Resource Description Framework (RDF) Schema Specification

W3C Proposed Recommendation 03 March 1999

This Version:
    http://www.w3.org/TR/1999/PR-rdf-schema-19990303
Newest Version:
    http://www.w3.org/TR/PR-rdf-schema

Editors:
    Dan Brickley, University of Bristol
    R.V. Guha, Netscape

Document Status and errata
Acknowledgments

Status of this document

This document is a Proposed Recommendation of the World Wide Web Consortium. Review comments on this specification should be sent by 2359Z April 06, 1999 to <www-rdf-comments@w3.org>. The archive of public comments is available at http://w3.org/Archives/Public/www-rdf-comments. W3C Members may send their formal comments, visible only to W3C staff, to <w3c-rdf-review@w3.org>.

This specification is a revision of the last-call working draft dated 1998-10-30 incorporating suggestions received in review comments and further deliberations of the W3C RDF Schema Working Group. The detailed differences are available for reviewers to compare.

The Working Group anticipates no further substantial changes to this specification. We encourage active implementation to test this specification during the Proposed Recommendation review period.

Publication as a Proposed Recommendation does not imply endorsement by the W3C membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite W3C Drafts as other than "work in progress".

The Resource Description Framework is part of the W3C Metadata Activity. The goal of this activity, and of RDF specifically, is to produce a language for the exchange of machine-understandable descriptions of resources on the Web. A separate specification describes the data model and syntax for the interchange of metadata using RDF.

Note: The HTML source of this document contains embedded RDF and will therefore not validate against the HTML4.0 DTD. A solution for those requiring DTD-style validation services may come from future W3C work.

Table of Contents

RDF Schemas

1. Introduction

The Resource Description Framework (RDF) is a foundation for processing metadata; it provides interoperability between applications that exchange machine-understandable information on the Web. RDF emphasizes facilities to enable automated processing of Web resources. RDF can be used in a variety of application areas; for example: in resource discovery to provide better search engine capabilities, in cataloging for describing the content and content relationships available at a particular Web site, page, or digital library, by intelligent software agents to facilitate knowledge sharing and exchange, in content rating, in describing collections of pages that represent a single logical "document", for describing intellectual property rights of Web pages, and for expressing the privacy preferences of a user as well as the privacy policies of a Web site. RDF with digital signatures will be key to building the "Web of Trust" for electronic commerce, collaboration, and other applications.

Descriptions used by these applications can be modelled as relationships among Web resources. The RDF data model, as specified in [RDFMS], defines a simple model for describing interrelationships among resources in terms of named properties and values. RDF properties may be thought of as attributes of resources and in this sense correspond to traditional attribute-value pairs. RDF properties also represent relationships between resources. As such, the RDF data model can therefore resemble an entity-relationship diagram. The RDF data model, however, provides no mechanisms for declaring these properties, nor does it provide any mechanisms for defining the relationships between these properties and other resources. That is the role of RDF Schema.

Resource description communities require the ability to say certain things about certain kinds of resources. For describing bibliographic resources, for example, descriptive attributes including "author", "title", and "subject" are common. For digital certification, attributes such as "checksum" and "authorization" are often required. The declaration of these properties (attributes) and their corresponding semantics are defined in the context of RDF as an *RDF schema*. A schema defines not only the properties of the resource (Title, Author, Subject, Size, Color, etc.) but may also define the kinds of resources being described (books, Web pages, people, companies, etc.).

This document does not specify a vocabulary of descriptive elements such as "author". Instead, it specifies the mechanisms needed to define such elements, to define the classes of resources they may be used with, to restrict possible combinations of classes and relationships, and to detect violations of those restrictions. Thus, this document defines a *schema specification language*. More succinctly, the RDF Schema mechanism provides a basic *type system* for use in RDF models. It defines resources and properties such as *Class* and *subClassOf* that are used in specifying application-specific schemas.

The typing system is specified in terms of the basic RDF data model - as resources and properties. Thus, the resources constituting this typing system become part of the RDF model of any description that uses them. The schema specification language is a declarative representation language influenced by ideas from knowledge representation (e.g. semantic nets, frames, predicate logic) as well as database schema specification languages (e.g. NIAM) and graph data models. The RDF schema specification language is less expressive, but much simpler to implement, than full predicate calculus languages such as CycL[CycL] and KIF [KIF].

RDF and the RDF Schema language were also based on metadata research in the the Digital Library community. In particular, RDF adopts a modular approach to metadata along the lines of the Warwick Framework [WF]. RDF represents an evolution of the Warwick Framework model in that the Warwick Framework allowed each metadata vocabulary to be represented in a different syntax. In RDF, all vocabularies are expressed within a single well defined model and syntax. This allows for a finer grained mixing of machine-processable vocabularies, and addresses the need [EXTWEB] to create metadata in which statements can draw upon multiple vocabularies that are managed in a decentralised fashion by various communities of expertise.

RDF Schemas might be contrasted with XML Document Type Definitions (DTDs). Unlike an XML DTD, which gives specific constraints on the structure of a document, an RDF Schema provides information about the interpretation of the statements given in an RDF data model. The RDF/XML syntax itself provides considerable flexibility in the syntactic expression of the data model. A syntactic schema alone is not sufficient for RDF purposes. RDF Schemas may also specify consistency constraints that should be followed by these data models.

1.1 Scope

The RDF Schema specification is not aimed at theoretical issues, but at solving a small number of immediate problems. Its creators expect that other problems (some of which are illustrated in the examples below) will share similar characteristics and that they also may be able to use the basic classes described in this specification.

The RDF Schema specification was directly influenced by consideration of the following problems:

1.1.1 Platform for Internet Content Selection (PICS)

The RDF Model and Syntax is adequate to represent PICS labels [PICS], however it does not provide a general-purpose mapping from PICS rating systems into an RDF representation. One such mapping is presented in section 7.

1.1.2 Simple Web Metadata

One obvious application for RDF is in the description of Web pages. This is one of the basic functions of the Dublin Core [DC] initiative. The Dublin Core is a set of 15 elements believed to be broadly applicable to describing Web resources to enable their discovery. The Dublin Core has been a major influence on the development of RDF. An important consideration in the development of the Dublin Core was to not only allow simple descriptions, but also to provide the ability to qualify descriptions in order to provide both domain specific elaboration and descriptive precision.

The RDF Schema specification provides a machine-understandable system for defining ’schemas’ for descriptive vocabularies like the Dublin Core. It allows designers to specify classes of Resource types and properties to convey descriptions of those classes, and constraints on the allowed combinations of classes, properties, and values.

An initial schema for the simple Dublin Core is provided in Appendix B. This schema defines the 15 elements as properties, and gives a description of their purpose. Despite the simplicity of the definition, it is believed that this schema serves as the foundation for more elaborate definitions. Future extensions to the Dublin Core are likely to specify the structure of the values of the properties, which will involve defining classes, the properties that apply to those classes, and some constraints on the property values. In order for browsers and authoring tools to understand and enforce these constraints, this information should be machine understandable. This document provides a machine understandable schema language for expressing such definitions and constraints.

1.1.3 Sitemaps and other such Navigation Tools

A sitemap is a hierarchical description of a Web site. A subject taxonomy is a classification system that might be used by content creators or trusted third parties to organise or classify Web resources. The RDF Schema specification provides a mechanism for defining the vocabularies needed for such applications.

Thesauri and library classification schemes are well known examples of hierarchical systems for representing subject taxonomies in terms of the relationships between named concepts. The RDF Schema specification provides sufficient resources for creating RDF models that represent the logical structure of thesauri (and other library classification systems).

1.1.4 P3P

The W3C Platform for Privacy Preferences Project (P3P) requires a grammar for constructing statements about a site’s data collection practices and personal preferences as exercised over those practices, as well as a syntax for exchanging structured data. The ability to provide third party assurances (signed statements) regarding P3P practices is also important. For instance, entities may wish to certify that P3P practice statements were properly generated in accordance with industry guidelines, have been audited, or are compliant with the relevant privacy regulations.

Although personal data collection practices could be described using an application-specific XML tagset, there are benefits to using a general metadata model for this data. Using a metadata schema to describe the formal structure of privacy practice descriptions will permit privacy practice data to be used along with other metadata in a query during resource discovery, and will permit a generic software agent to act on privacy metadata using the same techniques as used for other descriptive metadata.

1.2 Specifying the Schema

An RDF Schema can be expressed by the data model described in the RDF Model and Syntax [RDFMS] specification. The schema description language is simply a set of resources and properties defined by this specification and implicitly part of every RDF model using this schema machinery.

This document specifies the RDF Schema mechanism as a set of RDF resources (including properties), and constraints on their relationships.

1.2.1 Versioning and URI references

The Resource Description Framework is intended to be flexible and easily extensible; this suggests that a great variety of schemas will be created and that new and improved versions of these schemas will be a common occurence on the Web. Since changing the logical structure of a schema risks breaking other RDF models which depend on that schema, this specification recommends that a new URI is used whenever an RDF schema is changed.

In effect, changing the RDF statements which constitute a schema creates a new one; new schema namespaces should have their own URI to avoid ambiguity. Since an RDF Schema URI unambiguously identifies a single version of a schema, software that uses or manages RDF (eg. caches) should be able to safely store copies of RDF schema models for an indefinite period. The problems of RDF schema evolution share many characteristics with XML DTD version management and the general problem of Web resource versioning. A general approach to these issues is beyond the scope of this specification.

Since each RDF schema has its own unchanging URI, these can be used to construct unique URI references for the resources defined in a schema. This is achieved by combining the local identifier for a resource with the URI associated with that schema namespace. The XML representation of RDF uses the XML namespace mechanism for associating elements and attributes with URI references for each vocabulary item used.

Note that the formal URI corresponding to the RDF Schema namespace has not yet been assigned. For the purposes of this document we use as a temporary URI the identifier for the working draft of the RDF Schema specification. The same approach is adopted here for the URI of the RDF Model and Syntax namespace. Consequently, these URIs will change in the final version of this specification.

