

Semantic Web

RDF and RDFS

F. Abel, N. Henze, and D. Krause

IVS Semantic Web Group

13.11.2008

1 RDF: Resource Description Framework

- Purpose
- Basic Building Blocks
 - Resources
 - Properties
 - Statements
- RDF: XML-Based Syntax
- rdf:Type
- Reification
- Data Types
- RDF: Collections

2 RDF Schema (RDFS)

- RDFS: Purpose
 - Classes and properties
 - Class hierarchies
 - Property Hierarchies
- The RDF Schema Language
 - RDFS: Core Classes and Properties
 - RDFS: Restricting Properties
 - Container

Slides in this section are based on the Semantic Web Primer, Grigoris Antoniou and Frank van Harmelen and RDF Primer, 10 February 2004, <http://www.w3.org/TR/rdf-primer/>.

- *Model for representing information about resources in the WWW, in particular: metadata about Web resources (e.g., title, author, version, publication date, ...)*
- *Model for representing information about things that can be identified in the WWW, in particular: metadata about items (e.g., prices, specifications, availability information, etc.)*

Goal: *Allow applications to access and process this specified information*

- **Identifying** things using Web Identifiers (Uniform Resource Identifications, **URIs**)
- **Describing** things using **properties** and **property values**

Example

Consider the following XML document

```
<academicStaffMember> Grigoris Antoniou </academicStaffMember>
<professor> Michael Maher </professor>
<course name="Discrete Mathematics">
  <isTaughtBy> David Billington </isTaughtBy>
</course>
```

How to collect all academic staff members?

- 1 The XML way (XPath):

```
//academicStaffMember
→ Grigoris Antoniou
```

- 2 Semantically: answer must be Grigoris Antoniou, Michael Maher and David Billington.

All professors and lecturers are academic staff members (professor and lecturer are a subclass of academicStaffMember)

For this kind of information required: a **semantic model** of a particular domain

Example

Someone named John Smith created a particular Web page.

`http://www.example.org/index.html` *has a creator whose value is* **John Smith**.

RDF uses such kind of statements:

- The thing / resource (*`http://www.example.org/index.html`*), *the statement is about, is called* **subject**
- *The part that identifies the property (creator) or characteristic of the subject is called* **predicate**
- *And the value of that property (*John Smith*) is called* **object**

RDF

subject - predicate - object statements

What's needed for machines?

- A system of machine-processable identifiers for identifying a subject, predicate, or object in a statement without any possibility of confusion with a similar-looking identifier that might be used by someone else on the Web.
- A machine-processable language for representing these statements and exchanging them between machines.

Resources

Identifiers: URIs. URIs are not limited to identify network locations, they identify:

- Network-accessible things, such as an electronic document, an image, a service (e.g., "today's weather report for Los Angeles"), or a group of other resources.
- Things that are not network-accessible, such as human beings, corporations, and bound books in a library.
- Abstract concepts that do not physically exist, such as the concept of a "creator".

Properties

- Properties: special kind of resources
- Describe relations between resources, e.g., "written-by", "creator-of", ...
- Properties: are identified by URIs

Statements

`http://www.example.org/index.html` has a **creator whose value is **John Smith**.**

can be represented by an RDF *statement*:

- A subject `http://www.example.org/index.html`
- A predicate `http://purl.org/dc/elements/1.1/creator`
- And an object `http://www.example.org/staffid/85740`

Example

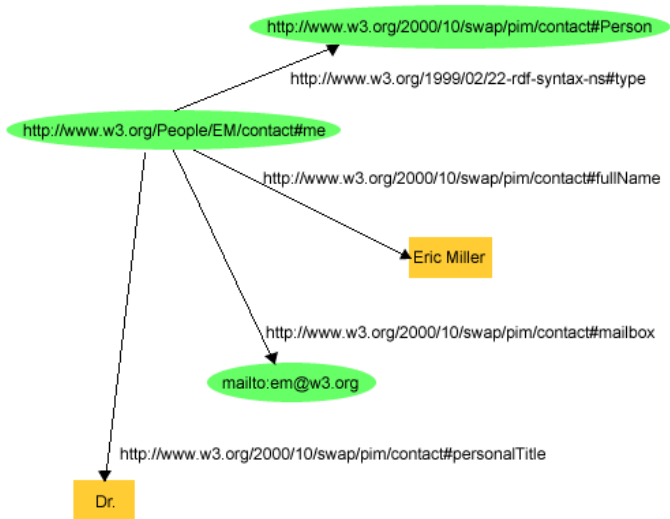
- " There is a Person identified by <http://www.w3.org/People/EM/contact#me>,
- whose name is Eric Miller,
- whose email address is em@w3.org,
- and whose title is Dr."

