

# RDF Schema

RDFS can be directly used to describe an ontology. Objects, Classes, and Properties, can be described. Predefined properties can be used to model instance of and subclass of relationships as well as domain restrictions and range restrictions of attributes. A speciality of RDFS is that properties are defined globally and are not encapsulated as attributes in class definitions. Therefore, a frame or object-oriented ontology can only be expressed in RDFS by reifying the property names with class name suffixes

## Introduction

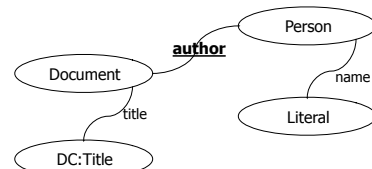
- The Resource Description Framework (RDF) defines a simple model for describing relationships among resources in terms of named properties and values. RDF properties may be thought of as attributes of resources and in this sense correspond to traditional attribute-value pairs. RDF properties also represent relationships between resources. As such, the RDF data model can therefore resemble an entity-relationship diagram.
- The RDF data model itself, however, provides no mechanisms for describing these properties, nor does it provide any mechanisms for describing the relationships between these properties and other resources.
- That is the role of this specification. The RDF vocabulary description language (RDF Schema) defines classes and properties that can be used to describe other classes and properties.

## RDF Schema (RDFS)

- RDF just defines the data model.
- Need for definition of vocabularies for the data model - an Ontology Language!
- RDF-Schemas describe rules for using RDF properties
  - Define a domain vocabulary for RDF
  - Organise this vocabulary in a typed hierarchy
- RDF Schemas are Web resources (and have URIs) and can be described using RDF
- Are not to be confused with XML Schemas
- RDFS is a framework for a vocabulary

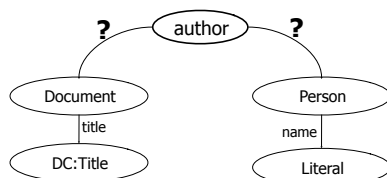
## Introductory example

- Person *writes/authors* Document.
- Person has *name* (Literal).
- Document has *title* (DC:title).



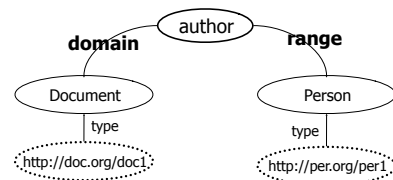
## Introductory example 2

- Person *writes/authors* Document.
- Person has *name* (Literal).
- Document has *title* (DC:title).
- If we want to focus on the PROPERTY, ...



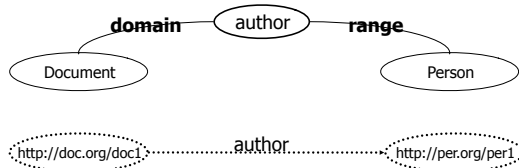
## Domain and Range

- Person *writes/authors* Document.
- Person has *name* (Literal).
- Document has *title* (DC:title).
- If we want to focus on the PROPERTY, ...



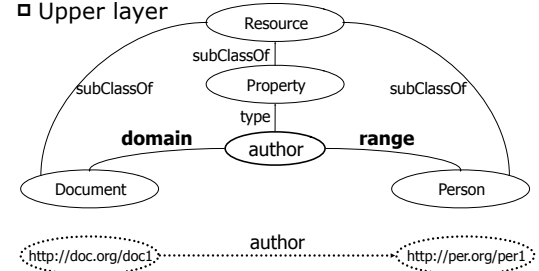
## Two-layer structure

- Person *writes/authors* Document.
- Person has *name* (Literal).
- Document has *title* (DC:title).
- If we want to focus on the PROPERTY, ...



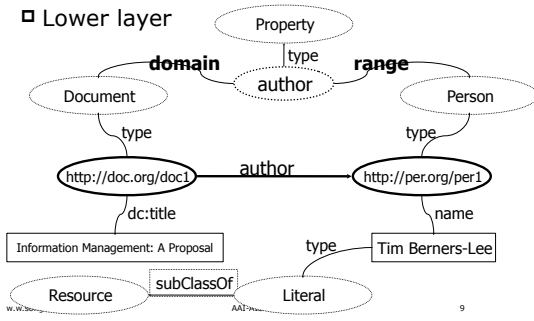
## Two-layer structure

- Upper layer



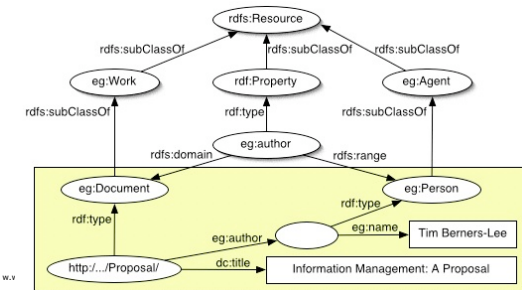
## Two-layer structure

- Lower layer



## Introductory example 3

A simple example is presented here in diagrammatic form, illustrating the use of the RDF Schema vocabulary for describing classes and properties, and the connection to application-level data.