1.2.2 Capitalization Convention

In accord with Appendix C.1 of the RDF Model and Syntax Specification we adopt the following convention concerning capitalization of property and class names:

1. Class names have the first letter capitalized, e.g., `rdfs:Resource`. Additional words in the class name are also capitalized, e.g., `rdfs:ConstraintProperty`.
2. Property names do not have the first letter capitalized, e.g., `rdfs:domain`. However, additional words in the name are capitalized, e.g., `rdfs:subClassOf`.

2. Classes and Properties

In this section we describe the core RDF Schema. We first define the type system and then introduce the core classes and properties.

2.1 The Type System

The RDF Schema defined in this specification is a collection of RDF resources that can be used to describe properties of other RDF resources (including properties) which define application-specific RDF vocabularies. The core schema vocabulary is defined in a namespace informally called ’`rdfs`’ here, and which will be given a formal URI in the final version of this specification.

As described in the RDF Model and Syntax specification, resources may be instances of one or more classes; this is indicated with the `rdf:type` property. Classes themselves are often organised in a hierarchical fashion, for example the class ’dog’ might be considered a sub-class of ’animal’ which is a sub-class of ’organism’ etc., meaning that any resource which is of`rdf:type` ’dog’ is*ipso facto* also of `rdf:type` ’animal’ and so on. This specification describes a property,`rdfs:subClassOf`, to denote such relationships between classes.

In addition to the `rdfs:subClassOf` property, this specification introduces a number of other resources for making statements about constraints on the consistent use of properties and classes in RDF data. For example, an RDF schema might describe limitations on the types of values that are valid for some property, or on the classes to which it makes sense to ascribe such properties. This specification gives a mechanism for describing such constraints, but does not say whether or how an application must process the constraint information. For example, while an RDF schema may express that a ’Book’ may have an ’author’ property, it does not say whether or how an application should act in processing that information. We expect that different applications will use these constraints in different ways. e.g., a validator will look for errors, an interactive editor might suggest legal values.

We anticipate the development of a set of classes corresponding to a set of "datatypes." This specification does not define datatypes, but does note that datatypes may be used as the value of the `rdfs:range` property.

2.1.1 Figure 1: Classes and Resources as Sets and Elements

Figure 1 illustrates the concepts of class, sub-class, and resource. A class is depicted by a rounded rectangle, a resource is depicted by a large dot. In the figure below, arrows are drawn from a resource to the class it defines. A sub-class is shown by having a rounded rectangle (the sub-class) completely enclosed by another (the super-class). If a resource is inside a class, then there exists either an explicit or implicit `rdf:type` property of that resource whose value is the resource defining the containing class. (These properties are shown as arcs in the directed labelled graph representation in figure 2).

The RDF resources depicted in figure 1 are described either in the remainder of this specification, or in the RDF Model and Syntax specification.

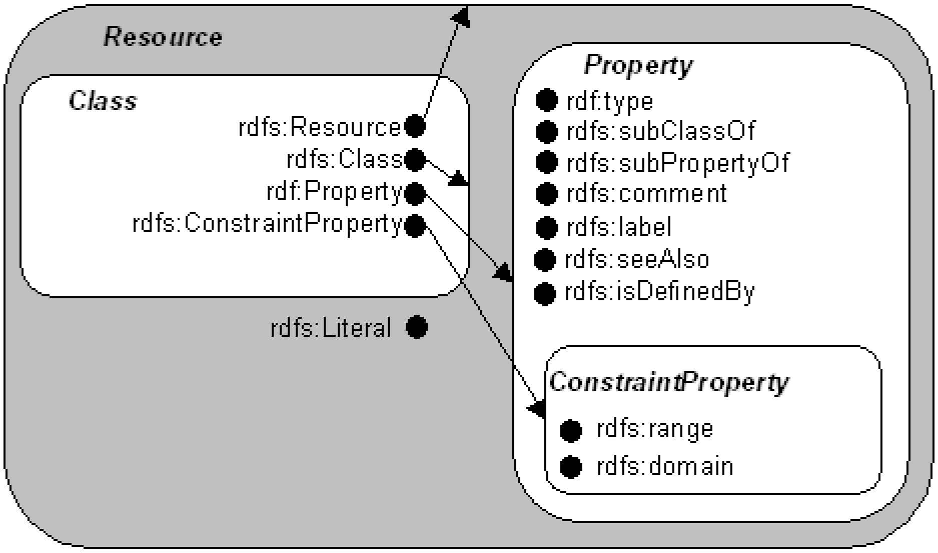


Figure 1: Classes and Resources as Sets and Elements

2.1.2 Figure 2: Class Hierarchy for the RDF Schema

Figure 2 shows the same information about the class hierarchy as in figure 1, but does so using a "nodes and arcs" graph representation of the RDF data model. If a class is a subset of another, then there is an `rdfs:subClassOf` arc from the node representing the first class to the node representing the second. Similarly, if a Resource was an instance of a Class, then there is an `rdf:type` arc from the resource to the node representing the class. (Not all such arcs are shown. We only show the arc to the most tightly encompassing class, and rely on the transitivity of the `rdfs:subClassOf` relation to provide the rest).

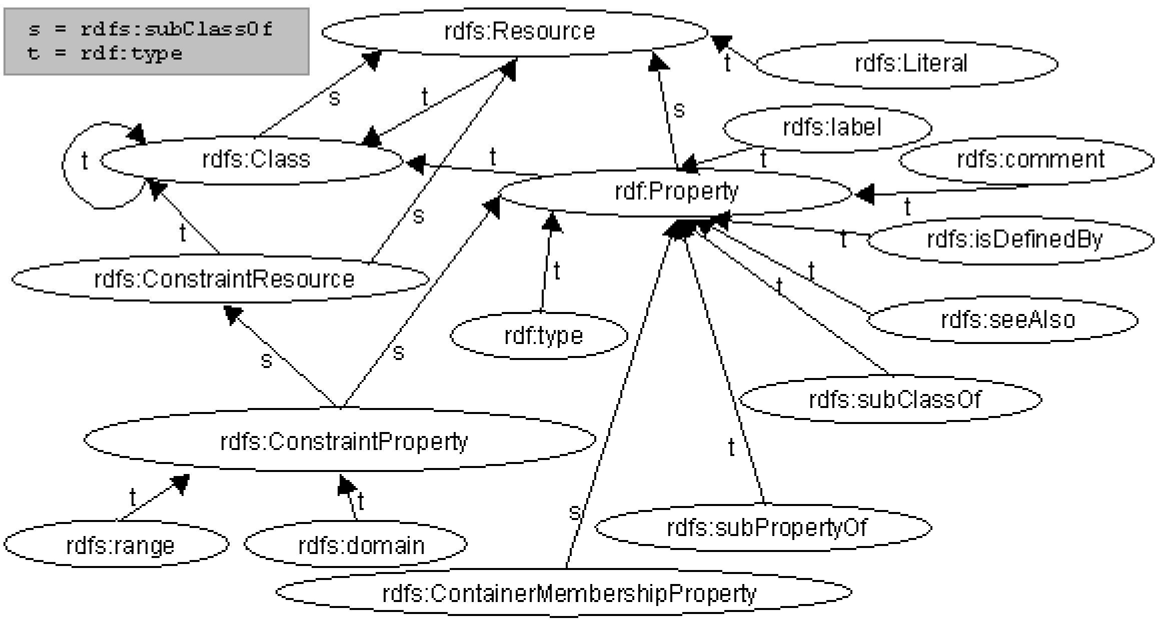


Figure 2: Class Hierarchy for the RDF Schema

2.2 Core Classes

The following resources are core classes that are defined as part of the RDF Schema machinery. Every RDF model that draws upon the RDF Schema vocabulary namespace (implicitly) includes these.

2.2.1 rdfs:Resource

All things being described by RDF expressions are called *resources*, and are considered to be instances of the class `rdfs:Resource`. The RDF class `rdfs:Resource` represents the set called ’Resources’ in the formal model for RDF presented in section 5 of the Model and Syntax specification [RDFSMS]. This roughly corresponds to the concept of *Object* in Java.

Note that the class `rdfs:Resource` is unrelated to the syntactic attribute ’resource’ used in the RDF/XML serialization syntax (just as the resource ’Description’ defined by the Dublin Core element set is unconnected to the syntactic element ’Description’ used in the RDF/XML syntax).

2.2.2 rdf:Property

The class `rdf:Property` represents the subset of RDF resources that are properties, i.e., all the elements of the set introduced as ’Properties’ in section 5 of the Model and Syntax specification [RDFSMS].

2.2.3 rdfs:Class

This corresponds to the generic concept of a *Type* or *Category*, similar to the notion of a Class in object-oriented programming languages such as Java. When a schema defines a new class, the resource representing that class must have an `rdf:type` property whose value is the resource `rdfs:Class`. RDF classes can be defined to represent almost anything, such as Web pages, people, document types, databases or abstract concepts.

2.3 Core Properties

Every RDF model which uses the schema mechanism also (implicitly) includes the following core properties. These are instances of the `rdf:Property` class and provide a mechanism for expressing relationships between classes and their instances or superclasses.

2.3.1 rdf:type

This indicates that a resource is a member of a class, and thus has all the characteristics that are to be expected of a member of that class. When a resource has an `rdf:type` property whose value is some specific class, we say that the resource is an *instance of* the specified class. The value of an `rdf:type` property for some resource is another resource which must be an instance of `rdfs:Class`. The resource known as `rdfs:Class` is itself a resource of `rdf:type` `rdfs:Class`. Individual classes (for example, ’Dog’) will always have an`rdf:type` property whose value is `rdfs:Class` (or some sub-class of `rdfs:Class`, as described in section 2.3.2). A resource may be an instance of more than one class.

