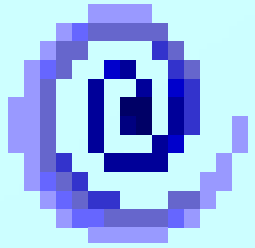
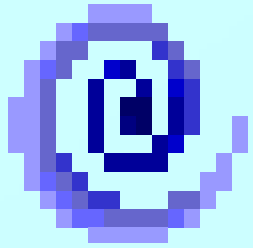


RDF Schema



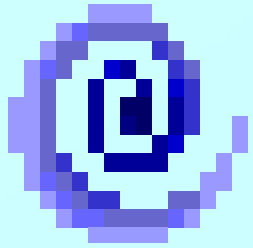
What is the RDF Schema Specification?

- Specifies how to use RDF to describe RDF vocabularies.
- Defines a basic vocabulary.
- Defines an extensibility mechanism in anticipation of additions to RDF.



Current Status

- **Resource Description Framework (RDF) Schema Specification 1.0**
 - **W3C Candidate Recommendation 27 March 2000**
 - **<http://www.w3.org/TR/rdf-schema>**



RDF Review

The Formal Model

There is a set called Resources.

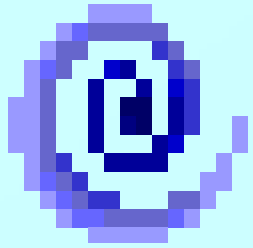
There is a set called Literals.

There is a subset of Resources called Properties.

There is a set called Statements, each element of which is a triple of the form

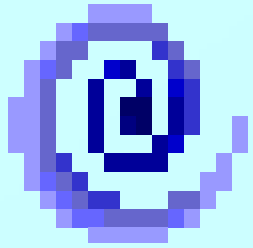
{pred, sub, obj}

Where pred is a property (member of Properties), sub is a resource (member of Resources), and obj is either a resource or a literal (member of Literals).



So we can describe things like:

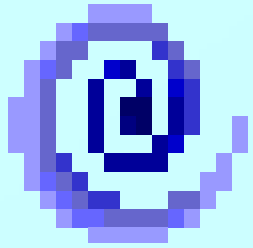
```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://description.org/schema/">
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>
      <rdf:Description about="http://www.w3.org/staffId/85740">
        <rdf:type resource="http://description.org/schema/Person"/>
        <v:Name>Ora Lassila</v:Name>
        <v:Email>lassila@w3.org</v:Email>
      </rdf:Description>
    </s:Creator>
  </rdf:Description>
</rdf:RDF>
```



What RDF Does

- This is nice, because it gives a mechanism for associating semantics with documents in a generally readable manner.
- But...

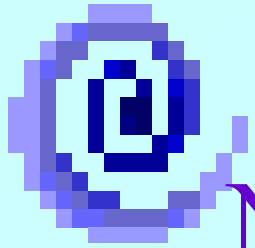
Semantic information isn't that useful without being structured so that it can be consistently interpreted.



If we don't use schemata, many representations are possible

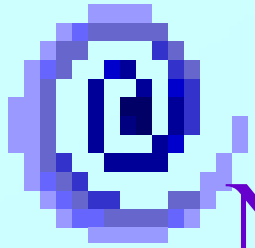
```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
    syntax-ns#" />
  <rdf:Description
    about="http://www.w3.org/Home/Lassila">
    <Creator>
      <rdf:Description
        about="http://www.w3.org/staffId/85740">
        <rdf:type
          resource="http://desc.org/schema/Person"/>
        <Name>Ora Lassila</Name>
        <Email>lassila@w3.org<Email>
      </rdf:Description>
    </Creator>
  </rdf:Description>
</rdf:RDF>
```

