XLink in XHTML to Represent RDF

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

Silvia Martelli
ISTI-CNR
Area della Ricerca di Pisa
via Moruzzi 1, Pisa – Italy
+39 050 3152939
silvia.martelli@guest.cnuce.cnr.it

Oreste Signore
ISTI-CNR
Area della Ricerca di Pisa
via Moruzzi 1, Pisa – Italy
+39 050 3152995
oreste.signore@ cnuce.cnr.it

ABSTRACT

Linking is a core common technology shared between the hypertext web and the semantic web. Extended XLinks can encode RDF graphs in the head of XHTML documents. These XLinks carry the semantic markup related to the document, typically using elements from Dublin Core. XLinks from the head into the body permit the use of the document's own displayed metadata. The use of XLink permits the use of RDF without the dreadful RDF/XML syntax. RDF/XML does not conform to XML Schema or DTD, and hence does not embed into validated XHTML. The XLinks are 'harvested' as RDF Statements following the method of Daniel. RDF/XML is not fit for the purpose of layering RDF over XML in the semantic web architecture; this need can be addressed in part by using XLink as an XML serialization for the RDF graph.

Categories and Subject Descriptors

I.7.2 [**Document Preparation**]: Hypertext/hypermedia, Markup Languages; I.2.4 [**Knowledge Representations Formalisms and Methods**]: Representation languages; H.3.7 [**Digital Libraries**]: Standards

General Terms

Languages, Standardization

Keywords

RDF, XHTML, XLink, metadata, Semantic Web Architecture, Dublin Core.

0. THE STATE OF THE ART

For the International Semantic Web Conference 2002, authors were encouraged to provide semantic markup for their abstracts [20].

A tool [21] was provided to help with this. The Semantic markup was created in RDF

… and the results become RDF/XML stored as *comments* within the HTML! (and comments within the RDF/XML caused formatting problems).

This paper addresses the syntactic discontinuity between the Web and the Semantic Web.

1. INTRODUCTION

A major use case for semantic web technologies, particularly RDF [30], has been semantic markup for the annotation of web pages. Such information may be who wrote the page etc. This basic web metadata is typically encoded in Dublin Core [29].

We survey syntaxes for such metadata including: the use of the HTML [38] `<meta>` tags; use of RDF/XML [2] or SHOE [23] markup within the document; and use of RDF/XML markup in an external document. All of these are less than ideal. The solution we propose is to use XLink as a syntax for representing RDF graphs, (following Daniel's harvesting [16]). We represent the RDF graph as extended links within the head element of an XHTML document. Following Daniel, we use the `xlink:arcrole` attribute to indicate the RDF property. We also use the `xlink:role` attribute to distinguish between blank nodes in the RDF graph and those labeled with URI references or literals.

The problems solved in this paper, concerning embedding RDF within XHTML are generic problems concerned with the use of RDF/XML as the syntax for layering RDF over XML. RDF/XML is dreadful. The depth of these problems reveals RDF/XML as not fit for purpose. The XLink technique described can be used for encoding arbitrary *ad hoc* semantic markup (as an RDF graph) in XML documents. This combined with the schema directed transformations envisaged in the Cambridge Communiqué [40] provides a solution to layering RDF over XML.

Thus, unlike Berners-Lee [5], we do not restrict XLink to marking up hypertext links, but find it a workable syntax for marking up semantic links as well. Moreover, we see linking as one of the core technologies that is shared across the hypertext web and the semantic web.

2. METADATA FOR WEB PAGES

For metadata in HTML documents the RDF Model and Syntax [30] recommends, "simply to insert the RDF in-line".

```
<link rel="schema.DC" href="http://dublincore.org/qdcmes/1.0/"
    title="DCMES plus DCMI recommended qualifiers">
<meta   name="DC.Identifier" scheme="URI"
    content="http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/figure1.gif">
<meta   name="DC.Title"              lang="en"         content="A simple RDF assertion">
<meta   name="DC.Type"             scheme="DCMIType"   content="image">
<meta   name="DC.Coverage.temporal" scheme="DCMIPeriod" content="start=1999-04-27">
<meta   name="DC.Creator"                              content="Miller, Paul">
<meta   name="DC.Format.extent"                        content="4033 bytes">
<meta name="DC.Relation.isVersionOf" lang="en" content="Figure 1 from RDF Model and Syntax">
```

Figure 1: Dublin Core in RFC 2731

```
<HEAD>
  <META NAME="author" CONTENT="#L0">    <META NAME="location" CONTENT="#L1">
  <META NAME="tel" CONTENT="#L2">       <META NAME="room" CONTENT="#L3">
</HEAD>
<BODY>
 This page is written by <SPAN id="L0">Frank van Harmelen</SPAN>.
 <SPAN id="L1"> His tel.nr. is <SPAN id="L2">47731</SPAN>, room nr. <SPAN id="L3">T3.57</SPAN>
 </SPAN>
</BODY>
```

Figure 2: Use of from Van Harmelen and Fensel [42]

Another common approach is to use Dublin Core metadata within the <meta> tags in the head element of the document, see RFC 2731 [29], and [15], see Figure 1.

