

# The Role of RDF in the IMS Family of Specifications

Mikael Nilsson\*  
Center for User-Oriented IT-design,  
Royal Institute of Technology, Stockholm

27th September 2001

## 1 Background

In May 2001, the IMS 1.2 meta-data specification was released. This specification represented an important step in the maturization of the IMS meta-data standard, and in the underlying IEEE LOM standard, combining many lessons that have been learned since the meta-data specification was first published. But it also contained a new element, included as an appendix to the XML binding document: an IMS meta-data RDF binding. The inclusion of an RDF binding in a draft version has spawned discussion and raised questions as to the role RDF plays, and should play, in the whole family of IMS specifications.

This document is an attempt at answering some of those questions; more precisely: What is RDF anyway, and what is the vision underlying it? What advantages and disadvantages does RDF have over a pure XML approach? Specifically for IMS, what advantages does using RDF provide? What could/should be the relative roles of XML and RDF in the IMS family of specifications? Naturally, the discussion is relevant for all meta-data efforts related to IMS, such as IEEE LOM and SCORM.

Sections 2 to 5 discuss the general features of RDF as compared to XML, and can be skipped by readers more interested in the consequences for the meta-data specification work. Sections 6 to 8 more specifically discuss RDF in relation to the whole family of IMS specifications and related efforts.

## 2 RDF origins: The Semantic Web vision

The Semantic Web is the name of a long-term project recently started by W3C with the stated purpose of realizing

the idea of having data on the Web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications.<sup>1</sup>

---

\*mini@nada.kth.se

<sup>1</sup>W3C Semantic Web Activity Statement at <http://www.w3.org/2001/sw/Activity>, part of the W3C Semantic Web site at <http://www.w3.org/2001/sw/>.

It was motivated by the very same problems that motivates the development of meta-data standards<sup>2</sup> – the fact that raw media, in the form of text, HTML, images or video streams, contains meta-information that may be readily deducible from the context for the human consumer (the name of the author, the kind of material contained within, etc.), but is mostly inaccessible to computers. Making this information available to computers in order to enhance their usefulness, was the driving vision that created the Semantic Web project.

Most traditional meta-data approaches take the view of meta-data as being mostly a digital indexing scheme to use in cataloging and digital libraries. What distinguishes the Semantic Web from these approaches to meta-data are two important things:

- The Semantic Web is designed to allow reasoning and inference capabilities to be added to the pure descriptions. In its simplest form, this includes stating facts such as “a hex-head bolt is a type of machine bolt”<sup>3</sup>, but extends to the deduction of complicated relationships. This is an important feature to allow intelligent agents and other software to not only passively swallow descriptions, but to act on them as well.
- The Semantic Web is a web-technology that lives on top of the existing web, by adding machine-readable information without modifying the existing Web. It is designed to be globally distributed with all this means in terms of scalability and flexibility.

The Semantic Web is a layered structure. XML forms the basis, being the transport syntax. RDF provides the information representation framework, and on top of this layer, schemas and ontologies provide the logical apparatus necessary for the expression of vocabularies, enabling intelligent processing of information.

### 3 Essential RDF

While the current Web allows you to *link* to anything from anything in a machine-understandable way, the Semantic Web will allow you to *say* anything about anything in a machine-readable way. Seen this way, RDF is the language in which Semantic Web meta-data statements are expressed. In fact, RDF can be said to consist purely of so-called *statements*. An RDF statement consists of three elements: a *subject*, a *predicate*, and an *object*. Statements are about Web resources, so subjects and objects are URIs, machine-readable identifiers. Objects can also be plain text strings. Saying “*The document <http://www.w3c.org/2001/sw/> was created by W3C*” is represented by the triple (“<http://www.w3c.org/2001/sw/>”, **created by**, “W3C”). To disambiguate the different predicates that can be used, every predicate must be given a URI. In this case, there is a standard predicate available in the Dublin Core vocabulary, namely “<http://purl.org/dc/elements/creator>”, which we can use. The triple then becomes

---

<sup>2</sup>The Semantic Web activity is, as a matter of fact, the successor of the W3C Metadata activity. Thus, the Semantic Web is the W3C meta-data architecture.

<sup>3</sup>Taken from Tim Berners-Lee, James Hendler, Ora Lassila: *The Semantic Web* (<http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html>), published in Scientific American, May 2001.