.. can be viewed as a graph

RDF models statements as nodes and arcs in a graph:

- A node for the subject
- A node for the object
- An arc for the predicate, directed from the subject node to the object node.

RDF: Three views of a statement - RDF Graph



... and can be written as triples: (N3 notation)

```
(http://www.w3.org/People/EM/contact#me,  
http://www.w3.org/2000/10/swap/pim/contact#mailbox,  
mailto:em@w3.org)
```

```
(http://www.w3.org/People/EM/contact#me,  
http://www.w3.org/2000/10/swap/pim/contact#personalTitle,  
"Dr.")
```

```
(http://www.w3.org/People/EM/contact#me,  
http://www.w3.org/2000/10/swap/pim/contact#fullName,  
"Eric Miller")
```

```
(http://www.w3.org/People/EM/contact#me,  
http://www.w3.org/1999/02/22-rdf-syntax-ns#type,  
http://www.w3.org/2000/10/swap/pim/contact#Person)
```

- We can think of a triple (x,P,y) as a logical formula $P(x,y)$, where the binary predicate P relates the object x to the object y .
- RDF allows only for binary predicates!

... and can be serialized to XML:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">

  <rdf:Description rdf:about="http://www.w3.org/People/EM/contact#me">
    <rdf:type rdf:resource="http://www.w3.org/2000/10/swap/pim/contact#Person">
      <contact:fullName>Eric Miller</contact:fullName>
      <contact:mailbox rdf:resource="mailto:em@w3.org"/>
      <contact:personalTitle>Dr.</contact:personalTitle>
    </rdf:Description>
  </rdf:RDF>
```

- XML element tag `rdf:RDF`
- The content of this element is a number of *descriptions*, tag `rdf:Description`
- Every description makes a statement *about* a resource: three different ways:
 - an `about` attribute: referencing to an existing resource
 - an `id` attribute: creating a new resource
 - without a name: creating an anonymous resource
- Namespaces in RDF: refer to RDF documents defining resources, which are then used in the importing RDF document - this allows the reuse of resources!
- The content of `rdf:Description` elements are called *property elements*. They define the property-value pairs, e.g. `contact:fullName`, "Eric Miller"
- Property elements must be read conjunctively:
`http://www.w3.org/People/EM/contact#me` has `contact:fullName` "Eric Miller" **and** has `contact:personalTitle` Dr.

Example

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:uni="http://www.mydomain.org/#">

  <rdf:Description
    rdf:about="http://www.mydomain.org/uni-ns/#949318">
    <uni:name>David Billington</uni:name>
    <uni:title>Associate Professor</uni:title>
  </rdf:Description>

  <rdf:Description
    rdf:about="http://www.mydomain.org/uni-ns/#CIT1111">
    <uni:courseName>Discrete Mathematics</uni:courseName>
    <uni:isTaughtBy>David Billington</uni:isTaughtBy>
  </rdf:Description>
```

- Descriptions can be nested

Example

```
<rdf:Description rdf:about="CIT1111">
  <uni:courseName>
    Discrete Mathematics
  </uni:courseName>
  <uni:isTaughtBy>
    <rdf:Description rdf:about="949318">
      <uni:name>David Billington</uni:name>
      <uni:title>
        Associate Professor
      </uni:title>
    </rdf:Description>
  </uni:isTaughtBy>
</rdf:Description>
```

- In the examples: we have descriptions about two categories: courses, and lecturers
- In RDF: such things are expressed using the `rdf:type` element