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
    syntax-ns#" />
  <rdf:Description
    about="http://www.w3.org/Home/Lassila">
    <author>
      <rdf:Description
        about="http://www.w3.org/staffId/85740">
        <rdf:type
          resource="http://desc.org/schema/Person"/>
        <name>
          <surname>Lassila</surname>
          <given>Ora</given>
        </name>
        <email>lassila@w3.org</email>
      </rdf:Description>
    </author>
  </rdf:Description>
</rdf:RDF>
```



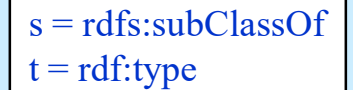
Not *another* schema scheme?

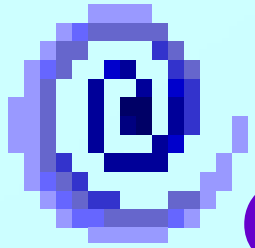
- Why another schema definition?
Can't we just use DTDs or XML Schema?



Not *another* schema scheme?

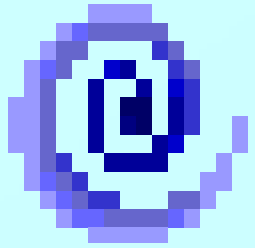
- Why another schema definition?
Can't we just use DTDs or XML Schema?
DTD and XML Schema definitions –
 - only define syntax.
 - don't have the power to describe things like class membership in a robust manner.
 - Don't refer to things outside of XML.





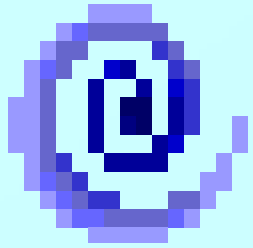
Core Classes - `rdfs:Resource`

- Things 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.



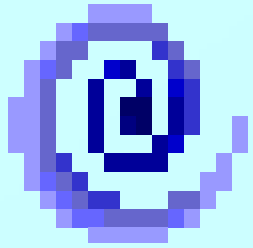
Core Classes - `rdf:Property`

- This represents the subset of RDF resources that are properties, i.e., all the elements of the set called 'Properties' the formal model.



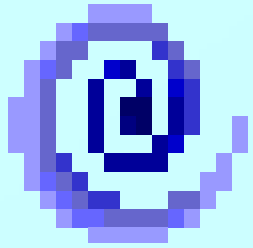
Core Classes - `rdfs:Class`

- 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.



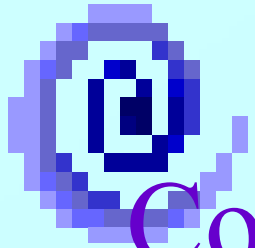
The Core Properties

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



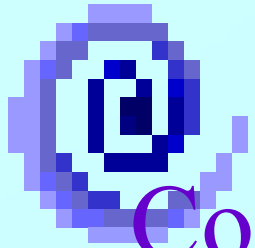
Core Properties - `rdf:type`

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. 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 subclass of `rdfs:Class`, as described in section 2.3.2). A resource may be an instance of more than one class.



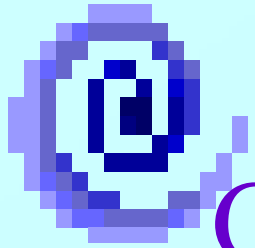
Core Properties - `rdfs:subClassOf`

- This specifies a subset/superset relation between classes. The `rdfs:subClassOf` property is transitive. If class A is a subclass of some broader class B, and B is a subclass of C, then A is also implicitly a subclass of 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 subclass of more than one class.
- A class cannot be a subclass of itself, nor of any of its own subclasses.



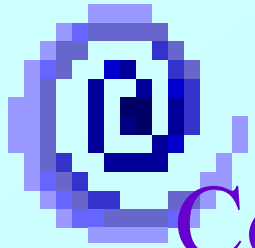
Core Properties - `rdfs:subPropertyOf`

- This instance of `rdf:Property` specifies that one property is a specialization of another. A property may be a specialization of zero, one or more properties. 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 subproperty of itself, nor of any of its own subproperties.



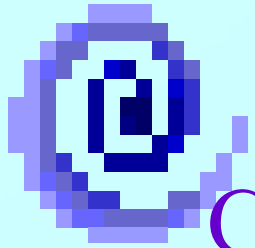
Core Properties - `rdfs:seeAlso`

- This specifies a resource that might provide additional 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`.



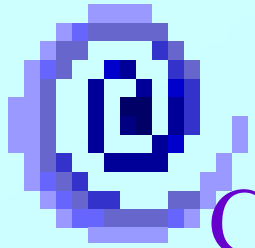
Core Properties - `rdfs:isDefinedBy`

- This is a subproperty 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`.



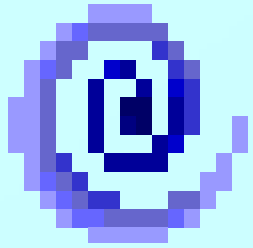
Constraints -rdfs:ConstraintResource

- This resource defines a subclass of rdfs:Resource whose instances are RDF schema constructs involved in the expression of constraints. This provides a mechanism that allows RDF processors to assess their ability to use the constraint information associated with an RDF model.
- The 1.0 specification doesn't provide a mechanism for the dynamic discovery of new forms of constraint.



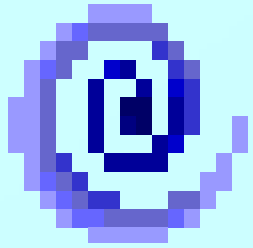
Constraints - `rdfs:ConstraintProperty`

- This resource defines a subclass 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`.



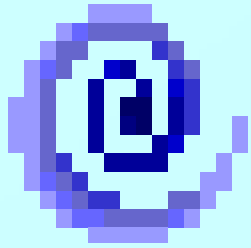
Constraints - rdfs:range