(“<http://www.w3c.org/2001/sw/>”, “<http://purl.org/dc/elements/creator>”, “W3C”).

This demonstrates that URIs can be used to name not only concrete digital documents on the web, but abstract entities as well. In order to talk about non-digital resources, we must give them URIs. For example, to talk about the organization W3C (i.e., use it the subject of a statement), we must give it a URI. Let’s give it the URI “<http://www.w3c.org/organization>”. We can now say things such as “<http://www.w3c.org/2001/sw/> was created by <http://www.w3c.org/organization>, which is an organization with the name ‘W3C’”, which is, in fact, three separate statements. More complicated RDF expressions like this are usually represented as graphs, where the subjects and objects are nodes, and the predicates are edges<sup>4</sup>.

This is all there is to basic RDF – nodes-and-arcs diagrams interpreted as statements about concepts or digital resources represented by URIs<sup>5</sup>. However, the need for standardized vocabularies for things like “organization” and the predicate “is a” etc. is evident. The basis for such vocabularies in RDF is *RDF Schema*<sup>6</sup>. This specification provides the basic vocabulary to express relationships between terms: resources being **instances** of terms (“<http://www.w3c.org/organization> is an organization”), terms being **subterms** of other terms (“a hex-head bolt is a type of machine bolt”) and so on. It also provides means to restrict the usage of predicates: “is a parent of” only **applies to** persons, etc. The terms **instance**, **subterm**, **applies to** are the kind of terms defined by the RDF Schema specification.

Using the vocabulary provided by RDF Schema, it is easy to create your own semantically rich vocabularies.

## 4 Uses of RDF for Resource Description

It is not immediately obvious that the simple statement model of RDF can be used to make the Semantic Web a reality. The most fundamental benefit of RDF compared to other meta-data approaches is that using RDF, you can say *anything about anything*. Anyone can make RDF statements about any identifiable resource. Using RDF, the problems of extending meta-data and combining meta-data of different formats, from different schemas disappear, as RDF does not use closed documents. Important uses of RDF to encode information for any resource you can name with a URI include:

**describe** Since a resource can have uses outside the domain foreseen by the author, any given description (meta-data instance) is bound to be incomplete. Because of the distributed nature of RDF, a description can be expanded, or new descriptions, following new formats (schemas), can be added. This allows for new creative uses of content in unforeseen ways. This is one of the important features of the current Web, where anyone can link to anything, that has been carried over into RDF.

**certify** There is no reason why only big organizations should be able to certify content – individuals such as teachers may want to certify a

---

<sup>4</sup>For a more detailed explanation of these concepts, see e.g. An Introduction of RDF by Eric Miller (<http://www.dlib.org/dlib/may98/miller/05miller.html>) or the RDF syntax and model specification <http://www.w3.org/TR/REC-rdf-syntax>.

<sup>5</sup>RDF also contains an important mechanism called *reification*, that allows you to state something about another RDF statement, such as who said it, whether it is true or false, etc.

<sup>6</sup>Defined in <http://www.w3.org/TR/rdf-schema>

certain content as a quality learning resource that is well suited for specific learning tasks. How to handle this kind of certification will be an important part of the Semantic Web.

**annotate** Everything that has an identifier can be annotated. There are already attempts in this direction: Annotea<sup>7</sup> is a project where annotations are created locally or on a server in RDF format. The annotations apply to HTML or XML documents and are automatically fetched and incorporated into web pages via a special feature in the experimental browser Amaya<sup>8</sup>.

**extend** Structured content (typically in XML format) will become common. Successive editing can be done via special RDF-schemas allowing private, group consensus or author-specific versions of a common base document. The versioning history will be a tree with known and unknown branches which can be traversed with the help of the next generation versioning tools.

**reuse** RDF is application independent. As the meta-data is expressed in a standard format independent of more advanced schemas that are used, even simplistic applications can understand parts of large RDF descriptions. If more advanced processing software is available (such as logic engines), more advanced treatment of the RDF descriptions is possible.

## 5 RDF and XML: Model, Syntax, Semantics

So far nothing has been said about XML. The reason is that the *RDF Model* can be defined completely without reference to XML. XML can, however, be used as a *syntax* for RDF statements. The RDF specification defines the standard syntax to encode RDF statements in XML.

