BASI DI DATI II – 2 modulo
Parte VII: RDF

Based on tutorials of O. Lassila, R.R. Swick, J. Cowan, D. Brickley, R.V. Guha, W3C

Prof. Riccardo Torlone

Università Roma Tre

Outline

- RDF
- RDF Schema

What is RDF ?

- Resource Description Framework (RDF) is a foundation for processing data and metadata in the Web
- It supports interoperability between applications that exchange machine-understandable information on the Web
- RDF emphasises facilities to enable automated processing of Web resources
- It is a mechanism for describing resources that makes no assumptions about a particular application domain

Why RDF ?

- for resource discovery to provide better search engine capabilities
- for describing the content and content relationships available at a particular Web site
- for intelligent software agents to facilitate knowledge sharing and exchange
- 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

Basic Objects in RDF Data Model

- Resources
- Properties
- Statements

Resources

- All things being described by RDF expressions are called resources:
  - entire Web page;
  - part of a Web page (e.g. a specific XML element within the document source);
  - whole collection of pages (e.g. an entire Web site);
  - an object that is not directly accessible via the Web (e.g. a printed book).

Resources and URIs

- A resource can be anything that has identity
- Uniform Resource Identifiers (URI) provide a simple and extensible means for identifying a resource
- Not all resources are network "retrievable"; e.g., human beings, corporations, and books in a library can also be considered resources

Properties and statements

- A property is a specific aspect, characteristic, attribute, or relation used to describe a resource
- Each property has a specific meaning, defines its permitted values, the types of resources it can describe, and its relationship with other properties
- A specific resource together with a named property plus the value of that property for that resource is an RDF statement

RDF statement

- Subject of an RDF statement is a resource
- Predicate of an RDF statement is a property of a resource
- Object of an RDF statement is the value of a property of a resource