Example

```
<rdf:Description rdf:ID="#949318">
  <rdf:type
    rdf:resource="http://www.mydomain.org/uni-ns/#lecturer"/>
  <uni:name>David Billington</uni:name>
  <uni:title>Associate Professor</uni:title>
</rdf:Description>

<rdf:Description
  rdf:about="http://www.mydomain.org/uni-ns/#CIT1111">
  <rdf:type
    rdf:resource="http://www.mydomain.org/uni-ns/#course"/>
  <uni:courseName>Discrete Mathematics</uni:courseName>
  <uni:isTaughtBy rdf:resource="#949318">
</rdf:Description>
```

Reference RDF Statements

The following description

```
<rdf:Description rdf:about="949318">  
  <uni:name>David Billington</uni:name>  
</rdf:Description>
```

reifies to

```
<rdf:Statement rdf:about="StatementAbout949318">  
  <rdf:subject rdf:resource="949318">  
  <rdf:predicate rdf:resource="&uni:name">  
  <rdf:object rdf:resource="David Billington">  
</rdf:Statement>
```

To allow data types in RDF `rdf:datatype` was introduced.

Example

```
<rdf:Description rdf:about="949318">
  <uni:name
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    David Billington
  </uni:name>
</rdf:Description>
```

- Consequence: RDF allows the use of any externally defined data typing scheme
- (Practically, most widely used data typing schemes are the one defined by XML Schema)

RDF allows for binary properties only

Example (The chess example)

X is the referee in the chess game between players Y and Z.

This would bleed into a Quadruple: referee(X,Y,Z)

How to model in RDF?

```
ref(chessGame,X)
```

```
player1(chessGame,Y)
```

```
player2(chessGame,z)
```

- `rdf:Bag`
 - unordered
 - may contain multiple occurrences
- `rdf:Seq`
 - ordered
 - may contain multiple occurrences
- `rdf:Alt`
 - a set of alternatives
- Content of container elements are named `rdf:_1`, `rdf:_2`, ..
- Container elements can have an optional ID attribute

Example (Examples for RDF Container)

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:uni="http://www.mydomain.org/#">
  <uni:lecturer rdf:about="949352" uni:name="Grigoris Antoniou"
    uni:title="Professor">
    <uni:coursesTaught>
      <rdf:Bag rdf:ID="GACourses">
        <rdf:_1:rdf:resource="CIT1112"/>
        <rdf:_2:rdf:resource="CIT1113"/>
      </rdf:Bag>
    </uni:coursesTaught>
  </uni:lecturer>

  <uni:course rdf:about="CIT1111" uni:courseName="Discrete Mathematics">
    <uni:lecturer>
      <rdf:Alt>
        <rdf:_1:rdf:resource="949352"/>
        <rdf:_2:rdf:resource="949318"/>
      </rdf:Alt>
    </uni:lecturer>
  </uni:course>
</rdf:RDF>
```

Typical application of containers: the representation of predicates with more than two arguments.

Example (The chess example revisited)

```
<referee rdf:about="...#X">
  <players>
    <rdf:Bag>
      <rdf:_1:rdf:resource=".../#Y"/>
      <rdf:_2:rdf:resource=".../#Z"/>
    </rdf:Bag>
  </players>
</referee>
```

Note:

- There is no possibility "to close" the container, to say: "these are *all* elements"
- RDF is a graph, so: there is no way to exclude the possibility that there is another graph somewhere that describes additional members

- Collections: describe groups containing only the specified members
- Realized via a predefined collection vocabulary consisting of the predefined types:
`rdf:List`, `rdf:first`, `rdf:rest`, `rdf:nil`

Example

CIT 2112 is exclusively taught by teachers 949111,949352, 949381:

```
<rdf:Description rdf:about="CIT2112">
  <uni:isTaughtBy>
    <rdf:List>
      <rdf:first><rdf:Description rdf:about="949111"/></rdf:first>
      <rdf:rest>
        <rdf:List>
          <rdf:first><rdf:Description rdf:about="949352"/></rdf:first>
          <rdf:rest>
            <rdf:List>
              <rdf:first><rdf:Description rdf:about="949318"/></rdf:first>
              <rdf:rest><rdf:Description rdf:about="&rdf:nil"/></rdf:rest>
            </rdf:List>
          </rdf:rest>
        </rdf:List>
      </rdf:rest>
    </rdf:List>
  </uni:isTaughtBy>
</rdf:Description>
```

Abbreviation: Using the `rdf:parseType` attribute:

```
<rdf:Description rdf:about="CIT2112">
  <uni:isTaughtBy rdf:parseType="Collection">
    <rdf:Description rdf:about="949111"/>
    <rdf:Description rdf:about="949352"/>
    <rdf:Description rdf:about="949318"/>
  </uni:isTaughtBy>
</rdf:Description>
```

Abbreviated Syntax

Three Rules

- 1** The namespace can be abbreviated with "&namespace_name;"
- 2** Childless property elements within description elements may be replaced by XML attributes (as in XML)
- 3** For description elements with a typing element we can use the name specified in the `rdf:type` element instead of `rdf:Description`

Example (Abbreviations)

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:uni="http://www.mydomain.org/uni-ns/#">

  <rdf:Description rdf:ID="CIT1111">
    <rdf:type rdf:resource="http://www.mydomain.org/uni-ns/#course"/>
    <uni:courseName>Discrete Mathematics</uni:courseName>
    <uni:isTaughtBy rdf:resource="#949318">
  </rdf:Description>
```

is equivalent to (according to rule 1)

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:uni="http://www.mydomain.org/uni-ns/#">

  <rdf:Description rdf:ID="CIT1111">
    <rdf:type rdf:resource="&uni;course"/>
    <uni:courseName>Discrete Mathematics</uni:courseName>
    <uni:isTaughtBy rdf:resource="#949318">
  </rdf:Description>
```

Example (Abbreviations)

is equivalent to (according to rule 1)

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:uni="http://www.mydomain.org/uni-ns/#">

  <rdf:Description rdf:ID="CIT1111">
    <rdf:type rdf:resource="&uni;course"/>
    <uni:courseName>Discrete Mathematics</uni:courseName>
    <uni:isTaughtBy rdf:resource="#949318">
  </rdf:Description>
```

is equivalent to (according to rule 2)

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:uni="http://www.mydomain.org/uni-ns/#">

  <rdf:Description rdf:about="CIT1111"
                 uni:courseName="Discrete Mathematics">
    <rdf:type rdf:resource="&uni;course"/>
    <uni:isTaughtBy rdf:resource="#949318">
  </rdf:Description>
```

Example (Abbreviations)

is equivalent to (according to rule 2)

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:uni="http://www.mydomain.org/uni-ns/#">

  <rdf:Description rdf:about="CIT1111"
                 uni:courseName="Discrete Mathematics">
    <rdf:type rdf:resource="&uni;course"/>
    <uni:isTaughtBy rdf:resource="#949318">
  </rdf:Description>
```

and is equivalent to (according to rule 3)

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:uni="http://www.mydomain.org/uni-ns/#">

  <uni:course rdf:about="CIT1111"
             uni:courseName="Discrete Mathematics">
    <uni:isTaughtBy rdf:resource="#949318">
  </uni:course>
```

Example (Lecturers at a university, RDF)

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:uni="http://www.mydomain.org/uni-ns/">

  <rdf:Description rdf:about="949352">
    <uni:name>Grigoris Antoniou</uni:name>
    <uni:title>Professor</uni:title>
  </rdf:Description>

  <rdf:Description rdf:about="949318">
    <uni:name>David Billington</uni:name>
    <uni:title>Associate Professor</uni:title>
    <uni:age rdf:datatype="&xsd;integer">27</uni:age>
  </rdf:Description>

  <rdf:Description rdf:about="949111">
    <uni:name>Michael Maher</uni:name>
    <uni:title>Professor</uni:title>
  </rdf:Description>
```

