

Resource Description Framework (RDF)

Nicholas Gibbins and Nigel Shadbolt

Intelligence, Agents, Multimedia Group

School of Electronics and Computer Science

University of Southampton

Southampton SO17 1BJ

United Kingdom

{nmg,nrs}@ecs.soton.ac.uk

KEYWORDS

Semantic Web, metadata, XML, linked data, ontologies, knowledge representation

ABSTRACT

The Resource Description Framework (RDF) is the standard knowledge representation language for the Semantic Web, an evolution of the World Wide Web that aims to provide a well-founded infrastructure for publishing, sharing and querying structured data. This article provides an introduction to RDF and its related vocabulary definition language RDF Schema, and explains its relationship with the OWL Web Ontology Language. Finally, it provides an overview of the historical development of RDF and related languages for Web metadata.

INTRODUCTION

The Resource Description Framework [1], or RDF, is a knowledge representation language for the Semantic Web [2,3], and is used to express knowledge about things both on and off the Web; RDF can be used to write metadata about web pages and to describe real-world objects with equal facility. RDF is a key language in the technical architecture of the Semantic Web, being both a format for describing instance data, and the foundation for the vocabulary definition languages RDF Schema [4] and OWL [5].

The Semantic Web is a development of the World Wide Web that aims to provide an infrastructure for machine-understandable information on the Web. By expressing information in a form that makes the meaning, or semantics, accessible to machines, the goal of the Semantic Web is to create a next-generation Web in which information can be mediated by software agents on behalf of and without intervention from their users. At present, information on the World Wide Web is predominantly textual and requires human understanding to interpret it; the meaning of such natural language information must be made explicit if a machine is to be able to make sense of it. In artificial intelligence, the long-standing discipline of knowledge representation has developed a variety of approaches for expressing information to facilitate reasoning; these knowledge representation languages are of varying degrees of complexity and expressivity, but they share the common feature that they encode information as structures that can be formally interpreted to give some sense of their meaning.

By itself, RDF provides a domain-neutral framework for information interchange that must be augmented with terms to meet the requirements of a specific application domain. For example, a bibliographic application needs to be able to talk about works

and authors, while an e-commerce application needs to be able to talk about orders and prices. This orientation to a particular application domain or domains is accomplished by means of one or more domain vocabularies, or *ontologies* [6]. In Computer Science, an ontology is a formal description of a domain of knowledge. Gruber [7] defines an ontology as “a specification of a conceptualisation”; in other words, a formal account of an abstract model of some application domain which includes, but is not limited to, the classes of entity that exist within the domain, and the relations that link those entities.

RDF has two accompanying ontology languages, RDF Schema [4] and the Web Ontology Language OWL [5]. RDF Schema is the simpler of the two, and provides a small set of modelling constructs for defining classes, properties, and certain global constraints on those properties. OWL is more expressive (and correspondingly more complex to reason with), and provides a larger set of modelling constructs that allows an ontology designer to more richly specify classes in terms of the necessary and sufficient conditions for membership of those classes. An in-depth treatment of OWL is beyond the scope of this article; we refer the reader to [8,9] for further information.

In addition to these languages for defining RDF vocabularies, the World Wide Web Consortium has also published an SQL-like query language for interrogating RDF graphs, known as the SPARQL Protocol and RDF Query Language [10,11] (or more simply as SPARQL).

THE TRIPLE MODEL

The information presented on the Semantic Web can be viewed as a directed, labelled graph in which the nodes stand for things and values, and the edges (with their labels) are the properties that relate nodes to one other. The fundamental unit of knowledge representation is the triple, a directed edge that represents a binary relationship between two nodes. Customarily, the components of this triple are referred to as the subject, the predicate and the object: the subject is the node that the directed edge leaves, the object is the node that it enters, and the predicate is the type of the edge. Alternatively, a triple could be considered to be a statement that a particular entity (the subject) has a property (the predicate) with a particular value (the object); this corresponds closely to object-oriented [12] or frame-based [13] models of knowledge representation.

The nodes in an RDF graph denote resources, which are things in the domain of discourse. Nodes may be Uniform Resource Identifiers (URIs) [14], literals, or blank nodes. URIs are used to denote things with identity, such as web pages, organisations, people, or books. Literals are values such as character strings, numbers or dates, which do not have identity; a (plain) literal is considered to denote itself, so two literals with the same value are treated as if they were the same literal. Finally, blank nodes are used to indicate the existence of a thing, without saying anything about the name of that thing.

With the careful use of URIs, RDF allows machines to unambiguously identify entities and vocabulary terms; each URI identifies a single thing, even though it is expected that there may be multiple URIs that identify the same thing (effectively synonyms for the thing). Adapting an example from the RDF Primer [15], we can

represent the statement (that a particular webpage was created by John Smith) as a triple with:

- a subject, <http://www.example.org/index.html>
- a predicate, <http://purl.org/dc/terms/creator>
- and an object, <http://www.example.org/staffid/85740>

In this example, we refer to the webpage using its URI:

<http://www.example.org/index.html>. We also use URIs to refer to John Smith

(<http://www.example.org/staffid/85740>), and to the creator relation

(<http://purl.org/dc/terms/creator>, taken from the Dublin Core vocabulary [16].