While it is possible to read such mark-up as RDF, it is only useful for metadata that conforms with the Dublin Core schema, or a schema written with interoperation with Dublin Core in mind. Hence, it lacks the open-endedness of RDF metadata, for which any schema [10], or no schema at all, can be used.

SHOE ([23],[24]) is an alternative ontology markup, which is not directly compatible with RDF. This also can be embedded directly in HTML. SHOE aims to improve on RDF by providing more semantic primitives and better mechanisms for managing schema and ontology evolution. SHOE, unlike RDF, provides an XML Document Type Definition (DTD) [9] for its mark-up. This would help enable the use of SHOE within validated HTML.

2.1 Using the Document's own Metadata

Van Harmelen and Fensel [42] show how HTML span elements can be used to pick out of the document body the key data that is the document metadata. See Figure 2. An advantage they pick out for this is that of avoiding duplication: "A basic tenet of information modeling is that redundancy inevitably leads to inconsistency".

We note that this practice of including the metadata inside the document is one that goes back millennia, and suggest that supporting such a well-established practice is a must for a solution to web metadata (cf. the links in printed texts discussed in [25]). The practical advantage is that the human readable document metadata, such as at the beginning of this article, and the machine readable metadata are the same bits: thus modifying an author's name to include a middle initial is done once, in one place for both purposes.

2.2 RDF/XML as Metadata markup?

A problem that emerged after the publication of the RDF Model and Syntax Recommendation is that such RDF/XML cannot be embedded in DTD valid HTML 4.0.

Instead, the revised RDF/XML syntax [2] recommendation suggests that the metadata should form a separate document, and can be related to the original document using a link element in the head. (Not unlike the way that pictures are included in HTML documents).

This can be regarded as the web equivalent of having a separate card in a card index as the primary repository of the document metadata. A long tradition can also be pointed to for this practice.

We note that a DTD has been developed for RDF/XML expressing simple Dublin Core [3]. Like with the SHOE DTD, this could, in principle, be added to the XHTML DTD. However, it does not extend to the more sophisticated requirements of qualified Dublin Core [28].

2.3 Requirements for Metadata markup Syntax

We can summarize this brief survey of the prior art by capturing the requirements and desiderata for Web metadata markup:

These are:

- Ability to represent Dublin Core metadata (both simple and qualified)
- Ability to represent other (arbitrary) semantic links
- Compatibility with HTML or XHTML [34]
- Compatibility with validated HTML or XHTML
- Compatibility with XML
- Compatibility with XML validated against some DTD or XML Schema
- Compatibility with RDF
- Ability to refer to the metadata already in-line in the document
- Ability to refer to new metadata in the document head

(See also the list of requirements in [1]).

None of the examples already surveyed has all of these properties.

This paper, presents the use of XLink as a serialization of RDF, to be used for metadata markup within XHTML. This solution, combined with "harvesting" techniques inspired by Daniel [16] *does* meet these requirements.

3. TECHNOLOGIES OVERVIEW

3.1 XML Linking Language: XLink

Linking is an essential component of hypertext, and hence of the web, since its inception. The traditional link mechanism in HTML was so crude to force the adoption of some tricks to specify the link semantics. XLink ([18]), instead, allows the specification of richer links, where we can state the *arcrole*, which can account for the *reason why* we have the link. This basic mechanism can be used to associate a document, or a part of it, to a concept in an ontology. Such linking lies at the heart of this paper.

XLink is the W3C Recommendation for creating and describing links between resources into XML documents. A resource is any addressable unit of information or service. An XLink link asserts the existence of an explicit relationship between some resources (or portions of them) by a linking element that is an XLink-conforming XML element. In order to use XLink, it is necessary to declare the XLink namespace [8] that provides global attributes for use on elements, which may be from any namespace.

XLink provides a way for creating both simple and extended links. A simple link associates exactly two resources, a local one and a remote one (i.e. a resource specified by an URI reference), with an arc from the former to the latter. Extended links can have arbitrary numbers of participating resources (local and/or remote). An extended link indicates rules for traversing among its participating resources by means of a series of *arc* elements. For

both simple and extended links, the meaning of the arc's ending resource relative to its starting resource can be specified using the semantic attribute `xlink:arcrole`. The manner in which the resource behaves within the extended link can be specified using the further attribute `xlink:role`.

3.2 RDF

The Resource Description Framework (RDF) can be used to represent arbitrary data. This is done using the RDF graph [27] in which many of the nodes and all of the arcs are labeled with URIs (conventionally shortened to qnames). These graphs are read as descriptions using RDF Vocabularies [10] and the RDF Formal Semantics [22]. Arcs are read as indicating a relationship between their two ends. So an arc from the URL of this document to the string "Silvia Martelli" and labeled with `dc:creator` indicates that this document and the string "Silvia Martelli" are in the relationship described by `dc:creator`. Reading the description of the relationship found at the full URL http://purl.org/dc/elements/1.1/creator, we discover that "Silvia Martelli" is the name of an author of this document.