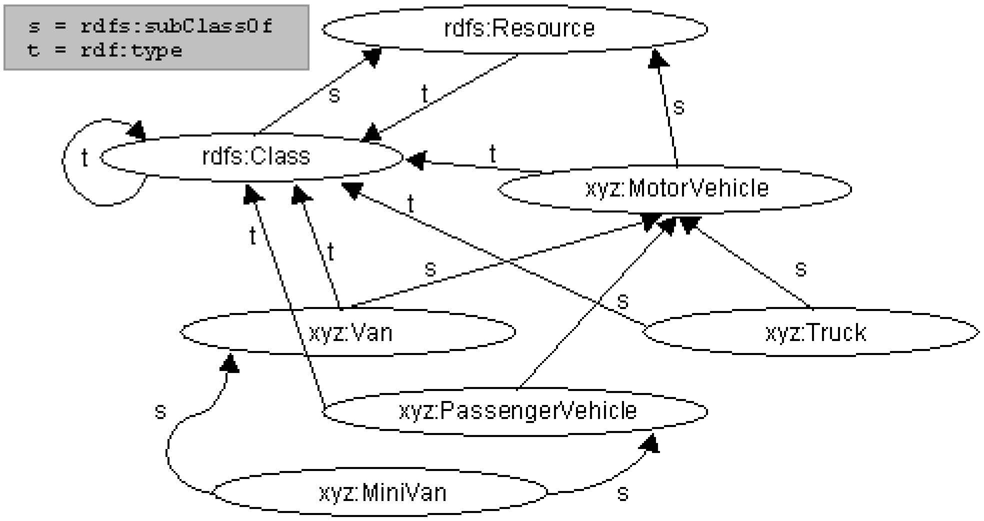
2.3.2 rdfs:subClassOf

This property specifies a subset/superset relation between classes. The `rdfs:subClassOf` property is transitive. If class A is a sub-class of some broader class B, and B is a sub-class of C, then A is also implicitly a sub-class of C. Consequently, resources that are instances of class A will also be instances of C, since A is a sub-set of both B and C. Only instances of `rdfs:Class` can have the `rdfs:subClassOf` property and the property value is always of `rdf:type` `rdfs:Class`. A class may be a sub-class of more than one class.

A class can never be declared to be a sub-class of itself, nor of any of its own sub-classes. Note that this constraint is *not* expressible using the RDF Schema constraint facilities provided below, and so does not appear in the RDF version of this specification given in Appendix A.

2.3.2.1 Example

This is a very simple example that expresses the following class hierarchy. We first define a class ’MotorVehicle’. We then define 3 subclasses of ’MotorVehicle’, namely ’PassengerVehicle’, ’Truck’ and ’Van’. We then define a class ’Minivan’ which is a sub-class of both ’Van’ and ’PassengerVehicle’.



The RDF/XML shown here uses the basic RDF syntax defined in section 2.2.1 of the Model and Syntax specification [RDFMS]. The RDF syntax also provides (in section 2.2.2) a more compact mechanism for making statements about the `rdf:type` of a resource, by allowing the identifier for a class (eg. ’Van’ in the examples below) to be used directly as an XML element name. For example, `<Van ID="v_323"/>` tells us both that there is an resource known as ’v_323’ and that it is an instance of the class `Van`. Similarly, `<rdfs:Class ID="MiniVan"/>` might be used to state that the resource known as `MiniVan` is itself a class. For clarity, these initial examples do not take advantage of the `rdf:type` abbreviation mechanism provided by the RDF serialisation syntax. In general, the XML representation of an RDF schema can make full use of all the syntactic flexibility of RDF.

```
<rdf:RDF xml:lang="en"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">

<!-- Note: this RDF schema would typically be used in RDF instance data
     by referencing it with an XML namespace declaration, for example
     xmlns:xyz="http://w3.org/examples/vehicles#". This allows us to use
     abbreviations such as xyz:MotorVehicle to refer unambiguously to the
     RDF class ’MotorVehicle’. -->

<rdf:Description ID="MotorVehicle">
 <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
 <rdfs:subClassOf
  rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource"/>
</rdf:Description>

<rdf:Description ID="PassengerVehicle">
 <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
 <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>

<rdf:Description ID="Truck">
 <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
 <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>
```

```
<rdf:Description ID="Van">
 <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
 <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>

<rdf:Description ID="MiniVan">
 <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
 <rdfs:subClassOf rdf:resource="#Van"/>
 <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
</rdf:Description>

</rdf:RDF>
```

2.3.3 rdfs:subPropertyOf

An instance of `rdf:Property` that is used to specify that one property is a specialization of another. A property may have zero, one, or more properties that it is a specialization of. If some property P2 is a subPropertyOf another more general property P1, and if a resource A has a P2 property with a value B, this implies that the resource A also has a P1 property with value B.

A property can never be declared to be a sub-property of itself, nor of any of its own sub-properties. Note that this constraint is *not* expressible using the RDF Schema constraint facilities provided below, and so does not appear in the RDF version of this specification given in Appendix A.

2.3.3.1 Example

If the property `biologicalFather` is a sub-property of the broader property `biologicalParent`, and if Fred is the `biologicalFather` of John, then it is implied that Fred is also the `biologicalParent` of John.

```
<rdf:RDF xml:lang="en"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">

<rdf:Description ID="biologicalParent">
 <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>

<rdf:Description ID="biologicalFather">
 <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
 <rdfs:subPropertyOf rdf:resource="#biologicalParent"/>
</rdf:Description>
</rdf:RDF>
```

2.3.4 seeAlso

The property `rdfs:seeAlso` specifies a resource that contains information about the subject resource. This property may be specialized using `rdfs:subPropertyOf` to more precisely indicate the nature of the information the object resource has about the subject resource. The object and the subject resources are constrained only to be instances of the class `rdfs:Resource`.

2.3.5 isDefinedBy

The property `rdfs:isDefinedBy` is a sub-property of `rdfs:seeAlso`, and indicates the resource defining the subject resource. As with `rdf:seeAlso`, this property can be applied to any instance of `rdfs:Resource` and may have as its value any `rdfs:Resource`.

The most common anticipated usage is to identify the RDF schema given a name for one of the properties or classes defined by that schema. Although XML namespace declarations will typically provide the URI where RDF vocabulary resources are defined, there are cases where additional information is required.

For example, constructs such as
`<rdfs:subPropertyOf rdf:resource="http://purl.org/dc/elements/1.0/Creator"/>`
do not indicate the URI of the schema that includes the vocabulary item `Creator` (ie., `http://purl.org/dc/elements/1.0/`).

In such cases, the `rdfs:isDefinedBy` property can be used to explicitly represent that information. This approach will also work when the URIs of the namespace and its components have no obvious relationship, as would be the case if they were identified using schemes such as GUIDs or MD-5 hashes.

2.4 RDF and Object Orientation

The RDF Schema type system is similar to the type systems of object-oriented programming languages such as Java. However, RDF differs from many such systems in that instead of defining a class in terms of the properties its instances may have, an RDF schema will define properties in terms of the classes of resource that they apply to. This is the role of the `rdfs:domain` and `rdfs:range` constraints described below. For example, we could define the "author" property to have a domain of "Book" and a range of "Literal", whereas a classical OO system might typically define a class "Book" with a property called "author" of type "Literal". One benefit of this property-centric approach is that it is very easy for anyone to say anything they want about existing resources, which is one of the axioms of the Web.

3. Constraints

An RDF schema can declare constraints associated with classes and properties. In particular, the concepts of *domain* and *range* are used in RDF schemas to make statements about the contexts in which certain properties "make sense".

Although the RDF data model does not allow for explicit properties (such as an `rdf:type` property) to be ascribed to Literals (atomic values), we nevertheless consider these entities to be members of classes (e.g. the string "John Smith" is considered to be a member of the class *rdfs:Literal*.) We expect future work in RDF and XML data-typing to provide clarifications in this area.

An RDF model that violates any of the consistency constraints specified in this document is said to be an *inconsistent* model. The following constraints are specified: `rdfs:domain` and `rdfs:range` constraints on property usage, the rule that `rdfs:subPropertyOf` and `rdfs:subClassOf` properties should not form loops, plus any further consistency constraints defined using the `rdfs:ConstraintResource` extensibility mechanism. Different applications may exhibit different behaviours in the face of an inconsistent model.

Some examples of constraints include:

- That the value of a property should be a resource of a designated class. This is expressed by the *range* property. For example, a range constraint applying to the ’author’ property may express that the value of an ’author’ property must be a resource of class ’Person’.
- That a property may only be used on resources of a certain class. For example, that an ’author’ property could only originate from a resource that was an instance of class ’Book’. This is expressed using the *domain* property.

RDF schemas can express constraints that relate vocabulary items from multiple independently developed schemas. Since URI references are used to identify classes and properties, it is possible to create new properties whose domain or range is constrained to be a class defined in another namespace.

The RDF Schema uses the constraint properties to constrain how its own properties can be used. These constraints are shown below in **figure 3**. Nodes with **bold** outlines are instances of rdfs:Class.