Presented graphically, this statement forms a graph containing two nodes with a single directed edge between them, as shown in Figure 1.

This triple can also be written as:

```
<http://www.example.org/index.html> <http://purl.org/dc/terms/creator>  
<http://www.example.org/staffid/85740> .
```

This example uses a syntax based on the Notation 3 syntax for RDF [17] (discussed in more depth below), in which terms in angle brackets denote resources identified by URIs. In the order written, the three terms are the subject, predicate and object of the triple, and the completed triple is followed by a period.

COMPARING XML AND RDF

The relationship between RDF and the eXtensible Markup Language (XML) [18] warrants further explanation. Both are used to represent structured information on the Web, and XML is used as the standard syntax for RDF, but the two languages are distinct and have different characteristics.

The most fundamental difference between XML and RDF is in the way in which they structure information. As described above, RDF structures information as a labelled, directed graph, whereas XML is limited to tree structures, which has consequences in the way that data from different sources can be merged. Merging two RDF files can be accomplished by taking the union of the underlying graphs; this operation does not require any knowledge of the vocabularies (schemas) that are used in these graphs. In contrast, merging two XML files that use different schemas is a more difficult operation, because the resulting file will not validate using either schema, but rather a new schema that combines the two original schemas must be used instead.

An XML document can be parsed to give an unambiguous structure, but this structure has no innate meaning; the meanings of XML schemas must be agreed between the relevant parties on a case-by-case basis. RDF differs from XML in that the triples that are created when an RDF file is parsed have a well-defined semantics, which facilitates further information exchange by providing a basic set of rules for manipulating the information while preserving its knowledge content. Together with the ease of merging RDF files, these formal semantics mean that RDF has considerable advantages for information interchange over XML, and can be used to facilitate interoperability between heterogeneous systems.

Finally, RDF uses URIs to refer to objects in the domain of discourse. This has been used to promote what is known as the linked data model of publishing [19,20]. Given that the resources in an RDF graph may be represented by URIs, the aim of the linked data model is that it should be possible to dereference those URIs and by doing so retrieve a description of the resources that they represent. In effect, linked data results from the hypertextual foundations of the World Wide Web (and so of the Semantic Web); users of the Web follow links to obtain human-readable information in the form of web pages, so Semantic Web agents should be able to do likewise. Linked data may be implemented using several different Web technologies; a general overview of the more common approaches is given in [21], and a summary of the best practices for publishing ontologies as linked data is given in [22].

FEATURE SUMMARY

RDF defines the triple model of data and a standard syntax for expressing those triples (RDF/XML), but does not constrain the vocabulary used in those triples: the URIs denoting categories and relations that we might wish to use in our descriptions. RDF Schema defines a basic vocabulary for use with RDF that lets users represent and define vocabularies that are specific to their problem domains. This basic vocabulary supports the following primitive modelling constructs:

- Named classes and explicit hierarchies of classes;
- Named properties and explicit hierarchies of properties;
- Global ranges and domains of properties;

- A reification vocabulary for using triples to describe other triples;
- Containers and collections, two methods for representing information about groups of resources;
- A set of ancillary properties for providing human-readable documentation and links to further information.

All of the vocabulary terms defined in RDF Schema are assigned URIs from one of two namespaces, `rdf` and `rdfs` (for largely historical reasons that originate with the first version of RDF, the terms defined in RDF Schema are divided across two different namespaces). For clarity, we will write `rdf:XXX` as a shorthand for `http://www.w3.org/1999/02/22-rdf-syntax-ns#XXX`, and `rdfs:YYY` as a shorthand for `http://www.w3.org/2000/01/rdf-schema#YYY`. This abbreviation style, whereby a lengthy URI is reduced to a short string consisting of a namespace abbreviation (representing a URI prefix), a separating colon, and a name defined in that namespace (which is to be appended to the URI prefix), is known as a Qualified Name or QName [23,24].

CLASSES

RDF Schema defines the property `rdf:type` that allows a user to indicate that a resource is an instance of a particular class. For example, if we wished to indicate that John Smith (denoted by the URI `http://www.example.org/staffid/85740`) was an instance of the `Person` class defined in the FOAF (Friend of a Friend) ontology [25], we would write the triple:


```
<http://www.example.org/staffid/85740> rdf:type <http://xmlns.com/foaf/0.1/Person>
```

.