- An instance of `ConstraintProperty` that is used to indicate the class(es) that the values of a property must be members of. The value of a range property is always a `Class`. Range constraints are only applied to properties.
- 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.



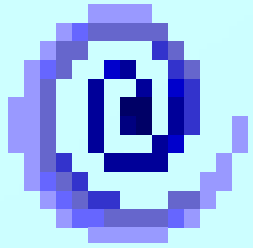
Constraints - rdfs:domain

- This is an instance of `ConstraintProperty` that is used to indicate the class(es) on whose members a property can be used.
- If a property has no domain property, it may be used with any resource. If it has exactly one domain property, it may only be used on instances of that class (which is the value of the domain property). If it has more than one domain property, the constrained property can be used with instances of any of those classes.



Constraints on ranges and domains

- The `rdfs:domain` of `rdfs:range` is the class `rdf:Property`. This indicates that the range property applies to resources that are themselves properties.
- The `rdfs:range` of `rdfs:range` is the class `rdfs:Class`. This indicates that any resource that is the value of a range property will be a class.
- The `rdfs:domain` of `rdfs:domain` is the class `rdf:Property`. This indicates that the domain property is used on resources that are properties.
- The `rdfs:range` of `rdfs:domain` is the class `rdfs:Class`. This indicates that any resource that is the value of a domain property will be a class.



Documentation Tags

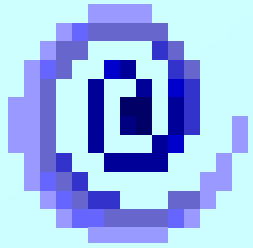
These support simple documentation and user-interface related annotations. Multilingual documentation of schemas is supported at the syntactic level through use of the `xml:lang` tagging facility.

`rdfs:comment`

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

`rdfs:label`

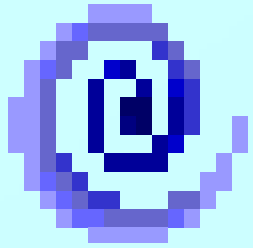
- This is used to provide a human-readable version of a resource name.



Container Membership

`rdfs:ContainerMembershipProperty`

- This class has as members the properties `_1`, `_2`, `_3` ... used to indicate container membership, as described in the formal model.



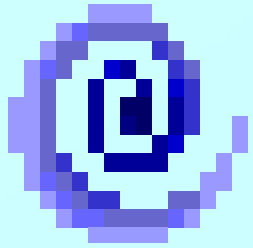
Other Things

`rdfs:Literal` - This corresponds to the set called the 'Literals' in the formal model.

`rdf:Statement` - This corresponds to the set called the 'Statement' in the formal model.

`rdf:subject` - This corresponds to the property called the 'subject' in the formal model. Its `rdfs:domain` is `rdf:Statement` and `rdfs:range` is `rdfs:Resource`.

`rdf:predicate` - This corresponds to the property called the 'predicate' in the formal model. Its `rdfs:domain` is `rdf:Statement` and `rdfs:range` is `rdf:Property`. This is used to identify the property used in the modeled statement.



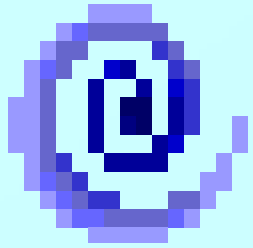
Other Things

`rdfs:Container` - This class is used to represent the Container classes described in the model. It is an instance of `rdfs:Class` and `rdfs:subClassOf` of `rdfs:Resource`.

`rdf:Bag` - This corresponds to the class called 'Bag' in the formal model. It is an instance of `rdfs:Class` and `rdfs:subClassOf` `rdfs:Container`.

`rdf:Seq` - This corresponds to the class called 'Sequence' in the formal model. It is an instance of `rdfs:Class` and `rdfs:subClassOf` `rdfs:Container`.

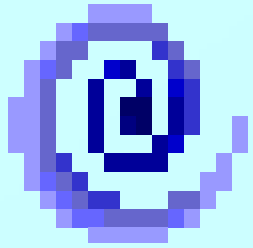
`rdf:Alt` - This corresponds to the class called 'Alternative' in the formal model. It is an instance of `rdfs:Class` and `rdfs:subClassOf` `rdfs:Container`.



Other Things

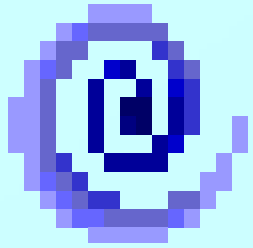
`rdf:object` - This corresponds to the property called the 'object' in the formal model. Its `rdfs:domain` is `rdf:Statement`. This is used to identify the property value in the modeled statement

`rdf:value` - This corresponds to the 'value' property described in the specification.



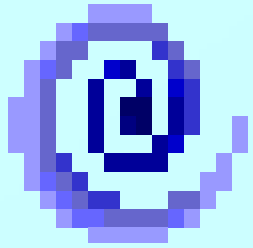
Example