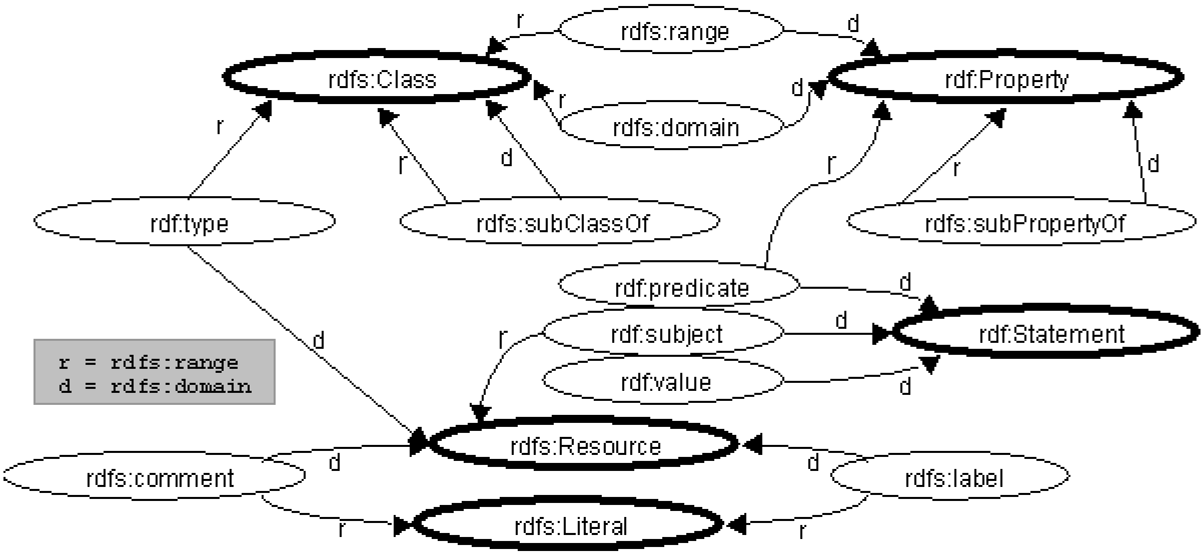


Figure 3: Constraints in the RDF Schema

3.1 Core Constraints

3.1.1 rdfs:ConstraintResource

This resource defines a sub-class of `rdfs:Resource` whose instances are RDF schema constructs involved in the expression of constraints. The purpose of this class is to provide a mechanism that allows RDF processors to assess their ability to check the consistency of an RDF model. Since this specification does not provide a mechanism for the dynamic discovery of new forms of constraint, an RDF 1.0 processor encountering previously unknown instances of `rdfs:ConstraintResource` can be sure that it is unqualified to judge the consistency of resources using properties defined using that resource.

3.1.2 rdfs:ConstraintProperty

This resource defines a sub-class of `rdf:Property`, all of whose instances are properties used to specify constraints. This class is a subclass of `rdfs:ConstraintResource` and corresponds to the subset of that class representing properties. Both `rdfs:domain` and `rdfs:range` are instances of `rdfs:ConstraintProperty`.

3.1.3 rdfs:range

An instance of `ConstraintProperty` that is used to constrain property values. The value of a `range` property is always a `Class`. The value of a property whose range is A is constrained to be an instance of class A. A property can have at most one *range* property. It is possible for it to have *no* range, in which case the class of the property value is unconstrained.

Although it is not possible to express two or more range constraints on a property, a similar outcome can be achieved by defining a common superclass for any classes that represent appropriate values for some property. For example, to express the constraint that a property `xyz:drivesMotorVehicle` can have values which are Vans, Trucks or PassengerVehicles, we assert that `xyz:drivesMotorVehicle` has a `rdfs:range` of `MotorVehicle`. If `Van`, `Truck` and `PassengerVehicle` are known to be sub-classes of `MotorVehicle`, then all these types of resource are acceptable values for `xyz:drivesMotorVehicle`. In cases where a common super-class does not exist, one can be defined in a schema in the normal manner.

As mentioned earlier, we anticipate the development of a set of classes corresponding to a set of "datatypes." Once available, these datatypes may be used as the value of the`rdfs:range` constraint property.

3.1.4 rdfs:domain

An instance of `ConstraintProperty` that is used to specify a class on which a property may be used. A property may have zero, one, or more than one class as its domain. If there is no domain property, it may be used with any resource. If there is exactly one domain property, it may only be used on instances of that class (which is the value of the domain property). If there is more than one domain property, the constrained property can be used with instances of any of the classes (that are values of those domain properties).

3.2 Example

Continuing with our earlier example of MotorVehicle, in this example, we define two properties : ’registeredTo’ and ’rearSeatLegRoom’. The ’registeredTo’ property is applicable to any MotorVehicle and its value is a Person (defined in the examples below). For the sake of this example, ’rearSeatLegRoom’ only applies to Minivans and PassengerVehicles. The value is a ’Number’ (we anticipate that some concept like this will be provided by the work on data types), which is the number of centimetres of rear seat legroom.

```
<rdf:RDF xml:lang="en"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">

<rdf:Description ID="registeredTo">
 <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
 <rdfs:domain rdf:resource="#MotorVehicle"/>
 <rdfs:range rdf:resource="#Person"/>
</rdf:Description>

<rdf:Description ID="rearSeatLegRoom">
 <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
 <rdfs:domain rdf:resource="#PassengerVehicle"/>
 <rdfs:domain rdf:resource="#Minivan"/>
 <rdfs:range rdf:resource="http://www.w3.org/datatypes#Number"/>
</rdf:Description>
</rdf:RDF>
```

4. Extensibility Mechanisms

The RDF Schema specification builds upon the foundations provided by XML and by the RDF Model and Syntax. It provides some additional facilities to support the evolution both of individual RDF vocabularies, and of the core RDF Schema specification vocabulary introduced in this document.

4.1. Evolvability of RDF vocabularies

4.1.1. Terminology

The phrase ’RDF vocabulary’ is used here to refer to those resources which evolve over time; ’RDF schema’ is used to denote those resources which constitute the particular (unchanging) versions of an RDF vocabulary at any point in time. Thus we might talk about the evolution of the Dublin Core vocabulary. Each version of the Dublin Core vocabulary would be a different RDF*schema*, and would have a corresponding RDF model and concrete syntactic representation.

4.1.2. Inter-Vocabulary Relationships

The resources defined in RDF schemas are themselves Web resources, and can be be described in other RDF schemas. This principle provides the basic mechanism for RDF vocabulary evolution. This specification does not attempt to provide a full framework for expressing mappings between schemas; it does however provide the `rdfs:subClassOf` and `rdfs:subPropertyOf` properties. The ability to express specialization relationships between classes (`subClassOf`) and between properties (`subPropertyOf`) provides a simple mechanism for making statements about how such resources map to their predecessors.

There are many scenarios for which these simple mechanisms are not adequate; a more general schema mapping mechanism for RDF may be developed in future W3C Activity.

4.1.2.1. Examples

A schema representing version 1.0 of some vocabulary might define classes corresponding to a number of vehicle types. The schema for version 2.0 of this vocabulary constitutes a different Web resource. If the new schema defines for example a class 'Van' whose members are a subset of the members of the class 'Van' in version 1.0, the`rdfs:subClassOf` property can be used to state that all instances of `V2:Van` are also instances of `V1:Van`.

Where the vocabulary defines properties, the same approach can be taken, using `rdfs:subPropertyOf` to make statements about relationships between properties defined in successive versions of an RDF vocabulary.

4.2. Evolvability of the RDF Schema Constraint Mechanism

This specification defines a sub-class of resources known as 'constraint resources' (section 3.1). This is provided to allow for the addition of new ways of expressing RDF consistency constaints. Future extensions to the Resource Description Framework may introduce new resources that are instances of the `rdfs:ConstraintResource` class. It is necessary to anticipate RDF content which draws upon properties or classes defined using constraints other than those available in this version of RDF. This specification introduces the notion of *consistent* and *inconsistent* RDF models. As yet unknown constraints may contribute to a more expressive framework for specifying whether an RDF model is or is not considered *consistent*.

RDF agents unfamiliar with the semantics of unknown instances of `rdfs:ConstraintResource` may therefore lack the knowledge to evaluate models for consistency when vocabulary items are defined using those unknown constraints. Since RDF itself may not represent declaratively the full meaning of these constraint resources, the acquisition of RDF statements about a new `ConstraintResource` may not provide enough information to enable its use in consistency checking. For example, when encountering a previously unknown constraint property type called`RDF3:mysteryConstraint` we may learn mechnically that it has a range of `rdf:Class` and a domain of `rdf:Property`. The range and domain constraints if encountered alone would be enough to tell us how to legally use `RDF3:mysteryConstraint`, but they do not tell us anything about the nature of the constraint expressed when it is used in that fashion.

The `rdfs:ConstraintResource` construct is provided here as a simple future-proofing mechanism, and addresses some of the issues discussed at greater length in the Extensible Web Languages W3C NOTE [EXTWEB]. By flagging new forms of constraint as members of this class, we indicate that they are intended to express Schema language contraints whose semantics must be understood for consistency checking to be possible. Membership of the class suggests, but does not imply, that those semantics may be inexpressible in a declarative form. Since the expressive facilties available within RDF for doing so are also likely to evolve, this distinction itself presents a moving target. All RDF agents will have implicit knowledge of certain constraints (for example, this specification declares that `subClassOf` properties must not form a loop in an RDF graph) which may or may not be capable of representation within (some version of) RDF. It may be the case that some future RDF specification provides facilties which will allow RDF agents to comprehend declarative specifications for as-yet uninvented constraint properties. In such a case, these agents could safely comprehend (some) previously unencountered forms of constraint. By providing the basic `rdfs:ConstraintResource` class, we anticipate such developments. All RDF agents written to this specification will appreciate their ignorance of the meaning of unknown instances of that class, since this specification provides no mechanism for learning about such constraints through the interpretation of RDF statements. Future specifications, should they offer such facilities, could also define sub-classes of `ConstraintProperty` to classify new constructs according to whether or not they had inexpressible semantics.

5. Documentation

The following properties are provided to support simple documentation and user-interface related annotations within RDF schemas. Multilingual documentation of schemas is supported at the syntactic level through use of the `xml:lang` language tagging facility. Since RDF schemas are expressed within the RDF data model, vocabularies defined in other namespaces may be used to provide richer documentation.

5.1 rdfs:comment

This is used to provide a human-readable natural language description of a resource.

5.2 rdfs:label

This is used to provide a human-readable version of a resource name.

6. Model and Syntax concepts

The RDF Model and Syntax specification [RDFMS] introduces certain concepts. A number of these are defined formally in another RDF Schema whose namespace URI reference is http://www.w3.org/1999/02/22-rdf-syntax-ns#. In addition, some further concepts are introduced in the RDF Model and Syntax specification but do not appear in the RDF Model and Syntax schema. These formally belong in the Schema namespace (for example, `rdfs:Literal` and `rdfs:Resource`). In cases where an RDF resource belongs to the `http://www.w3.org/1999/02/22-rdf-syntax-ns#` namespace, this document can provide only a convenience copy of that resource's definition.