In this example, `rdf:type` is a QName which represents the ‘type’ from the `rdf` namespace; the QName is expanded to the URI `http://www.w3.org/1999/02/22-rdf-syntax-ns#type` as described above. We can define a resource to be a class by declaring it to have an `rdf:type` of `rdfs:Class`. For example, to define a class of documents (represented by the URI `http://example.org/ontology#Document`), we would write:

```
<http://example.org/ontology#Document> rdf:type rdfs:Class .
```

These classes may be arranged into an explicit hierarchy by stating that a class is a subclass of another class, using the `rdfs:subClassOf` property:

```
<http://example.org/ontology#Newspaper> rdfs:subClassOf  
<http://example.org/ontology#Document> .
```

In this example, we have stated that the class `Newspaper` is a subclass of the class `Document`, which is equivalent to saying that every newspaper is also a document (or to put it a third way, that the set of newspapers is a subset of the set of documents).

PROPERTIES

In the same way that classes are defined in RDF Schema, properties may be defined by declaring a resource to be of `rdf:type rdf:Property`. Properties may be arranged into hierarchies to represent the specialisation and generalisation of relationships. For example, an “employed by” property might be considered to be a specialisation of an “affiliated with” property. A property can be declared to be a specialisation of another property by stating that it is `rdfs:subPropertyOf` the other.

```
<http://example.org/ontology#employedBy> rdfs:subPropertyOf  
<http://example.org/ontology#affiliatedWith> .
```

Properties can be further described by defining their domain and range using the `rdfs:domain` and `rdfs:range` properties. The domain of a property is the class of objects that it runs from. For example, the “employed by” property might run from a class `Person` (being the class of people). The range of a property is the class of objects that it runs to; the range of “employed by” might be the class of organisations. These constraints on properties permit a reasoner to make inferences about the types of objects that are involved in a relationship.

```
<http://example.org/ontology#employedBy> rdfs:domain  
<http://example.org/ontology#Person> .  
<http://example.org/ontology#employedBy> rdfs:range  
<http://example.org/ontology#Organisation> .
```

CONTAINERS AND COLLECTIONS

RDF Schema provides two different mechanisms for defining groups of objects, each with separate properties. The first of these, RDF containers, represents a group of objects with a resource which may be typed to informally indicate the intended semantics of the group; a group of type `rdf:Bag` is unordered, one of type `rdf:Seq` is an ordered sequence, and one of type `rdf:Alt` is a set of alternatives from which a single member should be selected (in practice, `rdf:Alt` is rarely used). The vocabulary pertaining to containers includes a countably infinite set of special properties (the ordinal container membership properties `rdf:_1`, `rdf:_2` and so on) which indicate both the membership of a resource in the group, and for the ordered `rdf:Seq`, the position of a member within the group.

A sequence of three resources (for example, three email addresses) would be represented as follows:

```
<http://example.org/data#seq1> rdf:type rdf:Seq .  
<http://example.org/data#seq1> rdf:_1 <mailto:groucho@example.org> .  
<http://example.org/data#seq1> rdf:_2 <mailto:chico@example.org> .  
<http://example.org/data#seq1> rdf:_3 <mailto:harpo@example.org> .
```

The container vocabulary has the disadvantage that it produces groups that are open and mutable. There is no way to state that a group is closed, that is that the stated members of a group are the only members; a third party could assert that the above sequence has an extra member by adding the following triple:

```
<http://example.org/data#seq1> rdf:_4 <mailto:karl@example.org> .
```

The collection vocabulary addresses this by supporting the definition of closed groups that can have no further members. A collection is a recursive data structure that defines a group as a resource of type `rdf:List`, which consists of the first object in the group (indicated using the `rdf:first` property), and another `rdf:List` which contains the remainder of the group (indicated using the `rdf:rest` property). This second `rdf:List` contains the second element of the collection, and another `rdf:List` with the remainder of the collection, and so on. This recursion is terminated by the term `rdf:nil`, which represents an empty list. The example above could be represented in the container vocabulary as follows:

```
<http://example.org/data#list1> rdf:type rdf:List .
<http://example.org/data#list1> rdf:first <mailto:groucho@example.org> .
<http://example.org/data#list1> rdf:rest <http://example.org/data#list2> .
<http://example.org/data#list2> rdf:type rdf:List .
<http://example.org/data#list2> rdf:first <mailto:chico@example.org> .
<http://example.org/data#list2> rdf:rest <http://example.org/data#list3> .
<http://example.org/data#list3> rdf:type rdf:List .
<http://example.org/data#list3> rdf:first <mailto:harpo@example.org> .
<http://example.org/data#list3> rdf:rest rdf:nil .
```

As can be seen, an RDF collection requires more triples to represent the same group of objects than does a container, but it has the property that an attempt to tamper with the collection by adding extra members will result in an ill-formed data structure in which one of the component `rdf:Lists` has more than one `rdf:first` or `rdf:rest` property.