## RDF Schema Model (General)

- Property-centric: Each property specifies what classes of subjects and objects it relates. New properties can be added to a class without modifying the class
  - resource, class, subclassOf, type
  - property, subPropertyOf
  - domain, range, constraintResource, constraintProperty
- Definitions can include constraints which express validation conditions
  - domain constraints link properties with classes
  - range constraints limit property values
- BUT... expressiveness inadequacy and poorly defined semantics

## Property-centric: Domain and Range

- The RDF Schema class and property 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 to which they apply. This is the role of the `rdfs:domain` and `rdfs:range` mechanisms.

## Property-centric: Domain and Range 2

- For example, we could define the `eg:author` property to have a domain of `eg:Document` and a range of `eg:Person`, whereas a classical OO system might typically define a class `eg:Book` with an attribute called `eg:author` of type `eg:Person`. Using the RDF approach, it is easy for others to subsequently define additional properties with a domain of `Document` or a range of `eg:Person`. This can be done without the need to re-define the original description of these classes.
- One benefit of the RDF **property-centric** approach is that it is very easy for anyone to say anything they want about existing resources, which is one of the architectural principles of the Web.

## RDF Core Classes and Properties 1

### **rdfs:Resource**

All things described by RDF are called *resources*, and are members of the class `rdfs:Resource`.

### **rdfs:Literal**

The class `rdfs:Literal` represents the self-denoting nodes called the 'literals' in the RDF graph structure. Property values such as textual strings are examples of RDF literals.

### **rdfs:Class**

This corresponds to the generic concept of a *type* or *category* of resource.

\* RDF class membership is used to represent types or categories of resources. Two classes may happen to have the same members, while remaining distinct resources.

## Example

```
<rdfs:Class rdf:ID="Animal">
  <rdfs:label>Animal</rdfs:label>
  <rdfs:comment>
    This class of animals is illustrative
    of a number of ontological idioms.
  </rdfs:comment>
</rdfs:Class>
```

## RDF Core Classes and Properties 2

### **rdf:Property**

`rdf:Property` represents those resources that are RDF properties.

### **rdf:type**

The `rdf:type` property indicates that a resource is a member of a 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 will always be a resource that is an instance of `rdfs:Class`. The resource known as `rdfs:Class` is itself a resource of `rdf:type rdfs:Class`.

## Example

```
<rdf:Property rdf:ID="hasParent">
  <rdfs:domain rdf:resource="#Animal"/>
  <rdfs:range rdf:resource="#Animal"/>
</rdf:Property>

<rdf:DatatypeProperty rdf:ID="shoesize">
  <rdfs:comment>shoesize is a DatatypeProperty whose
  range is xsd:decimal. shoesize is also a UniqueProperty
  (can only have one shoesize)
</rdfs:comment>
  <rdf:type
    rdf:resource="http://www.w3.org/2001/10/daml+oil#UniqueProperty"/>
  <rdfs:range
    rdf:resource="http://www.w3.org/2000/10/XMLSchema#decimal"/>
</rdf:DatatypeProperty>
```

## RDF Core Classes and Properties 3

### **rdfs:subClassOf**

The `rdfs:subClassOf` property represents a specialisation relationship between classes of resources. The `rdfs:subClassOf` property is transitive.

### **rdfs:subPropertyOf**

The property `rdfs:subPropertyOf` is an instance of `rdf:Property` that is used to specify that one property is a specialisation of another.

Sub-property hierarchies can be used to express hierarchies of range and domain constraints.

\* The term 'super-property' is sometimes used to indicate the relationship between some property and another more general property that it is a `rdfs:subPropertyOf`.

## Example

```
<rdfs:Class rdf:ID="Male">
  <rdfs:subClassOf rdf:resource="#Animal"/>
</rdfs:Class>

<rdfs:Property rdf:ID="hasFather">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  <rdfs:range rdf:resource="#Male"/>
</rdfs:Property>
```

## RDF Core Classes and Properties 4

### **rdfs:range** and **rdfs:domain**

An instance of `rdfs:Property`, used to indicate class(es) that the values of a property will be members of.