Appendix A provides an RDF/XML schema for the RDF resources defined in this document, including RDF Model concepts such as `Literal` and `Resource`. The RDF/XML Schema in Appendix A also makes RDF statements about resources defined in the RDF Model and Syntax namespace. These have the status of *annotations* rather than *definitions*. This is indicated through the use of the `<rdf:Description about="...">` syntax, as against the `<rdf:Description ID="...">` construct used for the definitions of RDF Schema resources such as `domain` and `range`.

6.1. rdfs:Literal

This corresponds to the set called the 'Literals' in the formal model for RDF presented in section 5 of the Model and Syntax specification [RDFMS]. Atomic values such as textual strings are examples of RDF literals.

6.2. rdf:Statement

This corresponds to the set called the 'Statement' in the formal model for RDF presented in section 5 of the Model and Syntax specification [RDFMS].

6.3. rdf:subject

This corresponds to the property called the ’subject’ in the formal model for RDF presented in section 5 of the Model and Syntax specification [RDFMS]. Its `rdfs:domain` is `rdf:Statement` and `rdfs:range` is `rdfs:Resource`. This is used to specify the resource described by a reified statement.

6.4. rdf:predicate

This corresponds to the property called the ’predicate’ in the formal model for RDF presented in section 5 of the Model and Syntax specification [RDFMS]. Its `rdfs:domain` is `rdf:Statement` and `rdfs:range` is `rdf:Property`. This is used to identify the property used in the modelled statement.

6.5. rdf:object

This corresponds to the property called the ’object’ in the formal model for RDF presented in section 5 of the Model and Syntax specification [RDFMS]. Its rdfs:domain is `rdf:Statement`. This is used to identify the property value in the modelled statement.

6.6 rdfs:Container

This class is used to hold the Container classes described in section 3 of the Model and Syntax specification [RDFMS]. It is an instance of `rdfs:Class` and `rdfs:subClassOf` of `rdfs:Resource`.

6.7. rdf:Bag

This corresponds to the class called ’Bag’ in the formal model for RDF presented in section 5 of the Model and Syntax specification [RDFMS]. It is an instance of `rdfs:Class` and `rdfs:subClassOf` `rdfs:Container`.

6.8. rdf:Seq

This corresponds to the class called ’Sequence’ in the formal model for RDF presented in section 5 of the Model and Syntax specification [RDFMS]. It is an instance of `rdfs:Class` and `rdfs:subClassOf` `rdfs:Container`.

6.9. rdf:Alt

This corresponds to the class called ’Alternative’ in the formal model for RDF presented in section 5 of the Model and Syntax specification [RDFMS]. It is an instance of `rdfs:Class` and `rdfs:subClassOf` `rdfs:Container`.

6.10. rdfs:ContainerMembershipProperty

This class has as members the properties `_1, _2, _3 ...` used to indicate container membership, as described in section 3 of the Model and Syntax specification [RDFMS]. This is a `rdfs:subClassOf` `rdf:Property`.

6.11. rdf:value

This corresponds to the ’value’ property described in section 2.3 of the Model and Syntax specification [RDFMS].

7. Examples

This section gives some brief examples of using the RDF Schema machinery to define classes and properties for some possible applications. Note that some of these examples use the abbreviated RDF syntax (mentioned in 2.3.2.1 above) to express class membership.

Example 1

In this example, `Person` is a class with a corresponding human-readable description of "The class of people". A `Person` is a sub-class of `Animal`. A Person may have an age property. The value of age is an integer. A Person may also have an `ssn` ("Social Security Number") property. The value of ssn is an integer. A Person’s marital status is one of {Married, Divorced, Single, Widowed}. This is achieved through use of the range constraint: we define both a ’maritalStatus’ property and a class ’MaritalStatus’ (adopting the convention of using lower case letters to begin the names of properties, and capitals for classes). We then use `rdfs:range` to state that a maritalStatus property only ’makes sense’ when it has a value which is an instance of the class MaritalStatus. The schema then defines a number of instances of this class. Whether resources declared to be of type MaritalStatus in another graph are trusted is an application level decision.

```
<rdf:RDF xml:lang="en"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">

<rdfs:Class rdf:ID="Person">
  <rdfs:comment>The class of people.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://www.classtypes.org/useful_classes#Animal"/
</rdfs:Class>

<rdf:Property ID="maritalStatus">
  <rdfs:range rdf:resource="#MaritalStatus"/>
  <rdfs:domain rdf:resource="#Person"/>
</rdf:Property>

<rdf:Property ID="ssn">
  <rdfs:comment>Social Security Number</rdfs:comment>
  <rdfs:range rdf:resource="http://www.datatypes.org/useful_types#Integer"/>
  <rdfs:domain rdf:resource="#Person"/>
</rdf:Property>

<rdf:Property ID="age">
  <rdfs:range rdf:resource="http://www.datatypes.org/useful_types#Integer"/>
  <rdfs:domain rdf:resource="#Person"/>
</rdf:Property>

<rdfs:Class rdf:ID="MaritalStatus"/>

<MaritalStatus rdf:ID="Married"/>
```

```
<MaritalStatus rdf:ID="Divorced"/>
<MaritalStatus rdf:ID="Single"/>
<MaritalStatus rdf:ID="Widowed"/>

</rdf:RDF>
```

Example 2

In this example we sketch an outline of an RDF vocabulary for use with searchable Internet services. `SearchQuery` is declared to be a class. Every SearchQuery can have both a `queryString` whose value is an `rdfs:Literal` and a `queryService` whose value is a `SearchService`. A `SearchService` is a sub-class of `InternetService` (which is defined elsewhere). A `SearchQuery` has some number of `result` properties (whose value is `SearchResult`). Each `SearchResult` has a `title` (value is a literal), a `rating` (value is between 0 and 1) and of course, the page itself.

The modularity of RDF allows other vocabularies to be combined with simple schemas such as this to characterise more fully the properties of networked resources. For example, Dublin Core or a library-based classification vocabulary might be used to describe the subject coverage or collections-level properties for each SearchService, while an independently managed "search protocols" vocabulary could be used to describe connection details for (say) LDAP, WHOIS++ or Z39.50 search interfaces offered by the service. By allowing the creation of statements which draw upon specialised schemas from various domains, RDF makes it possible for communities of expertise to contribute to a decentralised web of machine-readable vocabularies.

```
<rdf:RDF xml:lang="en"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">

<rdfs:Class rdf:ID="SearchQuery">
  <rdfs:subClassOf
   rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="SearchResult">
  <rdfs:subClassOf
   rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="SearchService">
  <rdfs:subClassOf
   rdf:resource="http://www.classtypes.org/useful_classes#InternetService"/>
</rdfs:Class>

<rdf:Property ID="queryString">
  <rdf:domain rdf:resource="#SearchQuery"/>
  <rdf:range
   rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal"/>
</rdf:Property>

<rdf:Property ID="queryService">
  <rdf:domain rdf:resource="#SearchQuery"/>
  <rdfs:range rdf:resource="#SearchService"/>
</rdf:Property>
```

```
<rdf:Property ID="result">
  <rdf:domain rdf:resource="#SearchQuery"/>
  <rdfs:range rdf:resource="#SearchResult"/>
</rdf:Property>

<rdf:Property ID="queryResultPage">
  <rdfs:domain rdf:resource="#SearchResult"/>
  <rdfs:range rdf:resource="http://www.datatypes.org/useful_types#WebPage"/>
</rdf:Property>

<rdf:Property ID="queryResultTitle">
  <rdf:domain rdf:resource="#SearchResult"/>
  <rdfs:range
   rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal"/>
</rdf:Property>

<rdf:Property ID="queryResultRating">
  <rdf:domain rdf:resource="#SearchResult"/>
  <rdfs:range rdf:resource="http://www.datatypes.org/useful_types#FloatZeroToOne"/
</rdf:Property>
</rdf:RDF>
```

Example 3

A Prototype PICS/RDF Schema

Status: this is work-in-progess illustrating how a PICS schema could be expressed in RDF.

There are three parts:

1. Part A covers the vocabulary in the PICS-1.1 Label Specification
2. Part B covers the PICS-1.1 Rating Service Description vocabulary
3. Part C covers an example PICS rating system from the PICS-1.1 Rating Service Description Specification

A. PICS-1.1 Label Vocabulary

The vocabulary that appears in a PICS-1.1 label is a mixture of descriptions of the service providing the rating, data having to do with the mechanics of transporting the ratings, descriptions of the ratings themselves, and descriptions of the target resource. The vocabulary associated with the transport mechanics (`error`, `for`) are not included here. The `generic` keyword is handled by the RDF `aboutURIPrefix` mechanism. Two deprecated PICS-1.1 options (`signature-RSA-MD5` and `MIC-MD5`) are omitted from this vocabulary.

The syntax presented in the RDF/XML fragments below assumes that the RDF Model and Syntax namespace is available using the prefix `rdf:`, and that the RDF Schema namespace is the current default XML namespace.