REIFICATION

Reification is a mechanism for treating triples as first-class objects about which other assertions can be made (that is, as resources which can be referred to as the subjects or objects of other triples). The reification vocabulary in RDF Schema introduces the class `rdf:Statement`, which represents triples, and the properties `rdf:subject`, `rdf:predicate` and `rdf:object`, which represent the three components of a triple.

If we were to reify our original example (that a webpage was written by John Smith), we would construct a small RDF graph containing four triples: one to represent the type of the reified statement, and one for each of the three components.

```
<http://example.org/data#stmt1> rdf:type rdf:Statement .
```

```
<http://example.org/data#stmt1> rdf:subject <http://www.example.org/index.html> .
```

```
<http://example.org/data#stmt1> rdf:predicate <http://purl.org/dc/terms/creator> .
```

```
<http://example.org/data#stmt1> rdf:object <http://www.example.org/staffid/85740>
```

```
.
```

In this example, the statement itself is represented by the URI `http://example.org/data#stmt1`; this may be used in other triples to express information about the statement. For example, if John Smith wanted to assert that he was the author of the statement, he could write:

<http://example.org/data#stmt1> <http://purl.org/dc/terms/creator>
<http://www.example.org/staffid/85740> .

The reification vocabulary provides a simple framework that supports the expression of fine-grained (triple level) provenance information about Semantic Web data.

However, RDF reification is rarely used in practice, with many RDF practitioners preferring to use named graphs (a set of triples that is identified by a URI) to achieve the same ends, that of making assertions about other assertions.

ANCILLARY PROPERTIES

Finally, RDF Schema contains a number of properties that are not used to define ontology concepts, but to provide human-readable documentation, or to link to related information. The `rdfs:label` property is used to attach a human-readable label to a resource (necessary, given that URIs may be verbose), while `rdfs:comment` is used to add a comment (a natural language definition or rationale, perhaps). The `rdfs:seeAlso` property is used to indicate another resource which contains further information about a given resource; this is specialised by the `rdfs:isDefinedBy` property, which can be used to indicate the web resource that defines a particular resource (for example, an ontology).

SYNTAX

The standard (normative) syntax for serialising RDF graphs for storage or transmission is RDF/XML [24]. This syntax is defined by a set of rules for

transforming an XML document into an RDF graph [18]; element names, element contents, attribute names and attribute values are turned into resources, predicates and literals. While this syntax builds on the availability of high-quality XML tools, it tends to verbose serialisations; in situations where RDF is to be read (and written) by people rather than machines, other non-XML syntaxes are preferred by many practitioners.

The best known of these alternative syntaxes is Notation 3 (N3) [17], a non-XML format that produces easy-to-understand serialisations that can be quickly written by hand (or reproduced compactly in printed materials). Although it is well-known, N3 is a work in progress and subject to change, and has ambitious design goals; the expressivity of N3 goes beyond that of both RDF and OWL, so an arbitrary document written in N3 might contain constructs that cannot be translated into RDF/XML. For this reason, a number of N3 variants have evolved. N-Triples [26] is the simplest of these, and represents triples with no abbreviations such as QNames. Turtle [27] includes the QName-based namespace and abbreviation mechanism used by N3, but restricts the expressivity of the language to match that of RDF/XML.

Finally, there are two different approaches to serialising RDF graphs by embedding them within XHTML documents; the intent with both is to expose the structured data in ordinary web pages in such a way that Semantic Web agents can recognise and extract that data for their own use. The first of these, RDFa [28] defines a convention for expressing triples using the attributes on the elements in an XHTML document. A SW agent, on receiving a document containing RDFa markup, would first parse it as XML to obtain an intermediate representation, such as that provided by the Document

Object Model (DOM), then parse this intermediate representation to obtain the encoded triples. GRDDL [29], the second technique, provides a method for attaching a transformation to an XML document; this transformation, typically written in the eXtensible Stylesheet Language (XSLT) [30], can be used to extract selected information from the XML document and turn it into an RDF/XML document which can then be parsed to yield the RDF triples.

All of these RDF syntaxes have the feature that they do not define a canonical way in which an RDF graph can be written to a flat file for storage (a process known as serialisation). Because a graph has no beginning or end, and no order to the edges, two implementations of the same serialisation algorithm may process the triples in the graph in different orders, producing two different sequences of characters in the flat file. This complicates the process of comparing RDF data; comparison must be performed on the graphs themselves, rather than on their serialisations.

SEMANTICS

RDF has a formal semantics [31] that defines the meaning of RDF statements by means of a model theory, an abstract, mathematical account based on the truth or falsity of statements. The goal of a model theory is that it enables the determination of the soundness and completeness of an inference process [32].