But one question remains: why can't XML and XML Schema be used to represent the same kind of information that RDF expresses? XML Schemas are, after all, powerful tools to express complex requirements on XML elements. This is true, and XML and XML Schemas can be used to do some of what RDF does, but not without much trouble. The reasons are several:

- *The RDF model and the XML model are fundamentally different.* The XML data model is a text-markup oriented labeled tree. RDF, by contrast, has a very simple model consisting of labeled arcs. Of course, any specific set of RDF statements forms a graph that can be serialized in XML. But as XML and XML Schema are designed primarily for fixed, tree-like documents, they are significantly less flexible for expressing meta-data, which by its very nature is subjective, distributed and expressed in diverse forms. The RDF model, while simpler, is flexible enough to support these principles.
- *The resources used in RDF and XML Schemas are fundamentally different.* The nodes that XML Schemas talk about are nodes in an XML document, at specific places in a document structure. In RDF, the nodes are not nodes in the document itself, but rather any resources

---

<sup>7</sup><http://www.w3.org/2001/Annotea/>

<sup>8</sup><http://www.w3.org/Amaya/>

that have URIs, and more often than not live outside the RDF document itself. Thus, RDF is designed to be a meta-data language.

- *The semantics of XML Schemas and RDF are fundamentally different.* XML Schemas have a primarily syntactic interpretation, restricting the set of XML documents that can be produced. RDF, on the other hand, has a primarily semantic interpretation. While XML Schemas are used for modeling XML documents, RDF is used to model knowledge, where tree-based representations are not enough.

The difference can be formulated in this way<sup>9</sup>: XML/XML Schema is a data modeling language, and RDF is a meta-data modeling language. When meta-data needs to be encoded as data, an XML syntax is very useful. However, *modeling* meta-data in pure XML severely restricts its flexibility.

## 6 Lessons from Using RDF in the IMS Specifications

From the discussion thus far, the consequences for IMS are still unclear. This unclarity was one of the most important reasons behind the construction of the RDF binding of the IMS 1.2 meta-data specification, and subsequent efforts to produce an RDF binding for IMS Content Packaging. Some important positive lessons learned in this effort were:

- Interoperability with other, separate, standards is greatly increased. The reason is simple: RDF allows *a single storage model* for very different types of data and schemas. For example, storing meta-data from different specifications in the same database is straightforward. To implement searching that includes dependencies between meta-data expressed in different schemas is simplified. An example of this is the Edutella<sup>10</sup> effort to build a peer-to-peer educational meta-data exchange network, which would meet severe difficulties in using, searching in, and translating between the different formats used for VCard, Dublin Core, Dublin Core Qualifiers, IEEE LOM, SCORM etc., which are meta-data standards without the common RDF format.
- *Reuse* of existing meta-data standards is greatly simplified. For example, there has been much discussion on whether to incorporate the VCard XML syntax in the XML binding. While desirable, this creates namespacing and XML DTD problems. In the RDF binding, the VCard RDF binding can be transparently included with *no extra effort*.
- Some terms do not have exact equivalents in other meta-data standards, but relate to some existing terms by, for example, being more narrow, more broad etc. As the whole IMS RDF binding was designed as an extension to Dublin Core meta-data<sup>11</sup>, the relationship between IMS meta-data elements and Dublin Core elements are formalized *in a machine-readable manner*. Thus, no conversion to/from Dublin Core meta-data is needed, and Dublin Core aware tools can understand the Dublin Core-parts of an IMS meta-data description.

---

<sup>9</sup>See also <http://www.w3.org/DesignIssues/RDF-XML.html> or in more detail: [www.ontoknowledge.org/oil/download/IEEE00.pdf](http://www.ontoknowledge.org/oil/download/IEEE00.pdf)