An example of an RDF graph from [28]:

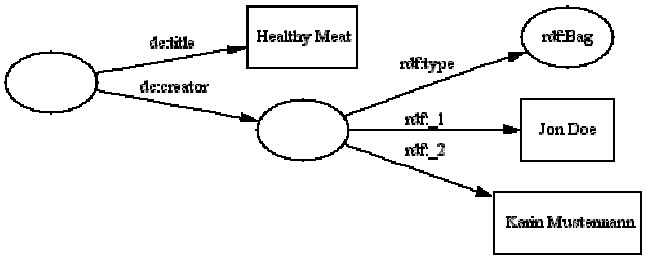

Figure 3: An RDF graph from [28]

RDF/XML [2] provides particular serializations of this graph in XML, an example being:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:dcterms="http://purl.org/dc/terms/">
<rdf:Description dc:title="Healthy Meat">
 <dc:creator>
   <rdf:Bag>
     <rdf:li>Jon Doe</rdf:li>
     <rdf:li>Karin Mustermann</rdf:li>
   </rdf:Bag>
 </dc:creator>
</rdf:Description>
</rdf:RDF>
```

Figure 4: RDF/XML from [28]

RDF/XML is an appalling syntax.

It suffers from the following defects:

- It has been impossible to furnish it with a DTD or XML Schema. Embedding it in validated XML is impossible.
- It does not use XML Namespaces in a way that is sufficiently compatible with other XML languages. Specifically elements and attributes from other namespaces can be used, but their syntax in RDF/XML is as specified by RDF/XML, and cannot be modified by any declaration in their defining namespace.
- An XML document may be a legal RDF/XML document when validated as XML and not otherwise, or conversely, or may have its meaning as an RDF/XML document changed by XML validation.
- There are a huge number of non-trivial variants for the serialization of an RDF graph. This provides for confusion and misunderstanding rather than utility.
- There are some very obscure and wholly indefensible constructs such as `rdf:bagID`.
- It is not possible to serialize all RDF Graphs using it, in particular those with at least one property URI, such as http://www.w3.org/, which does not end with a no-colon name from XML Namespaces [8].

In combination, these features make RDF/XML a source of perpetual confusion for the experienced user as well as the novice. Many leading participants in the field cannot read or write it, and the semantic web would be improved by its discontinuation[1].

In short RDF is a simple and elegant data model, with a clear and well-formulated semantic foundation lacking an adequate web syntax.

3.3 Harvesting

Both XLink and RDF provide a way for asserting relationships between resources. It is possible to define a mapping between links and RDF statements. This process is addressed as *harvesting* [16]. The key insight is the meaning of the *arcrole* attribute in XLink matches the meaning of the predicate in an RDF statement.

The underlying principles for harvesting are:

- each arc with an `xlink:arcrole` attribute originates at least one RDF *statement*;
- the starting resource is the RDF statement *subject*;
- the ending resource is the RDF statement *object*;
- the value of `xlink:arcrole` attribute is the RDF statement *predicate*.

The full harvesting process can be implemented using appropriate XSLT transformations, as described in [16].

4. SOWING RDF STATEMENTS AS XLINKS

"Whoever sows generously will also reap generously," St Paul

"You're going to reap just what you sow." – Lou Reed

A fundamental weakness with Daniel's harvest is that there isn't any crop. There is hardly any XLink data to which to apply such techniques [19]. We differ from Daniel in one key respect: we sow before we try to reap the harvest. We deliberately mark up our documents with XLink as a way of giving them semantic content. We use XLink as a serialization for the RDF graph.

4.1 A Simple Example

The semantic description of an XHTML document: the title, the authors, the keywords, the short description, is often contained in the document itself.

The author of the web page can directly identify these chunks of information, usually using a tool such as OntoMat-Annotizer [21]. To mark the position of this information in the page a `<span>` element with an unique `id` attribute can be used. This markup acts as an anchor for the semantic description. This description is built as an extended XLink in the `<head>` section. We confine the

[1] The current revision [2] is a substantial improvement on the older version [30]. The defects listed have not been addressable.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:xlink="http://www.w3.org/1999/xlink"
      xml:lang="en"
      xml:base="http://weblabsrv.cnuce.cnr.it/www2003/abstract.html"
      base=" http://weblabsrv.cnuce.cnr.it/www2003/abstract.html">
<!-- xlink:href="" is a same document reference -RFC 2396- and refers to this document-->
  <head xlink:type="extended">
    <link xlink:type="locator" xlink:href="" xlink:label="doc"
          xlink:role="http://www.w3.org/TR/rdf-concepts/#dfn-URI-reference"/>
    <link xlink:type="locator" xlink:href="#id-title" xlink:label="title"
          xlink:role="http://www.w3.org/TR/rdf-concepts/#dfn-plain-literal"/>
    <link xlink:type="locator" xlink:href="#id-carroll" xlink:label="author"
          xlink:role="http://www.w3.org/TR/rdf-concepts/#dfn-plain-literal"/>
    <link xlink:type="locator" xlink:href="#id-martelli" xlink:label="author"
          xlink:role="http://www.w3.org/TR/rdf-concepts/#dfn-plain-literal"/>
    <link xlink:type="locator" xlink:href="#id-signore" xlink:label="author"
          xlink:role="http://www.w3.org/TR/rdf-concepts/#dfn-plain-literal"/>
    <link xlink:type="locator" xlink:href="#id-abstract" xlink:label="description"
          xlink:role="http://www.w3.org/TR/rdf-concepts/#dfn-plain-literal"/>
    <link xlink:type="arc" xlink:from="doc" xlink:to="author"
          xlink:arcrole="http://purl.org/dc/elements/1.1/creator"/>
    <link xlink:type="arc" xlink:from="doc" xlink:to="title"
          xlink:arcrole="http://purl.org/dc/elements/1.1/title"/>
    <link xlink:type="arc" xlink:from="doc" xlink:to="description"
          xlink:arcrole="http://purl.org/dc/elements/1.1/description"/>
  </head>
  <body>
    <h1 id="id-title">Syntax for Semantic Enriching of Web Pages</h1>
    <p><span id="id-carroll">Jeremy J. Carroll</span>....</p>
    <p><span id="id-martelli">Silvia Martelli</span>....</p>
    <p><span id="id-signore">Oreste Signore</span>....</p>
    <h2> ABSTRACT</h2>
    <p id="id-abstract">Linking is a core common technology… </p>
  </body>
</html>
```