Example (Lecturers at a university, RDF)

```
<rdf:Description rdf:about="CIT1111">  
  <uni:courseName>Discrete Mathematics</uni:courseName>  
  <uni:isTaughtBy>David Billington</uni:isTaughtBy>  
</rdf:Description>
```

```
<rdf:Description rdf:about="CIT1112">  
  <uni:courseName>Concrete Mathematics</uni:courseName>  
  <uni:isTaughtBy>Grigoris Antoniou</uni:isTaughtBy>  
</rdf:Description>
```

```
<rdf:Description rdf:about="CIT2112">  
  <uni:courseName>Programming III</uni:courseName>  
  <uni:isTaughtBy>Michael Maher</uni:isTaughtBy>  
</rdf:Description>
```

Example (Lecturers at a university, RDF)

```
<rdf:Description rdf:about="CIT3112">
  <uni:courseName>Theory of Computation</uni:courseName>
  <uni:isTaughtBy>David Billington</uni:isTaughtBy>
</rdf:Description>

<rdf:Description rdf:about="CIT3116">
  <uni:courseName>Knowledge Representation</uni:courseName>
  <uni:isTaughtBy>Grigoris Antoniou</uni:isTaughtBy>
</rdf:Description>

<rdf:Description rdf:about="CIT3116">
  <uni:courseName>Knowledge Representation</uni:courseName>
  <uni:isTaughtBy>Grigoris Antoniou</uni:isTaughtBy>
</rdf:Description>

</rdf:RDF>
```

Example (Another RDF document representing the lecturers)

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:uni="http://www.mydomain.org/uni-ns#">

  <uni:lecturer rdf:about="949352" uni:name="Grigoris Antoniou"
                uni:title="Professor">
    <uni:coursesTaught>
      <rdf:Bag>
        <rdf:_1 rdf:resource="CIT1112"/>
        <rdf:_2 rdf:resource="CIT3116"/>
      </rdf:Bag>
    </uni:coursesTaught>
  </uni:lecturer>

  <uni:lecturer rdf:about="949318" uni:name="David Billington"
                uni:title="Associate Professor">
    <uni:coursesTaught>
      <rdf:Bag>
        <rdf:_1 rdf:resource="CIT1111"/>
        <rdf:_2 rdf:resource="CIT3112"/>
      </rdf:Bag>
    </uni:coursesTaught>
  </uni:lecturer>
```

Example (Another RDF document representing the lecturers:)

```
<uni:lecturer rdf:about="949111"  
    uni:name="Michael Maher"  
    uni:title="Professor">  
  <uni:coursesTaught rdf:resource="CIT2112"/>  
</uni:lecturer>  
  
<uni:course rdf:about="CIT1111"  
    uni:courseName="Discrete Mathematics">  
  <uni:isTaughtBy rdf:resource="949318"/>  
</uni:course>  
  
<uni:course rdf:about="CIT1112"  
    uni:courseName="Concrete Mathematics">  
  <uni:isTaughtBy rdf:resource="949352"/>  
</uni:course>
```

Example (Another RDF document representing the lecturers:)

```
<uni:course rdf:about="CIT2112"
             uni:courseName="Programming III">
  <uni:isTaughtBy rdf:resource="949111"/>
</uni:course>

<uni:course rdf:about="CIT3112"
             uni:courseName="Theory of Computation">
  <uni:isTaughtBy rdf:resource="949318"/>
</uni:course>

