

A COMPARATIVE STUDY OF RDF AND RDFa

Kumar Abhishek *

Prabhat Kumar **

Archana P. Kumar ***

ABSTRACT

For a long time now, RDF has shown great promise as a flexible format for storing, aggregating, and using metadata. Maybe for too long—its most well-known syntax, RDF/XML, is messy enough to have scared many people away from RDF. The W3C is developing a new, simpler syntax called RDFa (originally called "RDF/a") that is easy enough to create and to use in applications that it may win back a lot of the people who were first scared off by the verbosity, striping, container complications, and other complexity issues that made RDF/XML look so ugly.

RDF/XML doesn't have to be ugly, but even simple RDF/XML doesn't fit well into XHTML, because browsers and other applications designed around HTML choke on it. So, while the general plan for RDFa is to make it something that can be embedded into any XML dialect, the main effort has gone into making it easy to embed it into XHTML. This gives it an important potential role in the grand plan for the Semantic Web, in which web page data is readable not only by human eyes but by automated processes that can aggregate data and associated metadata and then perform tasks that are much more sophisticated than those that typical screen scraping applications can do now. In fact, the relationship between RDFa metadata and existing content in web pages has been an important driver in most use cases driving RDFa's progress.

Keywords: *RDF, RDFa, Semantic Web, XML files.*

* Department of Computer Science and Engineering, NIT, Patna.

** Department of Information Technology, NIT, Patna.

*** Department of Computer Science and Engineering, MIT, Manipal.

I. INTRODUCTION

Semantic Web is a group of methods and technologies to allow machines to understand the meaning - or "semantics" - of information on the World Wide Web. [1]. The semantic web is a vision of information that can be interpreted by machines, so machines can perform more of the tedious work involved in finding, combining, and acting upon information on the web.

On the Semantic Web (SemWeb), computers do the browsing (and searching, and querying, and...) for us. The SemWeb enables computers to seek out knowledge distributed throughout the Web, mesh it, and then take action based on it. Take an analogy: the current web is a decentralized platform for distributed *presentations*, while the SemWeb is a decentralized platform for distributed *knowledge*.

Resource Description Framework (RDF) is the W3C standard for encoding knowledge. What is meant by "semantic" in Semantic Web is not that computers are going to understand the meaning of anything, but that the logical pieces of meaning can be mechanically manipulated by a machine to useful *human ends*.

The rapidity of the growth experienced provides the impetus for researchers to focus on the creation and dissemination of innovative Semantic Web technologies, where the envisaged 'Semantic Web' is long overdue.

II. RESOURCE DESCRIPTION FRAMEWORK

Resource Description Framework was originally developed as a metadata model, the first specification of RDF came in the year 1999 authored by Ora Lassila and Ralph Swick [3].

RDF is based on XML and is defined as a language used for representing information about resources in the World Wide Web that is adding metadata to the web resource [3].

The basic idea of RDF is to identify things (resource) using Web Identifiers. These web identifiers are known as Uniform Resource Indicator (URI). RDF also describes resources in terms of properties and properties value [3].

RDF is also known as TRIPLES where triple is a magic number that is three piece of information needed to fully define a single bit of knowledge. The three pieces of information are subject, property type and property value. For example I (subject) have a name (property), which is Kumar Abhishek (property value) [4].

In English grammar rule a complete sentence (or statement) contains two things a *subject* and a *predicate*: the subject is the who or what of the sentence and the predicate provides information about the subject [4]. For example

The title of the article is "*Pranab Roy*."

In the above example subject is the article, and the predicate is title, with a matching value of "Pranab Roy" [4].

If the above English statement is translated to an RDF triple, the *subject* is the thing being described—in RDF terms, a *resource* identified by a URI and the *predicate* is a property type of the resource, such as an attribute, a relationship, or a characteristic [4].

Apart from subject and predicate the specification also introduces a third component, the *object* [4]. In RDF, the object is equivalent to the value of the resource property type for the specific subject [4]. RDF core committee decided to represent the data model in RDF using directed label graph [4]. The RDF directed graph consists of a set of nodes connected by arcs, forming a pattern of *node-arc-node*. The nodes come in three varieties: *uriref*, blank nodes, and literals [4].

The *uriref* consists of a Uniform Resource Identifier (URI) reference that provides a specific identifier unique to the node. Blank nodes are nodes that don't have a URI. The literals consist of three parts—a character string and an optional language tag and data type. Literal values represent RDF objects only, never subjects or predicates. In RDF literals are represented by drawing rectangles around them [4].

A) *RDF Data Model*

The representation of data in RDF is done via graph. In RDF as we know it comprises of three things subject-object and predicate, in RDF statement the term subject is either a Uniform Resource Identifier (URI) or a blank node, which denote resources [4].

Resources which are represented by blank nodes are termed as anonymous resources. These resources are not directly identified from RDF statement. The term predicate is also an URI which indicates a resource, representing a relationship. The object is a URI, blank node or a Unicode string literal [4].