Figure 5: Example Marked Up Document

XLink markup in the <head> section of the XHTML page, because it is the reserved area for containing information about the document. Moreover, the browser does not display this section.

Consider Figure 5, the abstract of this paper (as shown below). In that we use URIs from the RDF Concepts and Abstract Syntax document [27].

The following describe the additional markup (DC metadata for dc:title, dc:creator, and dc:description.), was added to the abstract:

- In the body of the XHTML document, for each title, author name and description, we insert span elements with an id.
- We create an extended XLink in the head section. We add a xlink:type="extended" attribute-value pair to the <head> element.
- We add a <link> element to indicate the document itself.
  - xlink:type="locator"
  - xlink:href="", an RFC 2396 ([6]) same document reference.
  - xlink:label="doc", a local name (extended link scoped) for the element.
  - xlink:role. We use rdfc:dfn-URI-reference[2]. This indicates that the role played by this resource within this link is that of an absolute URI reference.
- For each of the span elements added, we add a <link> element in the head section. The <link> element has attributes taken from the XLink namespace:
  - xlink:type="locator"
  - xlink:href, which points to the correspondent anchor in the body.
  - xlink:label, which gives a local name (having scope in the extended link itself) for this element.
  - xlink:role, which is, in the simple example, always a fixed URI indicating a plain literal. We use rdfc:dfn-plain-literal[2]. This indicates that the role played by this resource within this link is that of a simple text string. Other possible values are discussed below.
- Finally, we create arcs, each being another <link> element. Each arc generates as many RDF statements (triples) as are the couples of resources having the label specified in xlink:to and xlink:from. The attributes are:
  - xlink:type="arc"
  - xlink:from a local name of some other <link> element in this extended link. The subject resource of the triple is found by retrieving the elements having this value as their local name. If the xlink:role of that element is rdfc:dfn-URI-reference, then the

[2] In this description, we use the qname convention from RDF to shorten some URIs; specifically the rdfc prefix is bound to http://www.w3.org/TR/rdf-concepts/#.
So that rdfc:dfn-plain-literal is short for http://www.w3.org/TR/rdf-concepts/#dfn-plain-literal.

```
xmlns:rdfc = "http://www.w3.org/TR/rdf-concepts/#"
xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
foreach (relevantArc) {'*[@xlink:type="extended"]/*[@xlink:type="arc"&&@xlink:arcrole!=""]'}
  predicate:=value-of(xlink:arcrole)
  subjectLabel:=value-of(xlink:from)
  objectLabel:=value-of(xlink:to)
  foreach (relevantSubjectLinkElement) {"//*[@xlink:label=$subjectLabel]"}
   choose
     when (xlink:role="rdfc:dfn-URI-reference")
        subjectType:="rdfc:dfn-URI-reference"
        subjectValue:= value-of(xlink:href)
     when (xlink:role="rdfc:dfn-blank-node")
        subjectType:="rdfc:dfn-blank-node"
        subjectValue= value-of(xlink:label)
   endchoose
   foreach (relevantObjectLinkElement) {"//*[@xlink:label=$objectLabel]"}
     choose
       when (xlink:role="rdfc:dfn-URI-reference")
          objectType:="rdfc:dfn-URI-reference"
          objectValue:=value-of(xlink:href)
       when (xlink:role="rdfc:dfn-blank-node")
          objectType:="rdfc:dfn-blank-node"
          objectValue:=value-of(xlink:label)
       when (xlink:role=”rdfc:dfn-plain-literal”)
          objectType:=”rdfc:dfn-plain-literal”
          objectValue:=value-of(element that has the id equal to xlink:href minus ‘#’)
       when (xlink:role=”rdfs:XMLLiteral”)
          objectType:=”rdfs:XMLLiteral”
         objectValue:= value-of(element that has the id equal to xlink:href minus ‘#’)
     endchoose
     createTriple(subjectType,subjectValue,predicate,objectType,objectValue)
   endforeach
  endforeach
endforeach
```

Figure 8: Pseudocode for harvesting RDF Statements from XLinks

subject of the triple is the URI reference specified by `xlink:href` of that element.

- `xlink:to` a local name of some other `<link>` element in this extended link. The object resource of the triple is found by retrieving the elements having this value as their local name. If the `xlink:role` of that element is `rdfc:dfn-plain-literal`, then the object of the triple is the string found by dereferencing the `xlink:href` and taking the element content (restricted to being simple text).
- `xlink:arcrole`, being an absolute URI reference, the predicate of the triple.