The RDF semantics permit certain new triples to be inferred from those in an RDF graph. The following list of inferences is not comprehensive, but rather an indicative list of the sorts of inference that RDF permits:

- Transitivity of the `rdfs:subClassOf` relation;
(if A is a subclass of B, and B is a subclass of C, then A is a subclass of C)
- Transitivity of the `rdfs:subPropertyOf` relation;
(if A is a subproperty of B, and B is a subproperty of C, then A is a subproperty of C)
- Inferring general properties from more specific properties;
(if two resources are related by a property, and there is a generalisation of that property, then the resources are related by the superproperty)
- Inferring membership of general classes from membership of more specific classes;
(if a resource is an instance of a class A, and A is a subclass of B, then the resource must also be an instance of class B)
- Inferring membership of classes from domain and range constraints.
(if a resource has a property whose domain is B, then the resource must be an instance of B)

COMPARISON OF RDF SCHEMA AND OWL

RDF Schema and OWL both fulfil similar roles within the architecture of the Semantic Web, in that they both support the definition of formal vocabularies for classifying entities and specifying the natures of their inter-relations. The key difference between the two approaches is one of expressiveness; RDF Schema provides a concise set of primitives that are familiar to users of object-oriented

modelling methodologies, but these primitives are limited to expressing explicit subclass and superclass relationships between classes and properties.

The OWL language extends the modelling capabilities of RDF Schema using a family of formal logic known as description logics [33] in order to produce a more expressive ontology language that allows the user to specify the nature of a class in terms of the necessary and sufficient conditions that must be met by members of that class. Necessary conditions are those which must be satisfied by all resources that are members of a class, while sufficient conditions allow an OWL reasoner to infer that a resource that satisfies them must be a member of a the class. For example, “is an animal with four legs” is a necessary condition for the class Horse (if an entity is a horse, we can infer that it is an animal and that it has four legs), but not a sufficient condition (if an entity is an animal and has four legs, we cannot be certain that it is a horse, because things other than horses will satisfy those conditions).

If classes are defined in terms of their necessary and sufficient conditions for membership, it is possible to identify the implicit subclass hierarchies that must naturally follow from these definitions. For example, if the class Quadruped is defined with the sufficient condition that it is an animal with four legs (that is, if we have a four-legged animal, then we can infer that it is a quadruped), then we could infer that Horse is a subclass of Quadruped (every Horse is a Quadruped), even though this relationship has not been explicitly stated in our ontology.

HISTORY

RDF and RDF Schema have a strong lineage which draws on the knowledge representation formalisms developed over the past forty years. The graph-oriented triple model resembles the approaches made by network knowledge representation techniques (principally semantic networks [34] and conceptual graphs [35]). Although the simple class and property model of RDF Schema owes much to object-oriented approaches to software engineering, which in turn were derived in part from Minsky's frames [36], RDF Schema is a purely declarative representation (unlike object-oriented models with member functions on objects, or frames with demons attached to slots). There is a further difference between the approach taken by RDF, and that taken by traditional object-oriented languages such as C++ or Java. The former is a property-centric approach, in that classes are associated with properties by defining the global range and domain constraints on the properties, while the latter is more class-centric, and defines classes in terms of the properties that may appear on objects of that category. This is largely a difference of emphasis rather than of the fundamental underlying logic, however. It should be noted that OWL takes an approach closer to that of the frame-based systems, in that classes are defined in terms of the necessary and sufficient properties that their instances must have.

The lineage of RDF in the context of Web-based knowledge representation and metadata languages dates back to the mid 1990s, with the publication of the Meta Content Framework (MCF) and the Platform for Internet Content Selection (PICS). The two languages prepared the way for RDF, by introducing a key use case (web page description) and the notion of a triple. The first version of RDF was published as a W3C Recommendation in 1999, and this version was deprecated by a revision in

2004 which sought to provide a formal foundation for the more expressive ontology language OWL; the description of RDF in this article is based on the 2004 standard.

PLATFORM FOR INTERNET CONTENT SELECTION

PICS, the Platform for Internet Content Selection [37,38], is a W3C Recommendation that was designed as a means for attaching machine-readable ratings to web pages in order to inform filtering software of the nature of the content of the web pages, with the intent that such filtering software would better be able to control the content displayed to users.

The technology emerged in part as a response to the US Communications Decency Act of 1996 (Sec. 502), which effectively sought to place censorship restrictions on material distributed via the Internet. Rather than place the onus for removing unacceptable material on the publisher, PICS proposed a model whereby a potential consumer of such material could obtain an accurate description of the material that would enable them to decide whether or not they wished to view the material. The descriptions, known as content labels, consisted of values taken from some rating system, where a rating system comprised a set of independent dimensions on which an Internet resource could be evaluated. Such rating systems can be considered to be more complex versions of those used for film classification (for example, the film ratings G, PG, PG-13, R and NC-17 specified by the Motion Picture Association of America); the commonly used ICRA system [39] specified by the Family Online Safety Institute classifies works on seven different scales: nudity, sexual material, violence, language, potentially harmful activities, user-generated content, and context.