<uni:course rdf:about="CIT3116"
             uni:courseName="Knowledge Representation">
  <uni:isTaughtBy rdf:resource="949352"/>
</uni:course>
</rdf:RDF>
```

- RDF: universal languages that lets users describe resources
- RDF: independent from any application domain
- RDF: no means to define the semantics of any domain

→ express semantics of a particular application domain with RDF Schema

Classes and properties

- Specify the "things" that we want to talk about: the **classes** which define particular types of objects
- Restrict the values of the properties: restrict the **range of a property**

Discrete Mathematics is taught by David Billington
Requires to restrict the range of "is taught by" to be lecturer.

- Subclass-of relationship:
 - defines a hierarchy of classes
 - A is a subclass of B if every instance of A is also an instance of B
 - a class may have multiple superclasses

Example: "Courses must be taught by academic staff members only"

- Suppose X.Y. is a professor
- A professor is a staff member
- The instance X.Y. inherits the ability to teach from the superclass staff member
- This is called: "RDF *fixes the semantics* of 'is a subclass of' "
- **all** RDF processing software must reflect this meaning of 'is subclass of'

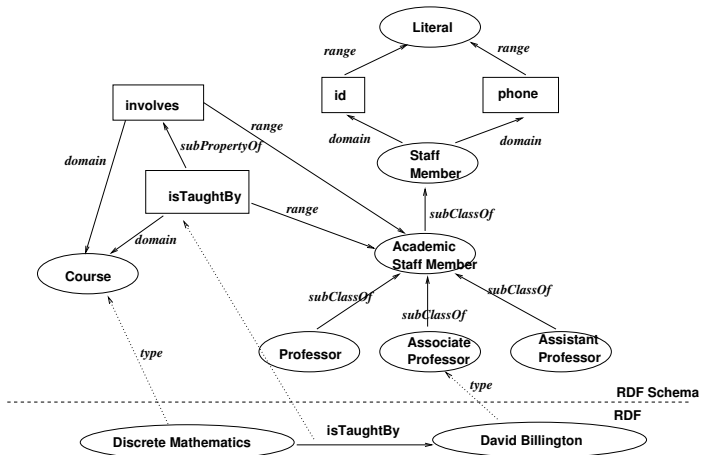
Comparison to Object-Oriented Programming

- In OO: An object class defines the properties that apply to it. To add a new property means to modify the class
- In RDF: properties are defined globally - they are not encapsulated as attributes in the class definition
Thus it is possible to define new properties without changing the class!

Property Hierarchies

- Hierarchical relationship for properties: *subproperty-of*
- "is taught by" is a subproperty of "involves": If a course c is taught by an academic staff member a , then c also involves a .
The converse is not necessarily true: A tutor is involved in a course, but does not teach it.
- P is a subproperty of Q if $Q(x, y)$ whenever $P(x, y)$

RDFS: Class hierarchies II



Note: Figure from "A Semantic Web Primer"

- Formal language for defining RDF Schema: RDF!
This means: modeling primitives of RDF Schema are defined using resources and properties - RDF.

Example (How to define "lecturer is subClassOf academicStaff")

- define resources "lecturer", "subClassOf", "academicStaff"
- define "subClassOf" to be a property
- write the triple "(subClassOf, lecturer, academicStaff)"

RDFS: Core Classes

- **rdfs:Resource** The class of all resources
- **rdfs:Class** The class of all classes
- **rdfs:Literal** The class of all literals (strings)
Currently: only data type of RDF/RDFS is string
- **rdf:Property** The class of all properties
- **rdf:Statement** The class of all reified statements

RDFS: Core Properties

- **rdf:type** relates a resource to its class (the resource is declared to be an instance of that class)
- **rdfs:subClassOf** relates a class to one of its superclasses; all instances of a class are instances of its superclass
- **rdfs:subPropertyOf** relates a property to one of its superproperties

rdfs:subClassOf and **rdfs:subPropertyOf** are transitive.

RDFS: Restricting Properties

- **rdfs:domain** specifies the domain of a property - the class of resources that may appear as subjects
- **rdfs:range** specifies the range of a property - the class of resources that may appear as values

Example (Range and domain)

```
<rdf:Property rdf:ID="phone">  
  <rdfs:domain rdf:resource="#staffMember"/>  
  <rdfs:range rdf:resource="&rdf;Literal"/>  
</rdf:Property>
```

- **rdfs:ConstraintResource** the class of all constraints
- **rdfs:ConstraintProperty** specifies all properties that define constraints.
It has only two instances: `rdfs:domain` and `rdfs:range`.
It is defined to be a subclass of `rdfs:ConstraintResource` and `rdf:Property`

Properties for Reification

- **rdf:subject** relates a reified statement to its subject
- **rdf:predicate** relates a reified statement to its predicate
- **rdf:object** relates a reified statement to its object

Container

- **rdf:Bag**
- **rdf:Seq**
- **rdf:Alt**
- **rdfs:Container** the superclass of all container classes, especially of Bag, Alt, Seq.

Utility Properties

- **rdfs:seeAlso** relates a resource to another resource that explains it
- **rdfs:isDefinedBy** subproperty of **rdfs:seeAlso**, relates a resource to the resource where its definition is found
- **rdfs:comment** Comments
- **rdfs:label** "human-friendly" label (name); also nice for naming the nodes in a graphical representation of RDF

Example (Lecturers at a university)