We preferred to not introduce new elements to build the extended XLink, to maintain higher compatibility with the XHTML DTD.

The XLink markup is verbose. The use of XML entities could shorten it.

4.2 More Sophisticated Sowing of XLinks

In the previous example we inserted only simple DC metadata in our page. Each RDF statement had the URI of the document as subject, a URI for expressing the property, and a plain literal value for the object.

However, in the RDF abstract syntax [27], there are six different kinds of statement:

- the subject can be a URI reference or a blank node;
- the property is always a URI reference;
- the object can be an URI reference, or a blank node, or a literal.

To distinguish different subject and object types we use the `xlink:role` attribute on both locator-type and resource-type element. For example to indicate a blank node we use the following markup:

```
<head xlink:type=”extended”>
 ...
 <link xlink:type="resource" xlink:label="a"
  xlink:role="http://www.w3.org/TR/rdf-
concepts/#dfn-blank-node"/>
 ...
</head>
```

Figure 6: A blank node in XLink

In this case the `xlink:label` attribute is used to distinguish among different blank nodes.

Moreover, we can use the `xlink:role` to indicate the datatype URI of typed literals in RDF: a special case that is important in XHTML is the special literal type `rdfs:XMLLiteral`, defined in [27]. This corresponds to the RDF/XML construction `rdf:parseType=”Literal”`. Within the harvesting process, recognizing this special datatype permits the inclusion of text with XHTML markup as literal values in the RDF graph. Such text is particularly important for certain text types: e.g. text which includes bi-directional text, marked up with the HTML `dir` attribute; Japanese text marked up with ruby annotation [39]; and mathematics texts marked up with MathML [12].

If a literal value needed in the metadata is not available in the body of the document, it is possible to explicitly include it in the link element of `xlink:type` “resource” in the head of the document. This is done by using the `xlink:title` attribute with the literal string as its value.

```
<head xlink:type="extended">
...
  <link xlink:type="resource"
        xlink:label="xx"
        xlink:title="Author full name"
        xlink:role="http://www.w3.org/TR/rdf-
concepts/#dfn-plain-literal"/>
...
</head>
```

Figure 7: A plain literal in the head

With XLink, neither the document nor the document type, needs extensive changes: in the XHTML DTD only two elements are affected.

5. HOW TO EXTRACT SEMANTIC INFORMATION

In order to extract (harvest) RDF statements from the web pages a simple XSL file transforms the XHTML into an intermediate XML representation of the triples. Finally, with a simple software program, the RDF/XML serialization of the statement or an other representation can be generated.

We describe the harvesting algorithm in pseudocode, see Figure 8.

This imperative style pseudo-code differs substantially from the corresponding XSLT, which imposes a strict functional programming paradigm. The XSLT version is available for download [32]. At the moment our XSL allows to extract an XML description of each RDF statement. We show below a fragment of the result we can obtain from harvesting XLinks from the example XHTML page shown before (Figure 5).

```
<triples
xmlns:xlink="http://www.w3.org/1999/xlink"
xml:base="http://weblabsrv.cnuce.cnr.it/www2003/ab
stract.html ">
...
<triple>
  <subject type="http://www.w3.org/TR/rdf-
concepts/#dfn-URI-reference"/>

<predicate>http://purl.org/dc/elements/1.1/creator
</predicate>
  <object type="http://www.w3.org/TR/rdf-
concepts/#dfn-plain-literal">
    Jeremy J. Carroll
  </object>
</triple>
...
</triples>
```

Figure 9: Sample Harvested Triple

6. XLINK MEETS THE REQUIREMENTS

We review the requirements.

- Ability to represent Dublin Core metadata

  This has been demonstrated.

- Ability to represent other (arbitrary) semantic links

  Instead of using URIs from the Dublin Core space as arcrole we can easily use other URIs from other semantic spaces. We use the standard technique of URIs to refer to those other spaces. Moreover, we can add blank nodes [27] into the RDF graph, and write arcs involving those in XLinks. This permits the serialization of an arbitrary RDF graph within the document, and thus gives a full semantic generality to the metadata markup. (Unlike with RDF/XML [2], we can serialize cases, such as a property URI: http://www.w3.org/).

- Compatibility with HTML or XHTML

  XLinks are described using XML attributes, which can be added to XHTML elements as shown. The same techniques could be used on HTML, but harvesting them would require more forgiving parsing technology than a standard XML parser.

- Compatibility with validated HTML or XHTML

  With current DTDs the xlinks that we add are not valid. However, the changes required to permit XLinks everywhere in XHTML are simple, and appear[3] to be a requirement for XHTML 2.0 (cf. [43]). Our approach requires XLink attributes on only two elements, both within the document head. These changes are substantial less than those required by HTML extensions for XLink using elements in the body as well as attributes such as RDDL[4] [7].

- Compatibility with XML

  None of the techniques we used were dependent upon features found in XHTML that are not generic XML features.