A.1. Document Properties

The PICS 'at' option is a higher-order relation between the document being rated and the rating statement. As such, it is modelled as a property of the (reified) rating statement.

```
  <rdf:Property ID="at"
    comment="The last modification date of the
    subject resource at the time the rating was assigned">
    <domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
  </rdf:Property>
```

A.2. Label Options

The rating service URI identifies the source of the labels. In PICS-1.1 the rating system (rating vocabulary) is identified within the service description at the service URI. In PICS/RDF the rating vocabulary should be identified with its own schema (and namespace name). The ratingService property therefore serves only to identify the source of the statements. It is an open question of semantics as to what responsibility the ratingService has for the *content* of the statements.

```
  <rdf:Property ID="ratingService"
    comment="The service from whom this rating was received.">
    <domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
  </rdf:Property>
```

The remainder of the label options identify additional properties of the rating statement itself.

```
  <rdf:Property ID="by"
    comment="An identifier for the person or entity within the rating service
             who was responsible for creating this particular rating.">
    <domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
  </rdf:Property>

  <rdf:Property ID="on"
    comment="The date on which this rating was issued.">
    <domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
  </rdf:Property>
```

The PICS ’until’ label option can also be written as ’exp’. We could use`subPropertyOf` to represent this synonym; instead, we eliminate it altogether in this respresentation of PICS.

```
  <rdf:Property ID="until"
    comment="The date on which this rating expires.">
    <domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
  </rdf:Property>
```

A.3. Other Information

The collection of statements may have the following two properties.

```
  <rdf:Property ID="comment"
    comment="Information for humans who may see the label; no associated semantics.
    <domain
     rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Container"/>
  </rdf:Property>
```

The ’completeLabel’ property will appear only on a container that has not (yet) been replaced by the contents of the complete label resource. When the complete label is used, the statements in the original collection are withdrawn from the database and the statements in the complete label resource are added.

```
<rdf:Property ID="completeLabel"
  comment="A complete label that can be used in place of the current one.">
  <domain
   rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Container"/>
</rdf:Property>
```

B. PICS-1.1 Rating Service Description Vocabulary

A PICS Rating Service Description is essentially a schema. Some of the vocabulary used in PICS-1.1 translates directly to the RDF Schema vocabulary. In this RDF representation of PICS the rating system is specified separately from the rating service, rather than as a part of the rating service.

```
<rdf:Property ID="ratingService"
  comment="A document describing the rating service in human-readable terms."/>

<rdf:Property ID="icon"
  comment="A graphic associated with the rating service or with a
  specific rating property."/>
```

The following properties are used within the rating system schema to describe the schema and the properties within the schema.

```
<rdf:Property ID="name"
  comment="The name of the rating system."/>

<rdf:Property ID="description"
  comment="A human-readable description of the rating system."/>

<rdf:Property ID="minimum"
  comment="The minimum numeric value permitted."/>

<rdf:Property ID="maximum"
  comment="The maximum numeric value permitted."/>

<Class rdf:ID="Integer"
  comment="Integer numbers."/>

<Class rdf:ID="Encoding"
  comment="A value with a label"/>

<Class rdf:ID="Multivalue">
  <subClassOf resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag"/>
  <!-- used as the range of properties that can take either a singleton or a Bag
</Class>

<Class rdf:ID="Multiordered">
  <subClassOf resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq"/>
  <!-- used as the range of properties that can take either a singleton or a Seq
</Class>
```

’Categories’ and the`transmit-as` property are simply the property name. The category name is the RDF `label` property.

C. Sample PICS Rating System

The "Good Clean Fun" rating system example can be written in RDF/XML as follows:

```
<!-- This=http://www.gcf.org/ratings -->

<Description
  xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
  xmlns:p="http://www.w3.org/TR/WD-pics2.0#"
  about="">

  <p:icon rdf:resource="icons/gcf.gif"/>
  <p:name>The Good Clean Fun Rating System</p:name>
  <p:description>Everything you ever wanted to know about soap, cleaners, and
    related products.  For demonstration purposes only.</p:description>
</Description>

<Property ID="suds"
  s:label="Soapsuds Index"
  p:minimum="0.0"
  p:maximum="1.0"/>

<Property ID="density"
  s:label="suds density">
  <s:range rdf:resource="#DensityValue"/>
</Property>

<Bag ID="DensityValue">
  <li>
    <p:Encoding
      rdf:value="0"
      s:label="none">
      <p:icon rdf:resource="icons/none.gif"/>
    </p:Encoding>
  </li>
  <li>
    <p:Encoding
      rdf:value="1"
      s:label="lots">
      <p:icon rdf:resource="icons/lots.gif"/>
    </p:Encoding>
  </li>
</Bag>

<Property ID="subject"
  s:label="document subject">
  <s:range rdf:resource="#SubjectValue">
</Property>

<p:Multivalue rdf:ID="SubjectValue">
  <li>
    <p:Encoding
      rdf:value="0"
      s:label="soap"/>
  </li>
  <li>
    <p:Encoding
      rdf:value="1"
      s:label="water"/>
  </li>
  <li>
    <p:Encoding
      rdf:value="2"
      s:label="soapdish"/>
  </li>
</p:Multivalue>

<Property ID="color"
  s:label="picture color">
  <s:range resource="#ColorValue">
</Property>

<s:Class rdf:ID="ColorValue"/>

<Property ID="hue"
  s:label="hue">
  <s:domain rdf:resource="#ColorValue"/>
  <s:range rdf:resource="#HueValue"/>
</Property>

<p:Multivalue rdf:ID="HueValue">
  <li>
    <p:Encoding
      rdf:value="0"
      s:label="blue"/>
  </li>
  <li>
    <p:Encoding
      rdf:value="1"
      s:label="red"/>
  </li>
  <li>
    <p:Encoding
      rdf:value="2"
      s:label="green"/>
  </li>
</p:Multivalue>

<Property ID="intensity">
  <s:domain rdf:resource="#ColorValue"/>
  <s:range rdf:resource="http://www.w3.org/TR/WD-pics2.0#Integer"/>
  <p:minimum>0</p:minimum>
  <p:maximum>255</p:maximum>
</Property>
```

8. Acknowledgements

Note: This document was prepared and approved for publication by the W3C RDF Schema Working Group (WG). WG approval of this document does not necessarily imply that all WG members voted for its approval.

David Singer of IBM is the chair of the group; we thank David for his efforts and thank IBM for supporting him and us in this endeavor. Particular thanks are also due to Andrew Layman for his editorial work on earlier versions of this specification.

The working group membership has included:

Nick Arnett (Verity), Dan Brickley (ILRT / University of Bristol), Walter Chang (Adobe), Sailesh Chutani (Oracle), Ron Daniel (DATAFUSION), Charles Frankston (Microsoft), Joe Lapp

(webMethods Inc.), Patrick Gannon (CommerceNet), RV Guha (Netscape), Tom Hill (Apple Computer), Renato Iannella (DSTC), Sandeep Jain (Oracle), Kevin Jones, (InterMind), Emiko Kezuka (Digital Vision Laboratories), Ora Lassila (Nokia Research Center), Andrew Layman (Microsoft), John McCarthy (Lawrence Berkeley National Laboratory), Michael Mealling (Network Solutions), Norbert Mikula (DataChannel), Eric Miller (OCLC), Frank Olken (Lawrence Berkeley National Laboratory), Sri Raghavan (Digital/Compaq), Lisa Rein (webMethods Inc.), Tsuyoshi Sakata (Digital Vision Laboratories), Leon Shklar (Pencom Web Works), David Singer (IBM), Wei (William) Song (SISU), Neel Sundaresan (IBM), Ralph Swick (W3C), Naohiko Uramoto (IBM), Charles Wicksteed (Reuters Ltd.), Misha Wolf (Reuters Ltd.)

Not all of the people listed above have been members throughout the entire duration of the working group, but all have contributed to the evolution of this document.

Appendix A: XML Serialization

The RDF specification of the above is given here in the serialization syntax. Please note that the namespace URIs listed are examples only; formal identifiers have not yet been assigned for these schemas. This schema includes annotations describing RDF resources defined formally in the RDF Model and Syntax specification, as well as definitions for new resources belonging to the RDF Schema namespace.

Note that there are some constraints (such as those given in 2.3.2 above) on certain RDF Schema resources which are themselves not fully expressible using the RDF Schema specification. For example, the RDF below does not tell us that subClassOf arcs should not (to use terminology from the nodes and arcs representation) form loops in any RDF model.

Basic XML Serialization

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">

<rdf:Description ID="Resource">
 <rdf:type resource="#Class"/>
  <rdfs:label xml:lang="en">Resource</rdfs:label>
  <rdfs:label xml:lang="fr">Ressource</rdfs:label>
  <rdfs:comment>The most general class</rdfs:comment>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label xml:lang="en">type</rdfs:label>
  <rdfs:label xml:lang="fr">type</rdfs:label>
  <rdfs:comment>Indicates membership of a class</rdfs:comment>
  <rdfs:range rdf:resource="#Class"/>
</rdf:Description>

<rdf:Description ID="comment">
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label xml:lang="en">comment</rdfs:label>
  <rdfs:label xml:lang="fr">commentaire</rdfs:label>
  <rdfs:domain rdf:resource="#Resource"/>
  <rdfs:comment>Use this for descriptions</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Liter
</rdf:Description>

<rdf:Description ID="label">
 <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
 <rdfs:label xml:lang="en">label</rdfs:label>
 <rdfs:label xml:lang="fr">label</rdfs:label>
 <rdfs:domain rdf:resource="#Resource"/>
 <rdfs:comment>Provides a human-readable version of a resource name.</rdfs:comment
 <rdfs:range
  rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Literal"/>
</rdf:Description>

<rdf:Description ID="Class">
  <rdf:type resource="#Class"/>
  <rdfs:label xml:lang="en">Class</rdfs:label>
  <rdfs:label xml:lang="fr">Classe</rdfs:label>
  <rdfs:comment>The concept of Class</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Resource"/>
</rdf:Description>

<rdf:Description ID="subClassOf">
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label xml:lang="en">subClassOf</rdfs:label>
  <rdfs:label xml:lang="fr">sousClasseDe</rdfs:label>
  <rdfs:comment>Indicates membership of a class</rdfs:comment>
  <rdfs:range rdf:resource="#Class"/>
  <rdfs:domain rdf:resource="#Class"/>
</rdf:Description>

<rdf:Description ID="subPropertyOf">
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label xml:lang="en">subPropertyOf</rdfs:label>
  <rdfs:label xml:lang="fr">sousPropriétéDe</rdfs:label>
  <rdfs:comment>Indicates specialization of properties</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/
</rdf:Description>