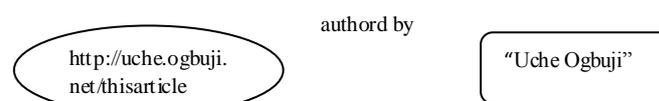


Figure 1- RDF Statement [5]

Figure 1 depicts a graph representation of RDF statements [5].

In the figure above, the object is a string: "Uche Ogbuji". Then the object is termed as *literal* in RDF, but an object could also be a resource [5].

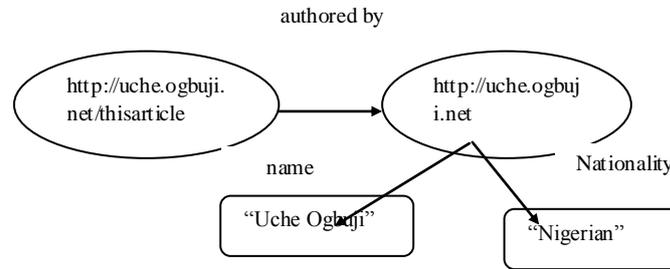


Figure 2: A small RDF model [5]

Figure 2 depicts combination of several RDF statement into a single diagram. The expansion of RDF is done on this basis.

RDF describes a Web-based resource by defining a directed graph of statements.

In the above figure, the object which is a literal "Uche Ogbuji" is replaced by a URI indicating this person, which in turn is the subject of several more statements. This type of collection of RDF statements is termed as *model* in RDF [5].

Uniform Resource Locator are used to point to Web Documents that describe the exact meaning (semantics) of each edge type.

Table 1: Grammar General Notation

Notation	Meaning
<i>event.accessor</i>	The value of an event accessor.
rdf:X	A URI .
"ABC"	A string of characters A, B, C in order.

A) *Advantages of RDF*

The advantage of RDF is that it is readily extensible with schemas that are also machine readable, bringing about an unprecedented level of automation.

XML allows great flexibility for adaptation of data formats, and RDF provides great flexibility for adaptation of data-processing rules.

Traditional database management tools are too highly structured, and therefore add hefty maintenance costs when the real world inevitably changes around the application.

The evolution would allow Web crawlers to gather more than just plain keywords

III. RESOURCE DESCRIPTION FRAMEWORK IN ATTRIBUTES (RDFa)

Use of standard cataloguing metadata would assist search engine Web crawlers and other machine agents the way HTML meta tags help search engines index Web pages.

The question arises what happens in the context of the *real world*, where developers use APIs and environments such as PHP? The developer just writes some more PHP scripts to deliver RDF/XML, at different URIs to the HTML data. (If the HTML data is really exactly equivalent to the RDF data, then content-negotiation can be used to deliver them at the same URI, but this should be used with caution, typically only when the HTML and RDF are generated from the same data and contain exactly the same information.)

For an existing site which has pages in (X)HTML, how can they integrate or reference the RDF-based data available if they were to publish it? A number of proposals exist how to 'marry' RDF and HTML. The result is RDFa, RDF in "a" attributes [3].

RDF often uses a subject, predicate, object combination called a triple to specify an attribute name/value pair about a particular resource. That's "attribute" in the object-oriented sense, not the XML sense; for example, a triple could specify that the resource with ID `http://example.com/artwork#fountain` has an author value of "Richard Mutt." To allow you to add metadata to a web page without affecting a browser's display of that page, RDFa uses some existing XHTML 1 attributes and a few new XHTML 2 attributes to store the subjects, predicates, and objects of these RDF triples. The objects may also be existing PCDATA in your web pages, with subject and predicate attributes letting this text play a dual role of human-readable displayed content and machine-readable metadata.

The following are the attributes used in RDFa:

- `about` and `src` – a URI specifying the resource the metadata is about
- `rel` and `rev` – specifying a relationship or reverse-relationship with another resource
- `href` and `resource` – specifying the partner resource
- `property` – specifying a property for the content of an element
- `content` – optional attribute that overrides the content of the element when using the property attribute
- `datatype` – optional attribute that specifies the datatype of text specified for use with the property attribute
- `typeof` – optional attribute that specifies the RDF type(s) of the subject (the resource that the metadata is about) [3].

The following **example** shows the format of XHTML+RDFa (viz. RDFa). [4]

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
  "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
```