PICS originally used a non-XML (Lisp-like) syntax for expressing content labels, with specific ratings represented by numeric values; a rating system might specify a rating category (dimension) for language, with ratings for ‘slang’ given the value 0, ‘mild expletives’ the value 1, ‘expletives’ the value 2, ‘obscene gestures’ the value 3, and so on. A content label also included information about the party that had generated the metadata, and could be digitally signed to support the verification of the identity of the rating party.

Although the practice of attaching PICS metadata to web pages was never widespread, several Web browsers implemented the standard, including Internet Explorer (from version 3.0 onwards). The principle of a standard syntax for Web metadata was a key step towards the advent of the Semantic Web, even if PICS was too tightly coupled to its application domain to serve well as a general-purpose metadata language. In 2000, W3C published a proposal for the use of RDF as the underlying syntax for content labels, which specified an RDF Schema vocabulary for expressing both content labels and rating systems [40].

META CONTENT FRAMEWORK

MCF (the Meta Content Framework) [41] originated at Apple Computer Inc. in 1995 as a metadata format for representing the content and structure of websites, and was used by the HotSauce web navigation system (a browser that allowed a user to fly through a 3D map of a web site). The author of MCF, Ramanathan V. Guha, moved from Apple to the Netscape Communications Corporation in 1997, and

reimplemented MCF as an application of the emerging XML dialect of SGML. This XML syntax for MCF has been an important influence on the development of RDF, and introduced a number of features that persist in the most recent version of RDF and its XML syntax.

The underlying model of an MCF description is a directed labelled graph much like that of RDF. The nodes in the graph can represent both literal values and objects with identity (which could be either web objects or real-world objects), and are linked by typed binary relations. Unlike RDF, it assumes that only Web-addressable objects have URLs; real-world objects are identified by non-URI strings.

RDF (1999 VERSION)

The original 1999 specifications of RDF [42] and RDF Schema [43] differ from the versions described above in three main ways. First, they do not contain a formal account of the semantics of the languages that would be suitable for reasoning. The 1999 RDF specification contains a definition of a triple and an explanation of the entailed triples produced by reification, and the RDF Schema specification informally defines `rdfs:subClassOf` and `rdfs:subPropertyOf` as transitive properties, but there is no model-theoretic account of the sort that would be needed to verify the soundness and completeness of a reasoning engine.

Secondly, they contain a number of features that were deprecated in the 2004 specification because they could not be given an unambiguous meaning. Chief amongst these features is the ability to express statements about the members of a

container. Given an `rdf:Bag` containing a number of resources, it was possible to write a single triple which applied to each member of the Bag by using the `rdf:aboutEach` attribute instead of the `rdf:about` attribute to indicate the subject of the triple. While this could be seen as a convenient shorthand for expressing repetitive information, there is an unfortunate interaction with the non-closed nature of RDF containers. In the absence of a framework for provenance, it is possible for a third party to state that a container has extra members; it thus becomes possible to make the original claim about the members of the container apply to more resources than were intended. A similar construct, `rdf:aboutEachPrefix`, allowed an author to express statements that would hold for all resources whose URIs began with a particular string. The set of triples entailed by this could not be determined without detailed knowledge of all the URIs with a particular prefix, and so the construct would therefore have unpredictable consequences as web pages were created and deleted over time. These two constructs, `rdf:aboutEach` and `rdf:aboutEachPrefix`, were included in the 1999 specification in order to support certain functionality provided by PICS, and in doing so to facilitate the transition from PICS to RDF. The 2004 specification also clarified the formal basis for reification; while the reification vocabulary remains part of the language, a triple does not entail its reification, nor vice versa.

Finally, the 2004 specifications contain a number of new constructs that were intended to bolster those constructs of RDF-1999 that were felt to have awkward consequences, but which were too widely used to be deprecated; the RDF container vocabulary was considered to be such, and was supplemented with the collection construct which enforced a well-formedness criterion on closed lists.

The RDF revision in 2004 aimed to make a number of corrections and improvements following feedback from implementers, and to complete the work on RDF Schema, which had previously only reached the state of W3C Candidate Recommendation. This new version of RDF was developed contemporaneously with the OWL Web Ontology Language, and so there was a close dialogue between the relevant W3C working groups in order to ensure that the revised form of RDF would form a suitable foundation for OWL; the earlier language DAML+OIL [44], from which OWL was derived, used RDF/XML as a serialisation syntax but disregarded its semantics (or lack thereof) and adopted a conventional description logic semantics [33]. An aim of the collaboration between these working groups was therefore to develop a semantics for RDF and RDF Schema that would be compatible with the semantics of OWL.

CONCLUSION

The Resource Description Framework is a versatile knowledge representation language that can be used to express information about both Web resources and real-world objects. In conjunction with the vocabulary description languages RDF Schema and OWL, both of which are built on RDF, it can be used to create complex descriptions of application domains that a computer may use to infer new knowledge. The role played by RDF in the development of the Semantic Web is crucial; it integrates with the existing Web architecture to provide a common infrastructure for the creation and publication of structured data.