The value of an `rdfs:range` or `rdfs:domain` property is always a `rdfs:Class`. The `rdfs:range` or `rdfs:domain` property can itself be used to express this, e.g: 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:range` and `rdfs:domain` property is only applied to properties. The `rdfs:domain` of `rdfs:range` is the class `rdfs:Property`. This indicates that the range property applies to resources that are themselves properties.

## RDF Core Classes and Properties 4'

### **Note: range, domain, and sub-property hierarchies**

Sub-property hierarchies can be used to express hierarchies of range and domain constraints. All `rdfs:range` and `rdfs:domain` properties that apply to an RDF property also apply to each of its sub-properties.

## RDF Core Classes and Properties 5

### **rdfs:label**

The `rdfs:label` property is used to provide a human-readable version of a resource's name.

### **rdfs:comment**

The `rdfs:comment` property is used to provide a human-readable description of a resource.

A textual comment helps clarify the meaning of RDF classes and properties. Such inline documentation complements the use of both formal techniques (Ontology and rule languages) and informal (prose documentation, examples, test cases). A variety of documentation forms can be combined to indicate the intended meaning of the classes and properties described in an RDF Schema. [Multilingual text is supported with `xml:lang`.](#)

## RDF Container Classes and Properties 1

### **rdfs:Container**

The `rdfs:Container` class is a super-class of the RDF Container classes, i.e., `rdf:Bag`, `rdf:Seq`, `rdf:Alt`.

### **rdf:Bag**

The `rdf:Bag` class represents RDF's 'Bag' container construct, and is a subclass of `rdfs:Container`.

### **rdf:Seq**

The `rdf:Seq` class represents RDF's 'Sequence' container construct, and is a subclass of `rdfs:Container`.

### **rdf:Alt**

The `rdf:Alt` class represents RDF's 'Alt' container construct, and is a subclass of `rdfs:Container`.

## RDF Container Classes and Properties 2

### **rdfs:ContainerMembershipProperty**

The `rdfs:ContainerMembershipProperty` class has as members the property `rdfs:member` and the properties `_1`, `_2`, `_3`, ... that can be used to indicate membership of `Bag`, `Seq`, and `Alt` containers. `rdfs:ContainerMembershipProperty` is a subclass of `rdf:Property`. Each container membership property is a `rdfs:subPropertyOf` the `rdfs:member` property.

### **rdfs:member**

The `rdfs:member` property is a super-property of the container membership properties.

(That is, each numbered container membership property has a `rdfs:subPropertyOf` relationship to the property `rdfs:member`).

## RDF Container Classes and Properties 3

### **rdf:List**

The `rdf:List` class represents the class of RDF Lists. It is used with the 'first', 'rest' and 'nil' constructs, and has special case support in the RDF/XML syntax.

### **rdf:first**

The `rdf:first` property represents a relationship between an `rdf:List` and its first item.

### **rdf:rest**

The `rdf:rest` property represents a relationship between an `rdf:List` item and the rest of the list, or its end (i.e., `rdf:nil`).

**rdf:nil** The `rdf:nil` resource represents an empty `rdf:List`.

## RDF Utility Classes and Properties 1

### **rdfs:seeAlso**

The property `rdfs:seeAlso` is used to indicate a resource that might provide additional RDF information about the subject resource.

### **rdfs:isDefinedBy**

The property `rdfs:isDefinedBy` is a subproperty of `rdfs:seeAlso`, and indicates the resource defining the subject resource.

### **rdf:value**

The `rdf:value` property identifies the principal value (usually a string) of a property when the property value is a structured resource.

## RDF Utility Classes and Properties 2

**rdf:Statement** The `rdf:Statement` class represents statements about the properties of resources.

`rdf:Statement` is the domain of the properties `rdf:predicate`, `rdf:subject` and `rdf:object`.

Different individual `rdf:Statement` instances may happen to have the same values for their predicate, subject, and object properties.

**rdf:subject** The subject of an RDF statement.

The `rdf:subject` property indicates a resource that is the subject of some RDF statement.

The `rdfs:domain` of `rdf:subject` is `rdf:Statement` and the `rdfs:range` is `rdfs:Resource`. This property can be used to specify the resource described by an RDF statement.

## RDF Utility Classes and Properties 3

### **rdf:predicate**

The predicate of an RDF statement.

The `rdfs:domain` of `rdf:predicate` is `rdf:Statement` and the `rdfs:range` is `rdfs:Resource`. This property can be used to specify the predicate used in an RDF statement.

### **rdf:object**

The object of an RDF statement.

The `rdfs:domain` of `rdf:object` is `rdf:Statement`. No range is defined for this property since values of `rdfs:object` can include both Literals and Resources. This property can be used to specify the object of an RDF statement.