- Compatibility with XML validated against some DTD or XML Schema

  Many new XML DTDs and XML Schema being produced by the W3C allow XLinks. Modifying an old DTD or Schema to permit arbitrary XLink attributes everywhere is trivial. (We believe XML would be improved by defining the syntax of namespace qualified attributes within the namespace of attribute qualification, rather than within the namespace of the qualification of the parent element; with such a generic change we could use XLinks everywhere wholly unproblematically)

- Compatibility with RDF

  Our data is RDF. We have defined an alternative XML serialization of the same abstract syntax [27].

- Ability to refer to the metadata already in-line in the document

  This has been demonstrated.

- Ability to refer to new metadata in the document head

  This has been demonstrated.

7. ARCHITECTURAL CONSIDERATIONS

The techniques of this paper concern both the semantic web and the (HTML) web. We do not restrict our architectural thoughts to one domain or the other. Linking is a key common aspect.

---

[3] We are neutral as to the HLink [37] vs Xlink controversy [44]. We note that harvesting hlinks as RDF statements would be possible but significantly more difficult. The difficulty is shared by most generic processing of HLink.

[4] RDDL (the Resource Directory Description Language) [7] is superficially similar to our approach. XLink is used within XHTML and can be transformed into RDF. However, this is conceived of as providing an RDF representation of non-RDF data (typically, RDDL descriptions of namespaces). It is not a serialization of an arbitrary RDF graph; e.g. blank nodes are not used as objects. RDDL uses simple XLinks on a new element `rddl:resource` in the body of the document, rather than our extended XLink using preexisting elements in the head.

7.1 Linking

We consider linking as primary in web architecture.[5] The essential characteristic of links is associations among documents and among resources. Links are the richness of the Web.

In a rude but effective approximation, we can classify links in two classes: associative links that capture such associations, and structural links, which are part of document structure. The associative links exploit the full potentialities of the web.

Another, non-orthogonal, classification of links is the distinction between extensional links (explicitly stored in the document) and intensional ones (that can be derived from the content). Therefore links, especially the intensional ones, by themselves convey some semantics. In fact, the Semantic Web can be seen as the semantics of a global set of links.

Semantically rich linking is the basis for implementing relationships and hierarchies of concepts, a fundamental component of any ontology. This is done within RDF and RDF schema, where these semantic links are represented both as labeled arcs within the RDF Graph [27] and as the class and property hierarchies within RDF Schema [10]. Moreover, at an ontology level, OWL [17] permits explicit links between ontologies using `owl:imports`.

XLink ([18]) is the W3C recommendation for linking, and we believe that it can and should be used for both these traditional hypertext links, and also for semantic links. Thus, becoming a further point of contact between Web and Semantic Web technologies and architecture.

We fail to understand Berners-Lee [5]'s motivation for suggesting that XLink should be used only for hypertext links; we find it natural to use a single linking technology across the web and semantic web; just as a single standard for character encoding (Unicode) and a single standard for web references (URIs [6]) are used across both.

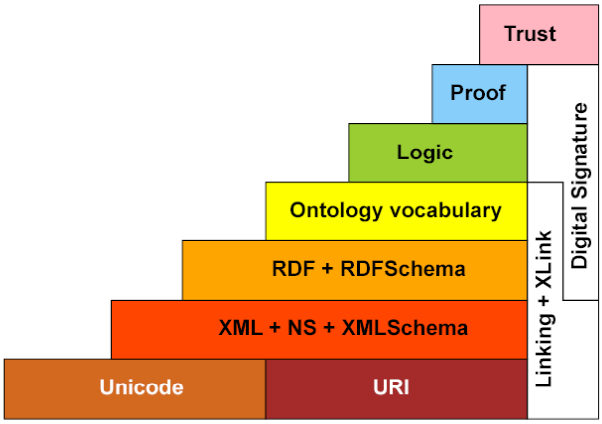

Figure 10: The modified Semantic Web stack

Thus we take Berners-Lee's architecture picture [4] and we augment it with linking as another core technology (Figure 10). Linking spans multiple layers; and is placed alongside digital signatures.

[5] The W3C's Technical Architecture Group have not treated linking as primary in their Architectural Principles [26].

7.2 RDF/XML: not fit for purpose

We have seen earlier that RDF/XML [2] does not have enough of the desirable features for metadata markup. Neither does it have the features required of it in this architecture picture.

RDF is layered on top of XML, indicating that RDF graphs [27] can be expressed in XML [9]. This requirement is normally understood as being addressed by RDF/XML [2]; in the figure RDF/XML may be thought of as cementing the RDF and XML layers together.

But RDF/XML is not compatible with XML Schema [41] or DTD validation.

Nor does RDF/XML respect XML namespaces [8]. It is possible to use elements and attributes from any namespace within an RDF/XML document; but it is not possible to effectively use the syntax associated with these elements. For example, a `xsi:type` attribute within an RDF/XML document will be interpreted as a property attribute creating a new arc in the graph. The associated value which is a qname (e.g. "`xsd:decimal`") will be interpreted as an opaque absolute URI in the (unknown) xsd scheme. Thus the independence of distinct XML namespaces is not respected.

Of course, these points are not new, and are only too well known by practitioners (see e.g. Patel-Schneider and Siméon's work: [36],[35]).

RDF/XML does not fit into, and is not fit for inclusion in, the semantic web architecture.