```
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:dc="http://purl.org/dc/elements/1.1/"
version="XHTML+RDFa 1.0" xml:lang="en">
<head>
  <title>John's Home Page</title>
  <base href="http://example.org/john-d/" />
  <meta property="dc:creator" content="Jonathan Doe" />
  <link rel="foaf:primaryTopic" href="http://example.org/john-d/#me" />
</head>
<body about="http://example.org/john-d/#me">
  <h1>John's Home Page</h1>
  <p>My name is <span property="foaf:nick">John D</span> and I like
  <a href="http://www.neubauten.org/" rel="foaf:interest"
  xml:lang="de">Einstürzende Neubauten</a>.
</p>
<p>
  My <span rel="foaf:interest" resource="urn:ISBN:0752820907">favorite
  book is the inspiring <span about="urn:ISBN:0752820907"><cite
  property="dc:title">Weaving the Web</cite> by
  <span property="dc:creator">Tim Berners-Lee</span></span>
</span>
</p>
</body>
</html>
```

In the example above, the document URI can be seen as representing an HTML document, but the document URI plus the "#me" string `http://example.org/john-d/#me` represents the actual person, as distinct from a document about them. The *foaf:primaryTopic* in the header tells us a URI of the person the document is about. The *foaf:nick* property (in the first span element) contains a nickname for this person, and the *dc:creator* property (in the meta element) tells us who created the document. The hyperlink to the Einstürzende Neubauten website contains `rel="foaf:interest"`, suggesting that John Doe is interested in this band. The URI of their website is a resource.

The *foaf:interest* inside the second p element is referring to a book by ISBN number. The resource attribute defines a resource in a similar way to the href attribute, but without

defining a hyperlink. Further into the paragraph, a span element containing an about attribute defines the book as another resource to specify metadata about. The book title and author are defined within the contents of this tag using the *dc:title* and *dc:creator* properties.

Here are the same triples when the above document is automatically converted to RDF/XML:[4]

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://example.org/john-d/">
    <dc:creator xml:lang="en">Jonathan Doe</dc:creator>
    <foaf:primaryTopic>
      <rdf:Description rdf:about="http://example.org/john-d/#me">
        <foaf:nick xml:lang="en">John D</foaf:nick>
        <foaf:interest rdf:resource="http://www.neubauten.org/">
          <foaf:interest>
            <rdf:Description rdf:about="urn:ISBN:0752820907">
              <dc:creator xml:lang="en">Tim Berners-Lee</dc:creator>
              <dc:title xml:lang="en">Weaving the Web</dc:title>
            </rdf:Description>
          </foaf:interest>
        </rdf:Description>
      </foaf:primaryTopic>
    </rdf:Description>
  </rdf:RDF>
```

A) *Advantages over RDF*

While doing comparison of RDFa over RDF. The following advantages were observed:

- RDFa lets you add triples of useful metadata to your XHTML with simple, straightforward markup.
- While these are popular for RDFa because you can insert them anywhere in the body of an HTML document, you can add the same RDFa attributes. This is not possible through RDF.

- Publisher Independence – each site can use its own standards
- Data Reuse – data is not duplicated. Separate XML and HTML sections are not required for the same content.
- Self Containment – The HTML and the RDF are separated
- Schema Modularity – The attributes are reusable
- Evolvability – additional fields can be added and XML transforms can extract the semantics of the data from an XHTML file.

VI. RECENT DEVELOPMENTS

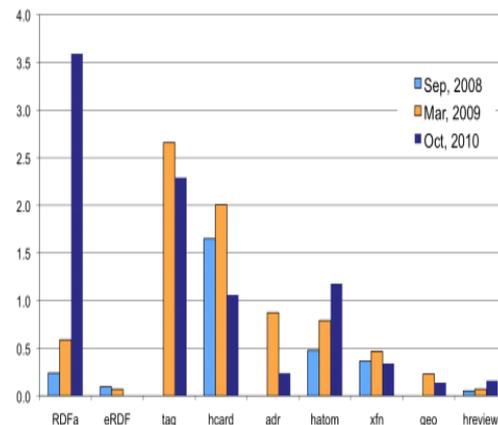


Fig.3. RDFa deployment on the Web (% of all web)

The data shows that the usage of RDFa has increased 510% between March, 2009 and October, 2010, from 0.6% of webpages to 3.6% of webpages (or 430 million webpages in our sample of 12 billion). This is largely thanks to the efforts of the developers at Yahoo! (Search Monkey), Google (Rich Snippets) and Facebook (Open Graph), all of whom recommend the usage of RDFa[3].

V. CONCLUSION

The RDFa architecture raises some user interface issues such as:

- If you represent the data as well as the hypertext, will the data be repeated, confusing the user?
- If you don't represent the data, will a user have access to the full power of data handling?
- There is a security threat if the user does not see (but agrees to) data which is not visible.
- Are there systems which currently assume that a localID within a hypertext document will identify an anchor? How will they cope with RDFa?

VI. FUTURE WORK

The tools for extracting data from the XHTML+RDFa formats have to be developed so that they can be deployed into to the existing web browsers. Even though some development in the extraction process has been developed such as Sindice Inspector and Ubiquity RDFa parser, more efficient parsers have to be developed. Does httpRange-14 apply to the RDFa setup equally? Making such improvements is required.

REFERENCES

1. W3C "Semantic Web Frequently Asked Questions". W3C. <http://www.w3.org/2001/sw/SW-FAQ>. Retrieved March 13, 2008.
2. Michael Grobe : RDF, Jena, SparQL and the "Semantic Web"
3. <http://www.w3.org/TR/rdf-primer>
4. Shelley Powers: Pratical RDF, O'Reilly,2003
5. IBM<http://www.ibm.com/developerworks/library/w-rdf>
6. "Introduction to RDFa" – www.xml.com
7. "RDFavsRDF/XML"- <http://www.w3.org/wiki>
8. "Introduction to RDF"- <http://www.ibm.com/developerworks/library>