<rdf:Description ID="seeAlso">
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label xml:lang="en">seeAlso</rdfs:label>
  <rdfs:label xml:lang="fr">voirAussi</rdfs:label>
  <rdfs:comment>Indicates a resource that provides information about the subject r
  <rdfs:range rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resou
  <rdfs:domain rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Reso
</rdf:Description>

<rdf:Description ID="isDefinedBy">
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:subPropertyOf rdf:resource="#seeAlso"/>
  <rdfs:label xml:lang="en">isDefinedBy</rdfs:label>
  <rdfs:label xml:lang="fr">esDéfiniPar</rdfs:label>
  <rdfs:comment>Indicates a resource containing and defining the subject resource.
  <rdfs:range rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resou
  <rdfs:domain rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Reso
</rdf:Description>

<rdf:Description ID="ConstraintResource">
  <rdfs:label xml:lang="en">ConstraintResource</rdfs:label>
  <rdfs:label xml:lang="fr">RessourceContrainte</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:subClassOf rdf:resource="#Resource"/>
  <rdfs:comment>Resources used to express RDF Schema constraints.</rdfs:comment>
</rdf:Description>

<rdf:Description ID="ConstraintProperty">
  <rdfs:label xml:lang="en">ConstraintProperty</rdfs:label>
  <rdfs:label xml:lang="fr">PropriétéContrainte</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Proper
  <rdfs:subClassOf rdf:resource="#ConstraintResource"/>
  <rdfs:comment>Properties used to express RDF Schema constraints.</rdfs:comment>
</rdf:Description>

<rdf:Description ID="domain">
  <rdf:type resource="#ConstraintProperty"/>
  <rdfs:label xml:lang="en">domain</rdfs:label>
  <rdfs:label xml:lang="fr">domaine</rdfs:label>
  <rdfs:comment>This is how we associate a class with
  properties that its instances can have</rdfs:comment>
</rdf:Description>

<rdf:Description ID="range">
  <rdf:type resource="#ConstraintProperty"/>
  <rdfs:label xml:lang="en">range</rdfs:label>
  <rdfs:label xml:lang="fr">étendue</rdfs:label>
  <rdfs:comment>Properties that can be used in a
  schema to provide constraints</rdfs:comment>
  <rdfs:range rdf:resource="#Class"/>
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/
</rdf:Description>

<rdf:Description ID="Property">
  <rdfs:label xml:lang="en">Property</rdfs:label>
  <rdfs:label xml:lang="fr">Propriété</rdfs:label>
  <rdfs:comment>The concept of a property.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Resource"/>
  <rdf:type resource="#Class"/>
</rdf:Description>

<rdf:Description ID="Literal">
  <rdfs:label xml:lang="en">Literal</rdfs:label>
  <rdfs:label xml:lang="fr">Littéral</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:comment>This represents the set of atomic values, eg. textual strings.</rd
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement">
  <rdfs:label xml:lang="en">Statement</rdfs:label>
  <rdfs:label xml:lang="fr">Déclaration</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:subClassOf rdf:resource="#Resource"/>
  <rdfs:comment>This represents the set of reified statements.</rdfs:comment>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#subject">
  <rdfs:label xml:lang="en">subject</rdfs:label>
  <rdfs:label xml:lang="fr">sujet</rdfs:label>
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"
  <rdfs:range rdf:resource="#Resource"/>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate">
  <rdfs:label xml:lang="en">predicate</rdfs:label>
  <rdfs:label xml:lang="fr">prédicat</rdfs:label>
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#object">
  <rdfs:label xml:lang="en">object</rdfs:label>
  <rdfs:label xml:lang="fr">objet</rdfs:label>
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"
</rdf:Description>

<rdf:Description ID="Container">
  <rdfs:label xml:lang="en">Container</rdfs:label>
  <rdfs:label xml:lang="fr">Enveloppe</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:subClassOf rdf:resource="#Resource"/>
  <rdfs:comment>This represents the set Containers.</rdfs:comment>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag">
  <rdfs:label xml:lang="en">Bag</rdfs:label>
  <rdfs:label xml:lang="fr">Ensemble</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:subClassOf rdf:resource="#Container"/>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq">
  <rdfs:label xml:lang="en">Sequence</rdfs:label>
  <rdfs:label xml:lang="fr">Séquence</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:subClassOf rdf:resource="#Container"/>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Alt">
  <rdfs:label xml:lang="en">Alt</rdfs:label>
  <rdfs:label xml:lang="fr">Choix</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:subClassOf rdf:resource="#Container"/>
</rdf:Description>

<rdf:Description ID="ContainerMembershipProperty">
  <rdfs:label xml:lang="en">ContainerMembershipProperty</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Proper
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#value">
  <rdfs:label xml:lang="en">object</rdfs:label>
  <rdfs:label xml:lang="fr">value</rdfs:label>
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>

</rdf:RDF>
```

Compact XML Serialization

The following is the same RDF data expressed using the more compact XML serialisation syntax. This representation is suitable for embedding in XML documents (such as this specification) intended for display, since the literal string values are encoded as the values of XML attributes, and as such are not displayed by HTML-oriented applications. The source code for this specification contains another copy of the following XML/RDF, making the document both human and machine readable.

```
<rdf:RDF xml:lang="en"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">

<rdfs:Class rdf:ID="Resource"
        rdfs:label="Resource"
        rdfs:comment="The most general class/">

<rdf:Property about="http://www.w3.org/1999/02/22-rdf-syntax-ns#type"
        rdfs:label="type"
        rdfs:comment="Indicates membership of a class">
  <rdfs:range rdf:resource="#Class"/>
</rdf:Property>

<rdf:Property ID="comment"
        rdfs:comment="Use this for descriptions"
        rdfs:label="comment">
  <rdfs:domain rdf:resource="#Resource"/>
  <rdfs:range rdf:resource="#Literal"/>
</rdf:Property>

<rdf:Property ID="label"
        rdfs:Label="label"
        rdfs:comment="Provides a human-readable version of a resource name">
 <rdfs:domain rdf:resource="#Resource"/>
 <rdfs:range rdf:resource="#Literal"/>
</rdf:Property>

<rdfs:Class rdf:ID="Class"
        rdfs:label="Class"
        rdfs:comment="The concept of Class">
  <rdfs:subClassOf rdf:resource="#Resource"/>
</rdfs:Class>

<rdf:Property ID="subClassOf"
        rdfs:label="subClassOf"
        rdfs:comment="Indicates membership of a class">
  <rdfs:range rdf:resource="#Class"/>
  <rdfs:domain rdf:resource="#Class"/>
</rdf:Property>

<rdf:Property ID="subPropertyOf"
        rdfs:label="subPropertyOf"
        rdfs:comment="Indicates specialization of properties">
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/
</rdf:Property>

<rdf:Property ID="seeAlso"
        rdfs:label="seeAlso"
        rdfs:comment="Indicates a resource that provides information about the sub
  <rdfs:range rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resou
  <rdfs:domain rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Reso
</rdf:Property>

<rdf:Property ID="isDefinedBy"
        rdfs:label="isDefinedBy"
        rdfs:comment="Indicates a resource containing and defining the subject res
  <rdfs:subPropertyOf rdf:resource="#seeAlso"/>
  <rdfs:range rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resou
  <rdfs:domain rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Reso
</rdf:Property>

<rdfs:Class rdf:ID="ConstraintResource"
  rdfs:label="ConstraintResource"
  rdfs:comment="Resources used to express RDF Schema constraints.">
  <rdfs:subClassOf rdf:resource="#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="ConstraintProperty"
        rdfs:label="ConstraintProperty"
        rdfs:comment="Properties used to express RDF Schema constraints.">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Proper
  <rdfs:subClassOf rdf:resource="#ConstraintResource"/>
</rdfs:Class>

<rdfs:ConstraintProperty rdf:ID="domain"
        rdfs:label="domain"
        rdfs:comment="This is how we associate a class with  properties that its i
  <rdfs:range rdf:resource="#Class"/>
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/
</rdfs:ConstraintProperty>

<rdfs:ConstraintProperty rdf:ID="range"
  rdfs:label="range"
  rdfs:comment="Properties that can be used in a
  schema to provide constraints">
  <rdfs:range rdf:resource="#Class"/>
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/
</rdfs:ConstraintProperty>

<rdfs:Class rdf:ID="Property"
        rdfs:label="Property"
        rdfs:comment="The concept of a property.">
  <rdfs:subClassOf rdf:resource="#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Literal"
  rdfs:label="Literal"
  rdfs:comment="This represents the set of atomic values, eg. textual strings.">
  <rdfs:subClassOf rdf:resource="#Resource"/>
</rdfs:Class>

<rdfs:Class about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"
        rdfs:labe="Statement"
        rdfs:comment="This represents the set of reified statements.">
  <rdfs:subClassOf rdf:resource="#Resource"/>
</rdfs:Class>

<rdf:Property about="http://www.w3.org/1999/02/22-rdf-syntax-ns#subject"
        rdfs:label="subject">
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"
  <rdfs:range rdf:resource="#Resource"/>
</rdf:Property>

<rdf:Property about="http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate"
        rdfs:label="predicate">
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Property>
```

```
<rdf:Property about="http://www.w3.org/1999/02/22-rdf-syntax-ns#object"
                rdfs:label="object">
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"
</rdf:Property>

<rdfs:Class rdf:ID="Container"
  rdfs:label="Container"
  rdfs:comment="This represents the set Containers.">
  <rdfs:subClassOf rdf:resource="#Resource"/>
</rdfs:Class>

<rdfs:Class about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag"
        rdfs:label="Bag">
  <rdfs:subClassOf rdf:resource="#Container"/>
</rdfs:Class>

<rdfs:Class about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Seq"
        rdfs:label="Sequence">
  <rdfs:subClassOf rdf:resource="#Container"/>
</rdfs:Class>

<rdfs:Class about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Alt"
        rdfs:label="Alt">
  <rdfs:subClassOf rdf:resource="#Container"/>
</rdfs:Class>

<rdfs:Class rdf:ID="ContainerMembershipProperty"
        comment="This is the class that the properties _1,_2, ... that are used to
  <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Proper
</rdfs:Class>

<rdf:Property about="http://www.w3.org/1999/02/22-rdf-syntax-ns#value"
  rdfs:label="value"/>
</rdf:RDF>
```

Appendix B: Dublin Core

The following represents an initial schema for the simple Dublin Core Element Set [DC]. The following schema is for illustration purposes only; the authoritative Dublin Core Schema will be made available by the Dublin Core Initiative.

This schema is provided as the foundation for the Dublin Core semantics. It is believed to be all that is needed to serve as the foundation for future, more elaborate definitions. Future extensions are likely to specify the structure of the values of the properties, which will involve defining classes, the properties that apply to those classes, and some constraints on the property values.