The recent round of RDF recommendations separates the various concerns of RDF into distinct but related recommendations. This rejection of RDF/XML [2] is neither a rejection of RDF as the fundamental web data structure [27], nor its associated semantics [22], nor even of the vocabulary used to build RDF schema [10].

7.3 RDF – the base semantic layer

As has been suggested by Patel-Schneider and Siméon, a better way to see this layering is with RDF providing a simple semantic model (based on dyadic predicates), and XML providing a surface serialization.

Unlike Patel-Schneider we do not believe it is possible to have a once for all definition of the semantics of an XML document. In particular we note that different W3C standards take a different point of view as to what is significant in an XML document (contrast the infoset [14] based standards, such as the DOM [31], with the XPath Nodeset [13] based standards such as Canonical XML [11]).

7.4 Cambridge Communiqué

The Cambridge Communiqué [40] discusses "the construction of application-oriented data structures (e.g. ones implementing the RDF model)" from XML during XML Schema validation. This is a sensible path to the true meaning of an XML document. I.e. the meaning of an XML document is (largely) schema specific.

The promises of this communiqué are still very relevant, and the program of work that it envisages is of increasing urgency.

However, such schema directed translation still misses the possibility of *ad hoc* metadata, and *ad hoc* meaning.

Semantic markup in XHTML documents is all about capturing metadata about the document that cannot be ascertained from the HTML elements alone. Thus a purely schema directed methodology captures only those aspects of an XHTML document that are relevant to the resource as an XHTML document. The

meta-information that is normally found in Dublin Core statements about the document is excluded.

Thus, in a world (that envisaged by the Cambridge Communiqué) in which the meaning of an XML document is primarily extracted with a schema directed translation, there is still a need for *ad hoc* markup. We believe that RDF/XML is unusable for such a role and that XLink is a much better fit.

7.5 XLink in the Semantic Web

RDF/XML should not be used as a Semantic Web technology. Much of the layering between XML and RDF should be addressed by the schema transformation approach sketched by the Cambridge Communiqué. In addition *ad hoc* semantic links should be addressed with the standard linking technology XLink. The absence of RDF/XML from our semantic web stack is not an oversight but a positive decision for cleansing.

8. CONCLUSION AND FUTURE WORK

We have shown that XLink can be used with validated XHTML documents to encode semantic markup. The use of a single web language (XLink) for both semantic and hypertextual links is a distinct advantage.

This markup needs to be combined with semi-automatic tools for semantic markup such as OntoMat-Annotizer [21] in order to be deployable. With the XSLT transform [32] we have developed, marked up pages can easily be loaded into semantic web tool kits such as Jena [33].

XLink can also be used for embedding RDF graphs in other XML documents; we believe that when combined with further work, along the lines suggested in the Cambridge Communiqué, these techniques will (and should) make RDF/XML redundant as a technology for the semantic web.

9. REFERENCES

[1] Altheim, M., and Palmer, S. B. (eds.). Augmented Metadata in XHTML. Sun Microsystems Working Draft, 2001. http://infomesh.net/2002/augmeta/

[2] Beckett, D. (ed.). RDF/XML Syntax revised. W3C Working Draft, 2002. http://www.w3.org/TR/2002/WD-rdf-syntax-grammar-20021108/

[3] Beckett, D., Miller E., Brickley, D., Expressing Simple Dublin Core in RDF/XML, DCMI Recommendation, 2002. http://dublincore.org/documents/2002/07/31/dcmes-xml/

[4] Berners-Lee, T. "Architecture". http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html

[5] Berners-Lee, T. When should I use XLink?, 2002. http://www.w3.org/DesignIssues/XLink.html

[6] Berners-Lee, T., Fielding, R., Masinter, L. (eds.). Uniform Resource Identifiers (URI): Generic Syntax. RFC 2396, Internet Engineering Task Force, 1998.

[7] Borden, J., Bray, T. Resource Directory Description Language (RDDL), 2002, http://www.openhealth.org/RDDL/20020218/rddl-20020218.html

[8] Bray, T., Hollander, D., Layman, A.(eds.). Namespaces in XML. W3C Recommendation, 1999. http://www.w3.org/TR/REC-xml-names

[9] Bray, T., Paoli, J., Sperberg-McQueen, C. M. (eds.). Extensible Markup Language (XML) 1.0. W3C Recommendation, 1998. http://www.w3.org/TR/1998/REC-xml-19980210

[10] Brickley, D., Guha, R.V. (eds.). Resource Description Framework (RDF) Schema Specification 1.0. W3C Candidate Recommendation, 2002. http://www.w3.org/TR/2002/WD-rdf-schema-20021112/

[11] Boyer, J. (ed.). Canonical XML Version 1.0. W3C Recommendation, 2001. http://www.w3.org/TR/xml-c14n

[12] Carlisle, D., Ion, P., Miner, R., Poppelier, N. (eds.). Mathematical Markup Language (MathML) Version 2.0. W3C Recommendation, 2001. http://www.w3.org/TR/MathML2

[13] Clark, J., DeRose, S. (eds.). XML Path Language (XPath) Version 1.0. W3C Recommendation, 1999. http://www.w3.org/TR/xpath

[14] Cowan, J., Tobin, R. (eds.). XML Information Set. W3C Recommendation, 2001. http://www.w3.org/TR/xml-infoset

[15] Cox, S., Miller, E., Powell, A. Recording qualified Dublin Core metadata in HTML meta elements. DCMI Working Draft, 15 August 2000.