<sup>10</sup><http://edutella.jxta.org/>

<sup>11</sup>See <http://dublincore.org/>

- There has also been much discussion on the topic of *vocabularies*. While these are essential for the use and extension of IEEE LOM meta-data, there does not exist a standard way to encode and distribute them. In RDF, this problem completely disappears, as vocabularies, as we have seen, are a fundamental part of the RDF Schema specification. Not only is there a standard way to list vocabulary items, but their interdependencies can be modeled in a standard way. And as if that was not enough, efforts such as DAML<sup>12</sup> and OIL<sup>13</sup> provide means to model vocabularies as full-fledged *ontologies*<sup>14</sup> expressed in RDF, if that is desired, while still maintaining compatibility with less-capable software.
- While extending the XML binding is certainly possible using XML Schemas, the process easily creates interoperability problems. In RDF, several independent means of meaningful extensions are available, none of which cause interoperability problems:
  - *Refinement* of the semantics of existing properties and terms by creating subterms etc. This cannot be done in a standard way using XML.
  - *Introduction of new properties* and terms describing resources. This is the kind of extension one can usually do in XML.
  - *Adding new properties* to a resource in other documents, which is possible since RDF does not work with meta-data instances as closed documents. For example, the RDF binding is designed so that translations of titles and description etc., can be managed separately. In the same way, different kinds of meta-data can be managed separately, and merged when needed<sup>15</sup>. This *modularity* is impossible to achieve in a clean and standard way using XML.

These possibilities are not only nice properties of RDF, but are completely indispensable in many cases.

- RDF already contains means for describing *meta-meta-data* (in any number of meta-steps)<sup>16</sup>, that can be as rich as ordinary meta-data.
- For meta-data that contain very complex interdependencies, such as IMS Content Packaging, the graph representation and modularity of RDF effectively *cleans up the format* and semantics of the specification.
- RDF allows for a clean integration of the different specifications in a layered way. Currently, the work on IMS Content Packaging in RDF is built on top of the IMS Meta-data RDF binding, which is built on top of the VCard RDF binding and the Dublin Core Qualifiers RDF binding, that extends the core Dublin Core RDF binding. Continuing

---

<sup>12</sup>See <http://www.daml.org/>

<sup>13</sup>See <http://www.ontoknowledge.org/oil/>

<sup>14</sup>See e.g. <http://www.ontology.org/main/papers/faq.html> for a definition of the word “ontology” in this context. The paper *Combining Ontologies and Terminologies in Information Systems* by Johann Gamper, Wolfgang Nejdl and Martin Wolpers (<http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/1999/tke99/>) is also of interest.

<sup>15</sup>For an important example of this, see the UNIVERSAL project <http://nm.wu-wien.ac.at/universal/>, and <http://www.ist-universal.org/>.

<sup>16</sup>Via *reification*, mentioned above.

upwards in this fashion, I see a very concrete potential for a *complete unification* of all IMS meta-data related specifications<sup>17</sup>. This is no light-minded suggestion, and the benefits, as seen in this list, are many.

These lessons are in no way coincidences. While XML was designed as a data interchange format, RDF was designed from the ground up to fulfil the role of an Internet architecture for meta-data. "*Resource Description Framework (RDF) is a foundation for processing metadata; it provides interoperability between applications that exchange machine-understandable information on the Web*"<sup>18</sup>. This is very clearly reflected in the findings above.

On another note, RDF presents several drawbacks, as has been made clear during the work on the RDF bindings.

- The underlying standards, notably the RDF Schema specification, but also the Dublin Core Qualifiers RDF binding, are still young, intensively discussed and possibly subject to change. The specifications underlying the XML binding are much more stable (even if the XML Schema specifications have changed recently). This is, of course, a temporary problem.
- Tool support for RDF is very immature at this point, and integration of Semantic Web technologies into the current Web is still only starting. XML support can be said to be mature in most respects. However, with the current pace of RDF adoption, tool support is rapidly increasing.
- Designing an RDF binding makes it necessary to revisit many of the assumptions in the underlying information model, which often is designed with an XML binding in mind. As the semantics of XML elements is not explicitly stated, much of the work in designing an RDF binding goes into defining the semantics of the elements. This has caused minor interoperability problems between the XML binding and the RDF binding, problems that can only be remedied by designing the information model with the RDF binding in mind.

From another perspective, this is a very positive side effect, as it significantly helps sharpen the information model. This has already been observed in the design of the IMS meta-data RDF binding, but from preliminary studies seems to be even more evident in the work on an RDF binding of IMS Content Packaging.

## 7 Possibilities on the Semantic Web

From a more strategic point of view, the emerging Semantic Web presents exciting new possibilities for uses of the IMS specifications. While XML standards are very good tools for enabling interoperability by specifying import and export formats for LMSs<sup>19</sup>, they tend to favor large, centrally managed, monolithic systems.

<sup>17</sup>Something similar has already been done in the UNIVERSAL project, where e.g. learner and contributor information is added on top of low-level meta-data.