REFERENCES

1. Klyne, G.; Carroll, J.J.; McBride, B. Resource Description Framework (RDF): concepts and abstract syntax, World Wide Web Consortium, 2004, <http://www.w3.org/TR/rdf-concepts/> (accessed February 2008).
2. Berners-Lee, T.; Hendler, J.; Lassila, O. The Semantic Web, *Scientific American*, May 2001, 29-37.
3. Shadbolt, N.; Hall, W.; Berners-Lee, T. The Semantic Web Revisited, *IEEE Intelligent Systems*, May/June 2006, 96-101.
4. Brickley, D.; Guha, R.V. RDF Vocabulary Description Language 1.0: RDF Schema, World Wide Web Consortium, 2004, <http://www.w3.org/TR/rdf-schema/> (accessed February 2008).
5. McGuinness, D.L.; van Harmelen, F. OWL Web Ontology Language Overview, World Wide Web Consortium, 2004, <http://www.w3.org/TR/owl-features/> (accessed February 2008).
6. Staab, S.; Studer, R., Eds., *Handbook on Ontologies*, 2nd Ed.; Springer: Berlin, 2009.
7. Gruber, T.R. A translation approach to portable ontology specifications, *Knowledge Acquisition* **1993**, 5(2), 199-220.
8. Dean, M.; Schreiber, G.; Bechhofer, S.; van Harmelen, F.; Hendler, J.; Horrocks, I.; McGuinness, D.L.; Patel-Schneider, P.F.; Stein, L.A. OWL Web Ontology Language Reference, World Wide Web Consortium, 2004, <http://www.w3.org/TR/owl-ref/> (accessed February 2008).
9. Horrocks, I.; Patel-Schneider, P.F.; van Harmelen, F. From SHIQ and RDF to OWL: the making of a Web Ontology Language, *Web Semantics: Science, Services and Agents on the World Wide Web* **2003**, 1, 7-26.

10. Prud'hommeaux, E.; Seaborne, A. SPARQL Query Language for RDF, World Wide Web Consortium, 2008, <http://www.w3.org/TR/rdf-sparql-query/> (accessed February 2008).
11. Clark, K.G.; Feigenbaum, L.; Torres, E. SPARQL Protocol for RDF, World Wide Web Consortium, 2008, <http://www.w3.org/TR/rdf-sparql-protocol/> (accessed February 2008).
12. Information technology – Open Distributed Processing – Unified Modelling Language (UML) Version 1.4.2, ISO/IEC 19501:2005.
13. Minsky, M. A Framework for Representing Knowledge. In *The Psychology of Computer Vision*; Winston, P.H., Ed.; McGraw-Hill: New York, 1975; 211-277.
14. Berners-Lee, T.; Fielding, R.; Masinter, L. Uniform Resource Identifier (URI): Generic Syntax, RFC3986, Internet Engineering Task Force, 2005, <http://www.ietf.org/rfc/rfc3986.txt> (accessed February 2008).
15. Manola, F.; Miller, E. RDF Primer, World Wide Web Consortium, 2004, <http://www.w3.org/TR/rdf-primer/> (accessed February 2008).
16. DCMI Metadata Terms, 2008, <http://dublincore.org/documents/dcmi-terms/> (accessed February 2008).
17. Berners-Lee, T. Notation 3, 2006, <http://www.w3.org/DesignIssues/Notation3.html> (accessed February 2008).
18. Bray, T.; Paoli, J.; Sperberg-McQueen, C.M.; Maler, E.; Yergeau, F. Extensible Markup Language (XML) 1.0 (Fourth Edition), World Wide Web Consortium, 2006, <http://www.w3.org/TR/xml> (accessed February 2008).
19. Berners-Lee, T. Linked Data, 2007, <http://www.w3.org/DesignIssues/LinkedData.html> (accessed February 2008).