```
  <?xml version='1.0'?>
  <rdf:RDF
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#"
     xmlns:dc="">

  <rdf:Description about = "">
    <dc:Title> The Dublin Core Element Set </dc:Title>
    <dc:Creator> The Dublin Core Metadata Inititative </dc:Creator>
    <dc:Description> The Dublin Core is a simple metadata element
        set intended to facilitate discovery of electronic
        resources. </dc:Description>
    <dc:Date> 1995-03-01 </dc:Date>
  </rdf:Description>

  <rdf:Description ID="Title">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:label>Title</rdfs:label>
    <rdfs:comment>The name given to the resource, usually by the Creator
    or Publisher.</rdfs:comment>
    <rdfs:isDefinedBy rdf:resource = ""/>
  </rdf:Description>

  <rdf:Description ID="Creator">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:label>Author/Creator</rdfs:label>
    <rdfs:comment>The person or organization primarily responsible for
    creating the intellectual content of the resource. For example,
    authors in the case of written documents, artists, photographers, or
    illustrators in the case of visual resources.</rdfs:comment>
    <rdfs:isDefinedBy rdf:resource = ""/>
  </rdf:Description>

  <rdf:Description ID="Subject">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:label>Subject</rdfs:label>
    <rdfs:comment>The topic of the resource. Typically, subject will be
    expressed as keywords or phrases that describe the subject or
    content of the resource. The use of controlled vocabularies and
    formal classification schemes is encouraged.</rdfs:comment>
    <rdfs:isDefinedBy rdf:resource = ""/>
  </rdf:Description>

  <rdf:Description ID="Description">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:label>Description</rdfs:label>
    <rdfs:comment> A textual description of the content of the resource,
    including abstracts in the case of document-like objects or content
    descriptions in the case of visual resources.</rdfs:comment>
    <rdfs:isDefinedBy rdf:resource = ""/>
  </rdf:Description>

  <rdf:Description ID="Publisher">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:label>Publisher</rdfs:label>
    <rdfs:comment>The entity responsible for making the resource
    available in its present form, such as a publishing house, a
    university department, or a corporate entity.</rdfs:comment>
    <rdfs:isDefinedBy rdf:resource = ""/>
  </rdf:Description>

  <rdf:Description ID="Contributor">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:label>Other Contributors</rdfs:label>
    <rdfs:comment>A person or organization not specified in a Creator
    element who has made significant intellectual contributions to the
    resource but whose contribution is secondary to any person or
    organization specified in a Creator element (for example, editor,
    transcriber, and illustrator).</rdfs:comment>
    <rdfs:isDefinedBy rdf:resource = ""/>
  </rdf:Description>
```

```
<rdf:Description ID="Date">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label>Date</rdfs:label>
  <rdfs:comment>A date associated with the creation or availability of
  the resource. Such a date is not to be confused with one belonging
  in the Coverage element, which would be associated with the resource
  only insofar as the intellectual content is somehow about that
  date. Recommended best practice is defined in a profile of ISO 8601
  [Date and Time Formats (based on ISO8601), W3C Technical Note,
  http://www.w3.org/TR/NOTE-datetime] that includes (among others)
  dates of the forms YYYY and YYYY-MM-DD. In this scheme, for example,
  the date 1994-11-05 corresponds to November 5, 1994.</rdfs:comment>
  <rdfs:isDefinedBy rdf:resource = ""/>
</rdf:Description>

<rdf:Description ID="Type">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label>Type</rdfs:label>
  <rdfs:comment>The category of the resource, such as home page,
  novel, poem, working paper, technical report, essay, dictionary. For
  the sake of interoperability, Type should be selected from an
  enumerated list that is currently under development in the workshop
  series.</rdfs:comment>
  <rdfs:isDefinedBy rdf:resource = ""/>
</rdf:Description>

<rdf:Description ID="Format">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label>Format</rdfs:label>
  <rdfs:comment>The data format of the resource, used to identify the
  software and possibly hardware that might be needed to display or
  operate the resource. For the sake of interoperability, Format
  should be selected from an enumerated list that is currently under
  development in the workshop series.</rdfs:comment>
  <rdfs:isDefinedBy rdf:resource = ""/>
</rdf:Description>

<rdf:Description ID="Identifier">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label>Identifier</rdfs:label>
  <rdfs:comment>A string or number used to uniquely identify the
  resource. Examples for networked resources include URLs and URNs
  (when implemented). Other globally-unique identifiers, such as
  International Standard Book Numbers (ISBN) or other formal names are
  also candidates for this element.</rdfs:comment>
  <rdfs:isDefinedBy rdf:resource = ""/>
</rdf:Description>

<rdf:Description ID="Source">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label>Source</rdfs:label>
  <rdfs:comment>Information about a second resource from which the
  present resource is derived. While it is generally recommended that
  elements contain information about the present resource only, this
  element may contain a date, creator, format, identifier, or other
  metadata for the second resource when it is considered important for
  discovery of the present resource; recommended best practice is to
  use the Relation element instead.  For example, it is possible to
  use a Source date of 1603 in a description of a 1996 film adaptation
  of a Shakespearean play, but it is preferred instead to use Relation
  "IsBasedOn" with a reference to a separate resource whose
  description contains a Date of 1603. Source is not applicable if the
  present resource is in its original form.</rdfs:comment>
  <rdfs:isDefinedBy rdf:resource = ""/>
```

```
</rdf:Description>

<rdf:Description ID="Language">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label>Language</rdfs:label>
  <rdfs:comment>The language of the intellectual content of the
  resource. Where practical, the content of this field should coincide
  with RFC 1766 [Tags for the Identification of Languages,
  http://ds.internic.net/rfc/rfc1766.txt ]; examples include en, de,
  es, fi, fr, ja, th, and zh.</rdfs:comment>
  <rdfs:isDefinedBy rdf:resource = ""/>
</rdf:Description>

<rdf:Description ID="Relation">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label>Relation</rdfs:label>
  <rdfs:comment>An identifier of a second resource and its
  relationship to the present resource. This element permits links
  between related resources and resource descriptions to be
  indicated. Examples include an edition of a work (IsVersionOf), a
  translation of a work (IsBasedOn), a chapter of a book (IsPartOf),
  and a mechanical transformation of a dataset into an image
  (IsFormatOf). For the sake of interoperability, relationships should
  be selected from an enumerated list that is currently under
  development in the workshop series.</rdfs:comment>
  <rdfs:isDefinedBy rdf:resource = ""/>
</rdf:Description>

<rdf:Description ID="Coverage">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label>Coverage</rdfs:label>
  <rdfs:comment>The spatial or temporal characteristics of the
  intellectual content of the resource. Spatial coverage refers to a
  physical region (e.g., celestial sector); use coordinates (e.g.,
  longitude and latitude) or place names that are from a controlled
  list or are fully spelled out. Temporal coverage refers to what the
  resource is about rather than when it was created or made available
  (the latter belonging in the Date element); use the same date/time
  format (often a range) [Date and Time Formats (based on ISO8601),
  W3C Technical Note, http://www.w3.org/TR/NOTE-datetime] as
  recommended for the Date element or time periods that are from a
  controlled list or are fully spelled out.</rdfs:comment>
  <rdfs:isDefinedBy rdf:resource = ""/>
</rdf:Description>

<rdf:Description ID="Rights">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label>Rights</rdfs:label>
  <rdfs:comment>A rights management statement, an identifier that
  links to a rights management statement, or an identifier that links
  to a service providing information about rights management for the
  resource.</rdfs:comment>
  <rdfs:isDefinedBy rdf:resource = ""/>
</rdf:Description>
</rdf:RDF>
```

Appendix C: References

[RDFMS]

Resource Description Framework (RDF) Model and Syntax; http://www.w3.org/TR/REC-rdf-syntax/

[NIAM]
G. M. Nijssen and Terry Halpin, Conceptual Schema and Relational Database Design, (Prentice Hall, Sydney:1989)

[CycL]
CycL: The CYC Representation Language; http://www.cyc.com/tech.html#cycl

[KIF]
Knowledge Interchange Format (KIF); http://logic.stanford.edu/kif/kif.html

[WF]
The Warwick Framework: A Container Architecture for Aggregating Sets of Metadata; Carl Lagoze, Clifford A. Lynch and Ron Daniel Jr.; http://cs-tr.cs.cornell.edu/Dienst/UI/2.0/Describe/ncstrl.cornell/TR96-1593

[EXTWEB]
Web Architecture: Extensible Languages ; Tim Berners-Lee and Dan Connolly; http://www.w3.org/TR/NOTE-webarch-extlang

[PICS]
Platform for Internet Content Selection; http://www.w3.org/PICS/

[DC]
The Dublin Core initiative; http://purl.oclc.org/metadata/dublin_core

[UML]
Unified Modelling Language (UML); http://www.rational.com/uml/resources

[XML-Data]
XML Data; http://www.w3.org/Submission/1998/01/

[XML-Link]
XML Linking Language (XLink) http://www.w3.org/TR/WD-xml-link

---

Comments
$Date: 1999/03/04 17:04:59 $