2003. W3C Recommendation, <http://www.w3.org/TR/soap12-part2/>.
- [20] G. Klyne and J. J. Carroll. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. World Wide Web Consortium (W3C), January 2003. W3C Working Draft, work in progress, <http://www.w3.org/TR/rdf-concepts/>.
- [21] O. Lassila and R. Swick. *Resource Description Framework (RDF) Model & Syntax*. World Wide Web Consortium (W3C), February 1999. W3C Recommendation, <http://www.w3.org/TR/REC-rdf-syntax>.
- [22] J. Marsh. *XML Base*. World Wide Web Consortium (W3C), June 2001. W3C Recommendation, <http://www.w3.org/TR/xmlbase/>.
- [23] D. L. McGuinness and F. van Harmelen. *OWL Web Ontology Language Overview*. World Wide Web Consortium (W3C), August 2003. W3C Candidate Recommendation, work in progress, <http://www.w3.org/TR/owl-features/>.
- [24] D. Quan, D. Huynh, and D. R. Karger. Haystack: A platform for authoring end user semantic web applications. In D. Fensel, editor, *Proceedings of Second International Semantic Web Conference (ISWC 2003)*, volume 2870/2003 of LNCS, pages 738 – 753. Springer-Verlag, September 2003.
- [25] M. Sintek and S. Decker. Triple — a query, inference, and transformation language for the semantic web. In I. Horrocks and J. Hendler, editors, *Proceedings of the First International Semantic Web Conference (ISWC 2002)*, LNCS, pages 364–378. Springer-Verlag, June 2002.

## APPENDIX

### A. N-TRIPLES PLUS EBNF

This EBNF is the notation used in XML 1.0 second edition over an alphabet of Unicode characters.

<b>ntriplesPlusDoc</b>	::=	<b>statement</b> *
<b>statement</b>	::=	<b>directive</b> ws* ' . ' ws*   <b>triples</b> ws* ' . ' ws*   <b>comment</b>   ws+
<b>directive</b>	::=	'@prefix' ws+ <b>prefixID</b> ws+ <b>uriref</b>
<b>triples</b>	::=	<b>subject</b> ws+ <b>predicateObjectList</b>
<b>predicateObjectList</b>	::=	<b>verb</b> ws+ <b>objectList</b> (ws+ ' ; ' ws* <b>verb</b> ws+ <b>objectList</b> )*
<b>objectList</b>	::=	<b>object</b> (ws+ ' , ' ws* <b>object</b> )*
<b>verb</b>	::=	<b>predicate</b>   'a'
<b>comment</b>	::=	'#' ( <b>character</b> - ( #xD   #xA ) ) * <i>a line break ends a comment</i>
<b>subject</b>	::=	<b>resource</b>   <b>blank</b>
<b>predicate</b>	::=	<b>resource</b>
<b>object</b>	::=	<b>resource</b>   <b>blank</b>   <b>literal</b>
<b>literal</b>	::=	<b>langString</b>   <b>datatypeString</b>
<b>langString</b>	::=	'" ' <b>string</b> ' "' ( '@ ' <b>language</b> ) ?
<b>datatypeString</b>	::=	'" ' <b>string</b> ' "' '^^^' ( <b>uriref</b>   <b>qname</b> )
<b>blank</b>	::=	<b>nodeID</b>   ' [ ] '   ' [ ' ws* <b>predicateObjectList</b> ws* ' ] '
<b>resource</b>	::=	<b>uriref</b>   <b>qname</b>
<b>nodeID</b>	::=	'_:' name
<b>qname</b>	::=	name? ':' name?
<b>prefixID</b>	::=	':'   name ':'
<b>uriref</b>	::=	'<' <b>relativeURI</b> '>'
<b>language</b>	::=	[a-z]+ ( '-' [a-z0-9]+ ) * <i>encoding a language tag.</i>
<b>name</b>	::=	[A-Za-z][A-Za-z0-9_]*
<b>relativeURI</b>	::=	<b>character</b> * with escapes as defined in [18] section 3.3 turned into an absolute URI reference by resolving against the current base URI
<b>string</b>	::=	<b>character</b> * with escapes as defined in [18] section 3.2
<b>ws</b>	::=	#x9   #xA   #xD   #x20
<b>character</b>	::=	[#x0-#x10FFFF] <i>A Unicode character in the range U+0 to U+10FFFF</i>