```
<rdfs:Class rdf:ID="staffMember">
  <rdfs:comment>The class of staff members</rdfs:comment>
</rdfs:Class>

<rdfs:Class rdf:ID="academicStaffMember">
  <rdfs:comment>The class of academic staff members
</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#staffMember"/>
</rdfs:Class>

<rdfs:Class rdf:ID="lecturer">
  <rdfs:comment>
    The class of lecturers.
    All lecturers are academic staff members.
  </rdfs:comment>
  <rdfs:subClassOf rdf:resource="#academicStaffMember"/>
</rdfs:Class>

<rdfs:Class rdf:ID="course">
  <rdfs:comment>The class of courses</rdfs:comment>
</rdfs:Class>
```

Example (Lecturers at a university)

```
<rdf:Property rdf:ID="isTaughtBy">
  <rdfs:comment>Assigns lecturers to courses.</rdfs:comment>
  <rdfs:domain rdf:resource="#course"/>
  <rdfs:range rdf:resource="#lecturer"/>
</rdf:Property>

<rdf:Property rdf:ID="teaches">
  <rdfs:comment>Assigns courses to lecturers.</rdfs:comment>
  <rdfs:domain rdf:resource="#lecturer"/>
  <rdfs:range rdf:resource="#course"/>
</rdf:Property>
```

Some instances:

```
<uni:lecturer rdf:ID="949318"
  uni:name="David Billington"
  uni:title="Associate Professor">
  <uni:teaches rdf:resource="#CIT1111"/>
  <uni:teaches rdf:resource="#CIT3112"/>
</uni:lecturer>
```

Example (Lecturers at a university)

```
<uni:lecturer rdf:ID="949352"  
  uni:name="Grigoris Antoniou"  
  uni:title="Professor">  
  <uni:teaches rdf:resource="#CIT1112"/>  
  <uni:teaches rdf:resource="#CIT1113"/>  
</uni:lecturer>  
  
<uni:course rdf:ID="CIT1111"  
  uni:courseName="Discrete Mathematics">  
  <uni:isTaughtBy rdf:resource="#949318"/>  
</uni:course>  
  
<uni:course rdf:ID="CIT1112"  
  uni:courseName="Concrete Mathematics">  
  <uni:isTaughtBy rdf:resource="#949352"/>  
</uni:course>
```

Example (The motor vehicles example, <http://www.w3.org/2000/03/example/vehicles>)

```
<!-- This is an example RDF Schema -->
<!-- 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://www.w3.org/2000/03/example/vehicles#".
      This allows us to use abbreviations such as xyz:MotorVehicle to
      refer unambiguously to the RDF class 'MotorVehicle'. -->

<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#">

  <rdf:Description ID="MotorVehicle">
    <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf
      rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Description>

  <rdf:Description ID="PassengerVehicle">
    <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
  </rdf:Description>
```

Example (The motor vehicles example,
<http://www.w3.org/2000/03/example/vehicles>)

```
<rdf:Description ID="Truck">  
  <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>  
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>  
</rdf:Description>
```

```
<rdf:Description ID="Van">  
  <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>  
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>  
</rdf:Description>
```

```
<rdf:Description ID="MiniVan">  
  <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>  
  <rdfs:subClassOf rdf:resource="#Van"/>  
  <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>  
</rdf:Description>
```

Example (The motor vehicles example, <http://www.w3.org/2000/03/example/vehicles>)

```
<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="http://www.w3.org/2000/03/example/classes#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/2000/03/example/classes#Number"/>
</rdf:Description>
</rdf:RDF>
```