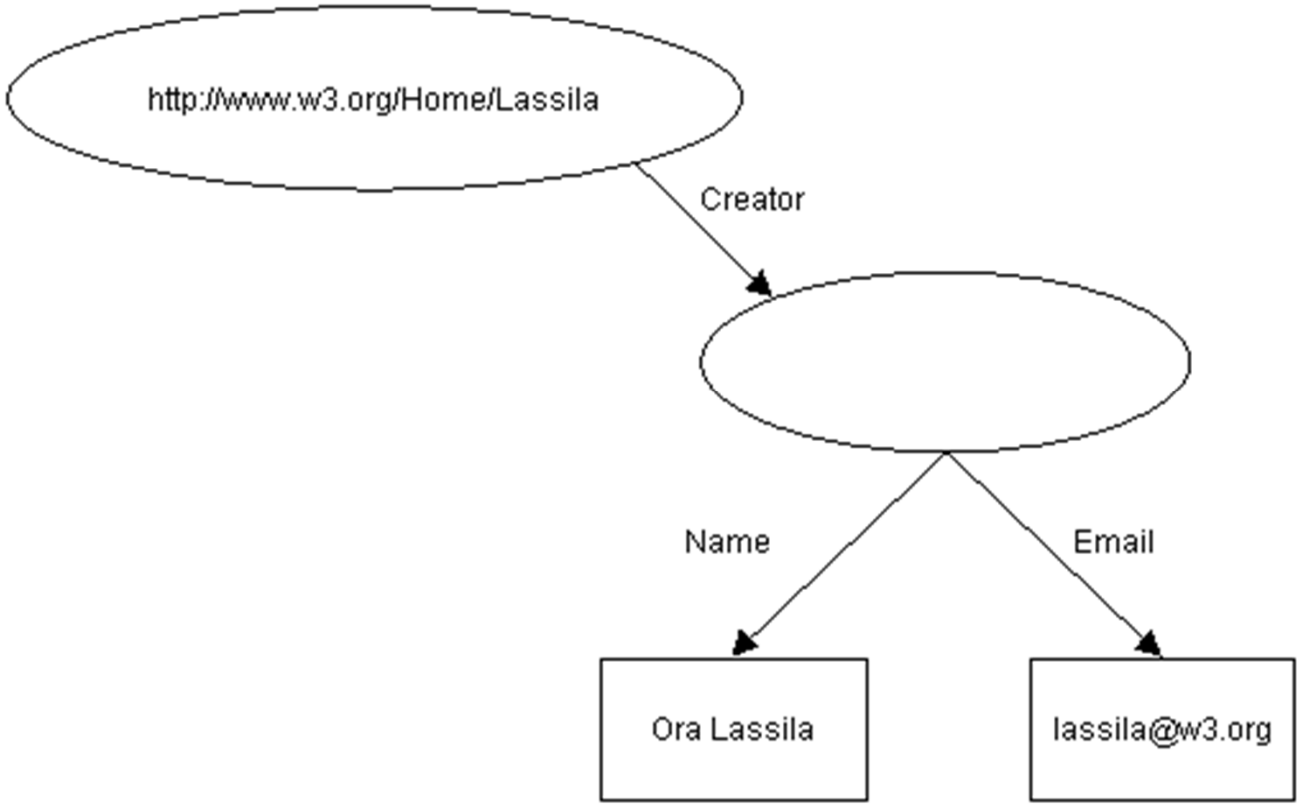
Example of RDF Statement

- Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila.

Subject (resource)	http://www.w3.org/Home/Lassila
Predicate (property)	Creator
Object (literal)	“Ora Lassila”

Property with Structural Value Example (1)

- The individual whose name is Ora Lassila, email <lassila@w3.org>, is the creator of http://www.w3.org/Home/Lassila.



Property with Structural Value Example (2)

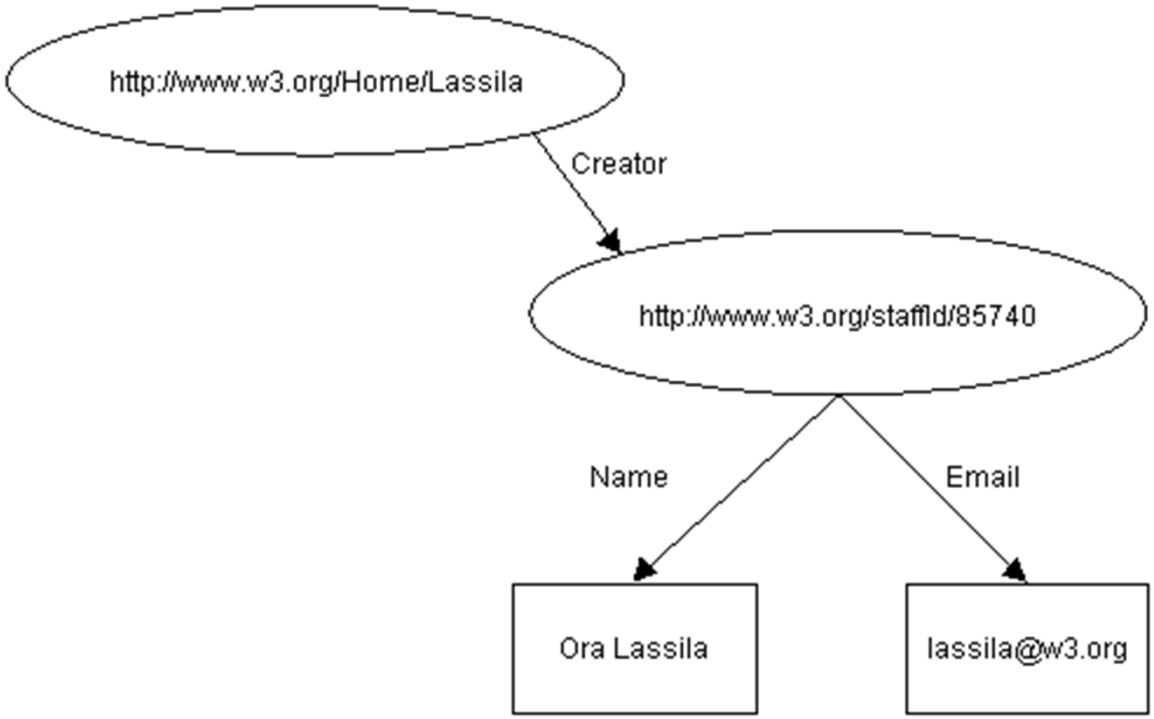
Subject (resource)	http://www.w3.org/Home/Lassila
Predicate (property)	Creator
Object (literal)	SOMETHING