<sup>18</sup>RDF specification: <http://www.w3.org/TR/REC-rdf-syntax>

<sup>19</sup>Learning Management Systems – a software system for managing an interactive learning environment.

By enabling the use of IMS specifications in Semantic Web technologies, a much wider range of applications are imaginable:

- Intelligent software agents can be implemented, helping the learner to find and use globally distributed learning resources.
- Personal annotations of any learning resource becomes a feasible technology, as demonstrated by Annotea<sup>20</sup>.
- Collaborative and distributed authoring and course construction becomes much simpler thanks to the modularity of the information.
- Reuse of learning material by cross-fertilization suddenly becomes a reality, creating important synergy effects.

The Semantic Web promises to create a web-based eco-system for learning resources, freeing the material from being trapped in closed systems. One important example of this kind of technology is Edutella<sup>21</sup>, an RDF-based peer-to-peer system under development, being designed to allow distributed access to learning resource meta-data expressed in many different schemas. By combining meta-data from many sources in a controlled but distributed way, cross-annotation and mutual reuse of material becomes a reality.

In short, the vision of the Semantic Web is an important vision for online learning as well.

## 8 Suggestions for IMS on the use of RDF

We now turn to a proposal for the possible future role of RDF within IMS. It is not intended as a final proposal, but only to ignite discussion on this important topic.

**What status should an RDF binding have?** The first problem is what status an RDF binding should have in a single IMS specification. From the experience with the IMS 1.2 Meta-data binding, it has become clear that an RDF binding is a complete reimplementaion of the information model. Thus, the only natural role for an RDF binding is to be interpreted as a separate binding, to live alongside any XML binding.

**Should all IMS specifications have RDF bindings?** In light of the analysis above, it should be clear that many of the IMS specifications would benefit from an RDF binding. Any specification that would be useful in a Semantic Web scenario is a candidate for having an RDF binding, and this certainly includes most specifications.

**What about the XML bindings?** There is definitely a need for XML bindings. In many scenarios, exchange of meta-data instances is the main concern, and XML is a very good technology for that – especially in the case of light-weight, unintelligent clients.

<sup>20</sup><http://www.w3.org/2001/Annotea/>

<sup>21</sup><http://edutella.jxta.org/>



### **What about compatibility between the RDF and the XML bindings?**

This has been highlighted as being problematic in some aspects in the IMS 1.2 Meta-data RDF binding. Some sorts of interoperability problems are inevitable, especially the ones resulting from the fact that RDF does not use closed meta-data instances in the same way as XML. Other kinds of problems, such as which tokens to use for vocabulary items, can be rectified by early cooperation between RDF and XML binding designers – for example, using URIs to denote vocabulary items in XML, making them addressable in RDF as well. In any case, automatic conversion to/from XML is an important aspect of the design of an RDF binding.

### **How should development of RDF specifications within IMS take place?**

As has been pointed out above, the specifications produced by the different parties in the meta-data community (Dublin Core, VCard, LOM, SCORM) all have complex interdependencies, and increasingly so. It has become evident that the reuse of specifications within IMS is not at the desired level. This has to do with the closed nature of XML bindings. RDF bindings, on the other hand, are much more intensely interdependent as a consequence of the heavy reuse of vocabularies. A coordinated effort within the IMS project to produce a coherent set of RDF bindings would be necessary. This effort will return important feedback to the information model designers, so that in the end, producing an RDF binding will be much less work than formulating the information models and the initial XML binding.

Such an effort will also greatly reduce the difficulties involved in adding another IMS specification to an existing software system – it would only involve adding a new RDF Schema, which has been designed to interoperate with the existing schemas. This has important consequences for the adoption of IMS specifications.

## **9 Conclusion**

The future for RDF within IMS is bright, and the possibilities opened up by RDF and Semantic Web technologies promise to take the IMS project to a new level of applications. But some effort from IMS to produce the necessary specifications will be needed. In particular:

- Each IMS specification needs an RDF binding.
- The RDF efforts within IMS need to be coordinated to produce a coherent set of RDF-based specifications.
- Interoperability discussions with other standards groups such as Dublin Core and IEEE LOM need to be intense in order to maximize interoperation.

RDF provides an important technological platform to handle the interoperability demands of the emerging specification and vocabulary jungle, inside and outside of IMS.