[16] Daniel, R. J. Harvesting RDF Statements from Xlinks. W3C Note, 2000. http://www.w3.org/TR/xlink2rdf

[17] Dean, M. et al. OWL Web Ontology Language 1.0 Reference. W3C Working Draft, 2002. http://www.w3.org/TR/owl-ref/

[18] DeRose, S., Maler, E., Orchard, D. (eds.) XML Linking Language (XLink) Version 1.0. W3C Recommendation, 2001. http://www.w3.org/TR/2001/REC-xlink-20010627/

[19] DuCharme, B. XLink: Who Cares? (2002). http://www.xml.com/pub/a/2002/03/13/xlink.html

[20] Ehrig, M. (ed.). ISWC abstracts. (2002). http://annotation.semanticweb.org/iswc/documents.html

[21] Handschuh, S., and Staab, S. Authoring and Annotation of Web Pages in CREAM. in Proceedings of WWW2002 Conference (Honolulu, Hawaii, May 7-11, 2002). http://www.aifb.uni-karlsruhe.de/WBS/sha/papers/aa_cream_www11.pdf

[22] Hayes, P. (ed.). RDF Model Theory. W3C Working Draft, 2002. http://www.w3.org/TR/2002/WD-rdf-mt-20021112/

[23] Heflin, J., and Hendler, J. Semantic Interoperability on the Web. in Proceedings of Extreme Markup Languages 2000. http://www.cs.umd.edu/projects/plus/SHOE/pubs/extreme2000.pdf

[24] Heflin, J., Hendler, J., Luke, S. SHOE: A Blueprint for the Semantic Web. in Fensel, D., Hendler, J., Lieberman, H., Wahlster, W. (eds.). Semantic Web: Why, What and How?

[25] Hemrich, M., and Schafer, U. XML-based linking concepts. in Proceedings of 23rd International Online Information Meeting, pp. 31-38, Published: Woodside, UK, 1999.

[26] Jacobs, I. Architectural Principles of the World Wide Web. W3C Working Draft, 2002. http://www.w3.org/TR/webarch/

[27] Klyne, G., Carroll, J. J. Resource Description Framework RDF Concepts and Abstract Syntax. W3C Working Draft, 2002. http://www.w3.org/TR/2002/WD-rdf-concepts-20021108/

[28] Kokkelink, S., and Schwänzl, R. Expressing Qualified Dublin Core in RDF / XML. DCMI Proposed Recommendation, 2002. http://dublincore.org/documents/2002/04/14/dcq-rdf-xml/

[29] Kunze, J. A. Encoding Dublin Core metadata in HTML. RFC 2731, Internet Engineering Task Force, December 1999.

[30] Lassila, O., and Swick, R. R. (eds.). Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, 1999. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

[31] Le Hors, A. et al. Document Object Model (DOM) Level 2 Core Specification. W3C Recommendation, 2000. http://www.w3.org/TR/DOM-Level-2-Core/

[32] Martelli S. XSLT for Harvesting Semantics from Extended Xlinks. (2002). http://weblabsrv.cnuce.cnr.it/www2003/xlinktransformation.xsl

[33] McBride, B. Jena. IEEE Internet Computing (July/August 2002). http://dsonline.computer.org/0211/f/w6jena.htm

[34] W3C HTML Working Group. XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition). W3C Recommendation, 2002. http://www.w3.org/TR/xhtml1/

[35] Patel-Schneider, P. F., and Siméon, J. Building the Semantic Web on XML. ISWC 2002.

[36] Patel-Schneider, P. F., and Siméon, J. The Yin/Yang Web: XML Syntax and RDF Semantics. in Proceedings of WWW2002 Conference (Honolulu, Hawaii, May 7-11, 2002).

[37] Pemberton, S., and Ishikawa, M. HLink: Link recognition for the XHTML Family. W3C Working Draft, 2002. http://www.w3.org/TR/hlink/

[38] Raggett, D., Le Hors, A., Jacobs, I. HTML 4.01 Specification. W3C Recommendation, 1999. http://www.w3.org/TR/html401

[39] Sawicki, M., Suignard, M., Ishikawa, M., Dürst, M., Texin, T. Ruby Annotation. W3C Recommendation, 2001. http://www.w3.org/TR/ruby/

[40] Swick, R. R., and Thompson, H. S. The Cambridge Communiqué. W3C Note, 1999. http://www.w3.org/TR/schema-arch

[41] Thompson, H. S., Beech, D., Maloney, M., Mendelsohn, N. XML Schema Part 1: Structures. W3C Recommendation, 2001. http://www.w3.org/TR/xmlschema-1/

[42] Van Harmelen, F., and Fensel, D. Practical Knowledge Representation for the Web. IJCAI 99.

[43] Walsh, N. TAG Comments on XHTML 2.0 and Hlink. http://lists.w3.org/Archives/Public/www-tag/2002Sep/0183.html

[44] Williams, S. Summary of Technical Discussion of XLink/HLink on www-tag. (2002). http://www.w3.org/2001/tag/2002/1021-XLink-Hlink.html