Subject (resource)	SOMETHING
Predicate (property)	Name
Object (literal)	“Ora Lassila”

Subject (resource)	SOMETHING
Predicate (property)	Email
Object (literal)	lassila@w3.org

Property with Structural Value Example (3)

- The individual referred to by employee id 85740 is named Ora Lassila and has the email address lassila@w3.org.
- The resource http://www.w3.org/Home/Lassila was created by this individual.



Property with Structural Value Example (4)

Subject (resource)	http://www.w3.org/Home/Lassila
Predicate (property)	Creator
Object (literal)	http://www.w3.org/staffid/85740

Subject (resource)	http://www.w3.org/staffid/85740
Predicate (property)	Name
Object (literal)	“Ora Lassila”

Subject (resource)	http://www.w3.org/staffid/85740
Predicate (property)	Email
Object (literal)	lassila@w3.org

RDF Serialisation Syntax

[1] RDF ::= ['<rdf:RDF>'] description* ['</rdf:RDF>']

[2] description ::= '<rdf:Description' idAboutAttr? '>' propertyElt* '</rdf:Description> '

[3] idAboutAttr ::= idAttr | aboutAttr

[4] aboutAttr ::= 'about="' URI-reference '"'

[5] idAttr ::= 'ID="' IDsymbol '"'

[6] propertyElt ::= '<' propName '>' value '</' propName '>' | '<' propName resourceAttr '/>'

[7] propName ::= Qname

[8] value ::= description | string

[9] resourceAttr ::= 'resource="' URI-reference '"'

[10] Qname ::= [NSprefix ':'] name

[11] URI-reference ::= string, interpreted per [URI]

[12] IDsymbol ::= (any legal XML name symbol)

[13] name ::= (any legal XML name symbol)

[14] NSprefix ::= (any legal XML namespace prefix)

[15] string ::= (any XML text, with "<", ">", and "&" escaped)

RDF Statement Example

- Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila.

RDF Statement Example

- Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila.

```
<rdf:RDF>
  <rdf:Description about=
        "http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

a specific namespace prefix as reference to an *ontology* where predicates are defined, e.g.:
`xmlns:s="http://description.org/schema/"`

RDF Statement Example

- Versione completa

```
<?xml version="1.0"?>
<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>Ora Lassila</s:Creator>
   </rdf:Description>
</rdf:RDF>
```

RDF Abbreviated Syntax

[2a] description ::= '<rdf:Description' idAboutAttr? propAttr* '/>'
| '<rdf:Description' idAboutAttr? propAttr* '>'
propertyElt* '</rdf:Description>'
| typedNode

[6a] propertyElt ::= '<' propName '>' value '</' propName '>'
| '<' propName resourceAttr? propAttr* '/>'

[16] propAttr ::= propName '="' string '"'
(with embedded quotes escaped)

[17] typedNode ::= '<' typeName idAboutAttr? propAttr* '/>'
| '<' typeName idAboutAttr? propAttr* '>'
property* '</' typeName '>'

RDF Abbreviated Syntax

- While the serialisation syntax shows the structure of an RDF model most clearly, often it is desirable to use a more compact XML form.