```
<rdf:RDF xml:lang="en"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdfs:Class rdf:ID="Person">
    <rdfs:comment>The class of people.</rdfs:comment>
    <rdfs:subClassOf
      rdf:resource="http://www.w3.org/2000/03/example/classes#Animal"/>
  </rdfs:Class>
  <rdf:Property ID="maritalStatus">
    <rdfs:range rdf:resource="#MaritalStatus"/>
    <rdfs:domain rdf:resource="#Person"/>
  </rdf:Property>
```



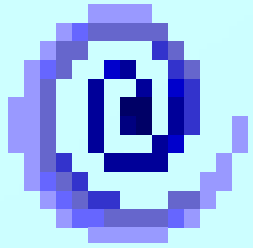
Example (continued)

```
<rdf:Property ID="ssn">  
  <rdfs:comment>Social Security Number</rdfs:comment>  
  <rdfs:range  
    rdf:resource="http://www.w3.org/2000/03/example/classes#Integer"/>  
  <rdfs:domain rdf:resource="#Person"/>  
</rdf:Property>  
<rdf:Property ID="age">  
  <rdfs:range  
    rdf:resource="http://www.w3.org/2000/03/example/classes#Integer"/>  
  <rdfs:domain rdf:resource="#Person"/>  
</rdf:Property>
```



Example (continued)

```
<rdfs:Class rdf:ID="MaritalStatus"/>  
<MaritalStatus rdf:ID="Married"/>  
<MaritalStatus rdf:ID="Divorced"/>  
<MaritalStatus rdf:ID="Single"/>  
<MaritalStatus rdf:ID="Widowed"/>  
</rdf:RDF>
```

References

- Resource Description Framework (RDF) Schema Specification 1.0: <http://www.w3.org/TR/rdf-schema>
- Resource Description Framework (RDF) Model and Syntax Specification:
<http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>
- Expressing Simple Dublin Core in RDF/XML:
<http://www.dublincore.org/documents/dcmes-xml/>
(note that this does not use an RDF Schema).