20. Bizer, C.; Heath, T.; Idehen, K.; Berners-Lee, T. Linked Data on the Web, In Proceedings of the 17th International World Wide Web Conference (WWW2008), Beijing, China, April 21-25, 2008; Association of Computing Machinery: New York, NY, 2008.
21. Sauermaun, L.; Cyganiak, R.; Ayers, D.; Völkel, M. Cool URIs for the Semantic Web, World Wide Web Consortium, 2008, <http://www.w3.org/TR/cooluris/> (accessed April 2008).
22. Berrueta, D.; Phipps, J. Best Practice Recipes for Publishing RDF Vocabularies, World Wide Web Consortium, 2008, <http://www.w3.org/TR/swbp-vocab-pub/> (accessed February 2008).
23. Bray, T.; Hollander, D.; Layman, A.; Tobin, R. Namespaces in XML 1.0, World Wide Web Consortium, 2006, <http://www.w3.org/TR/REC-xml-names/> (accessed February 2008).
24. Beckett, D. RDF/XML Syntax Specification (Revised), World Wide Web Consortium, 2004, <http://www.w3.org/TR/rdf-syntax-grammar/> (accessed February 2008).
25. Brickley, D.; Miller, L. FOAF Vocabulary Specification, 2007, <http://xmlns.com/foaf/spec/> (accessed February 2008).
26. Grant, J.; Beckett, D. RDF Test Cases, World Wide Web Consortium, 2004, <http://www.w3.org/TR/rdf-testcases/> (accessed February 2008).
27. Beckett, D.; Berners-Lee, T. Turtle – Terse RDF Triple Language, World Wide Web Consortium, 2008, <http://www.w3.org/TeamSubmission/turtle/> (accessed February 2008).

28. Adida, B.; Birbeck, M. RDFa Primer: Embedding Structured Data in Web Pages, Working Draft, World Wide Web Consortium, 2007, <http://www.w3.org/TR/xhtml-rdfa-primer> (accessed February 2008).
29. Connolly, D. Gleaning Resource Descriptions from Dialects of Languages (GRDDL), World Wide Web Consortium, 2007, <http://www.w3.org/TR/grddl/> (accessed February 2008).
30. Clark, J. XSL Transformations (XSLT) Version 1.0, World Wide Web Consortium, 1999, <http://www.w3.org/TR/xslt> (accessed February 2008).
31. Hayes, P. RDF Semantics, World Wide Web Consortium, 2004, <http://www.w3.org/TR/rdf-mt/> (accessed February 2008).
32. Hunter, G. Metalogic: An Introduction to the Metatheory of Standard First Order Logic, University of California Press: Berkeley, CA, 1973.
33. Baader, F.; Nutt, W. Basic Description Logics. In *The Description Logic Handbook: Theory, Implementation and Applications*, 2nd Ed.; Baader, F.; Calvanese, D.; McGuinness, D.L.; Nardi, D.; Patel-Schneider, P.F., Eds.; Cambridge University Press: Cambridge, UK, 2007; 47-104.
34. Quillian, M.R. Word Concepts: A Theory and Simulation of Some Basic Semantic Capabilities, *Behavioral Science* **1967**, 12, 410-430.
35. Sowa, J.F. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks Cole Publishing Co.: Pacific Grove, CA, 2000.
36. Minsky, M. A Framework for Representing Knowledge. In *Readings in Cognitive Science*; Collins, A.; Smith, E.E., Eds. Morgan-Kaufmann: San Francisco, CA, 1992.

37. Resnick, P. Filtering Information on the Internet, *Scientific American*, 1997, 276(3), 62-64.
38. Miller, J.; Resnick, P.; Singer, D. Rating Services and Rating Systems (and Their Machine Readable Descriptions), World Wide Web Consortium, 1996, <http://www.w3.org/TR/REC-PICS-services> (accessed February 2008).
39. The ICRA Vocabulary, <http://www.icra.org/vocabulary/> (accessed February 2008).
40. Brickley, D.; Swick, R. PICS Rating Vocabularies in XML/RDF, World Wide Web Consortium, 2000, <http://www.w3.org/TR/rdf-pics> (accessed February 2008).
41. Guha, R.V.; Bray, T. Meta Content Framework Using XML, World Wide Web Consortium, 1997, <http://www.w3.org/TR/NOTE-MCF-XML/> (accessed February 2008).
42. Lassila, O.; Swick, R. Resource Description Framework (RDF) Model and Syntax Specification, World Wide Web Consortium, 1999, <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/> (accessed February 2008).
43. Brickley, D.; Guha, R.V. Resource Description Framework (RDF) Schema Specification, World Wide Web Consortium, 1999, <http://www.w3.org/TR/1999/PR-rdf-schema-19990303/> (accessed February 2008).
44. Connolly, D.; van Harmelen, F.; Horrocks, I.; McGuinness, D.L.; Patel-Schneider, P.F.; Stein, L.A. DAML+OIL Reference Description, World Wide Web Consortium, 2001. <http://www.w3.org/TR/daml+oil-reference> (accessed February 2008)

FURTHER READING

Antoniou, G.; van Harmelen, F. *A Semantic Web Primer*, 2nd Ed.; MIT Press: Cambridge, MA, 2008.

Berners-Lee, T. *Weaving the Web: the past, present and future of the World Wide Web by its inventor*; Texere Publishing: London, 1999.

Fensel, D. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, 2nd Ed.; Springer: Berlin, 2004.

Fensel, D.; Hendler, J.; Lieberman, H.; Wahlster, W. *Spinning the Semantic Web: bringing the World Wide Web to its full potential*; MIT Press: Cambridge, MA, 2003.

Fig. 1: An RDF graph containing a single triple.