```
<rdf:RDF>
   <rdf:Description
            about="http://www.w3.org/Home/Lassila"
            s:Creator="Ora Lassila" />
</rdf:RDF>
```

Serialisation vs. Abbreviated Syntax

- The individual referred to by employee id 85740 is named Ora Lassila and has the email address lassila@w3.org. The resource http://www.w3.org/Home/Lassila was created by this individual.

```
<rdf:RDF>
    <rdf:Description about="http://www.w3.org/Home/Lassila">
      <s:Creator>
        <rdf:Description about="http://www.w3.org/staffId/85740">
          <v:Name>Ora Lassila</v:Name>
          <v:Email>lassila@w3.org</v:Email>
        </rdf:Description>
      </s:Creator>
    </rdf:Description>
</rdf:RDF>
```

Serialisation syntax used

```
<rdf:RDF>
    <rdf:Description about="http://www.w3.org/Home/Lassila">
      <s:Creator rdf:resource="http://www.w3.org/staffId/85740"
         v:Name="Ora Lassila"
         v:Email="lassila@w3.org" />
    </rdf:Description>
</rdf:RDF>
```

Abbreviated syntax used

An alternative notation for blank nodes

```
<rdf:RDF>
    <rdf:Description about="http://www.w3.org/Home/Lassila">
       <s:Creator rdf:parseType="Resource">
          <v:Name>Ora Lassila</v:Name>
          <v:Email>lassila@w3.org</v:Email>
       </s:Creator>
    </rdf:Description>
</rdf:RDF>

<rdf:RDF>
    <rdf:Description about="http://www.w3.org/Home/Lassila">
      <s:Creator rdf:parseType="Resource"
         v:Name="Ora Lassila"
         v:Email="lassila@w3.org" />
    </rdf:Description>
</rdf:RDF>
```

RDF N3 syntax

- Notation3, or N3 as it is more commonly known, is a shorthand non-XML serialization of RDF models, designed with human-readability in mind: N3 is much more compact and readable than XML RDF notation. The format is being developed by Tim Berners-Lee and others from the Semantic Web community.

```
<rdf:RDF
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://en.wikipedia.org/wiki/Tony_Benn">
    <dc:title>Tony Benn</dc:title>
    <dc:publisher>Wikipedia</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

RDF sample in XML notation

```
@prefix dc:<http://purl.org/dc/elements/1.1/>

<http://en.wikipedia.org/wiki/Tony_Benn>
    dc:title "Tony Benn";
    dc:publisher "Wikipedia".
```

RDF sample in N3 notation

RDF N3 basics

- Collection of statements like:
  <#pat> <#knows> <#jo> .
- Shortcuts: (5 statements in 1)
  <#pat> <#child> <#al>, <#chaz>, <#mo> ;
  <#age> 24 ;
  <#eyecolor> "blue" .
- Blank nodes:
  <#pat> <#child> [<#age> 4] , [<#age> 3].
- Sharing concepts:
  <#bill> <http://purl.org/dc/elements/1.1/title> "Primer".
- Using prefix:
  @prefix dc: <http://purl.org/dc/elements/1.1/> .
  <#bill> dc:title "Primer".
- Using the default prefix (for the document we are writing):
  @prefix : <#> .
  :pat :child [:age 4] , [:age 3].

Some N3 syntax specifics

N3 Syntactic Sugar: Commas

```
<http://www.princeton.edu> ed:hasDept <http://www.cs.princeton.edu> ,
     <http://www.math.princeton.edu> , <http://history.princeton.edu> .
```

N3 Syntactic Sugar: Semicolons

```
<http://princeton.edu> geo:lat "40.35" ; geo:long "-74.66" .
```

Blank Nodes in Notation 3

```
<http://www.w3.org/2000/01/rdf-schema#> edu:hasDept
     [ dc:title "Department of Computer Science" ] ,
     [ dc:title "Department of Psychology" ] .
```

Ontological statements in N3

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
:Professor  a  rdfs:Class
<http://www.cs.jyu.fi/ai/vagan> a  :Professor
```

RDF N3 examples

- Simple statement
  - `:John :Loves :Mary .`
- Reified statement
  - `[:John :Loves :Mary] :accordingTo :Bill .`
- Goal statement:
  - `gb:I gb:want [:John :Loves :Mary] .`
- The prefix `gb:` is used here to denote the ontology of S-APL.

Containers

- Frequently it is necessary to refer to a collection of resources
- RDF containers are used to hold such lists of resources or literals.
- There are three types of a container:
  - bag
  - sequence
  - alternative

Container Syntax

[18] container ::= sequence | bag | alternative

[19] sequence ::= '<rdf:Seq' idAttr? '>' member* '</rdf:Seq>'

[20] bag ::= '<rdf:Bag' idAttr? '>' member* '</rdf:Bag>'

[21] alternative ::= '<rdf:Alt' idAttr? '>' member+ '</rdf:Alt>'

[22] member ::= referencedItem | inlineItem

[23] referencedItem ::= '<rdf:li' resourceAttr '/>'

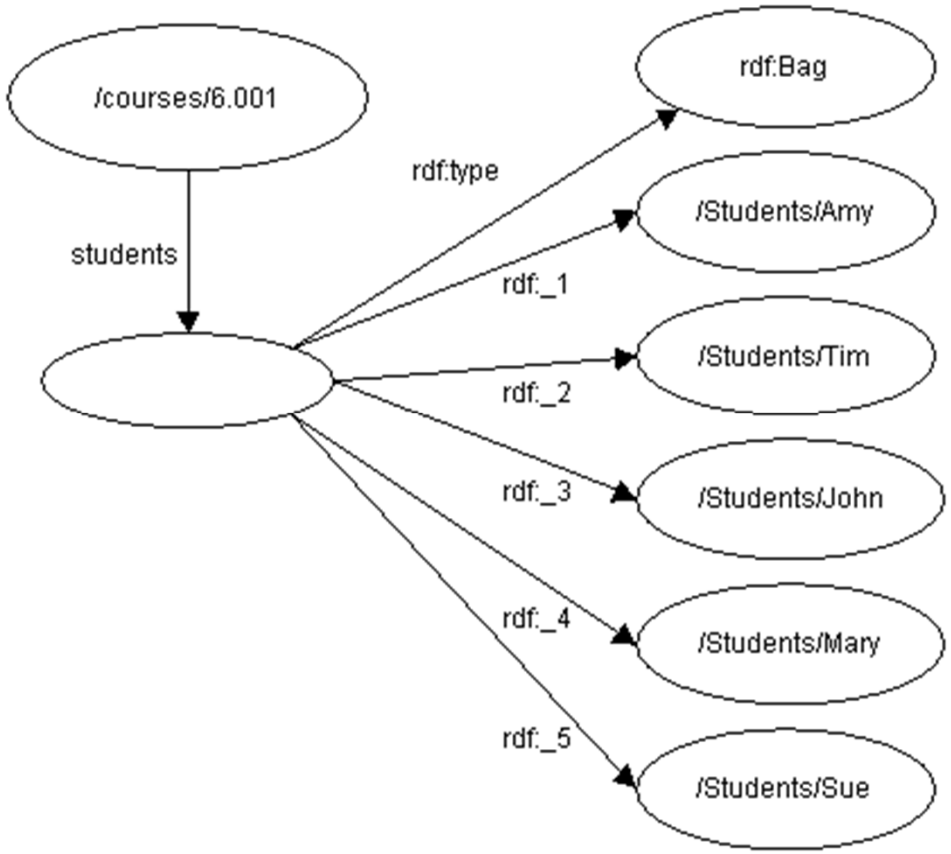
[24] inlineItem ::= '<rdf:li>' value '</rdf:li>'

Containers. Bag.

- An unordered list of resources or literals.
- Bags are used to declare that a property has multiple values and that there is no significance to the order in which the values are given.
- Bag might be used to give a list of part numbers where the order of processing the parts does not matter. Duplicate values are permitted.

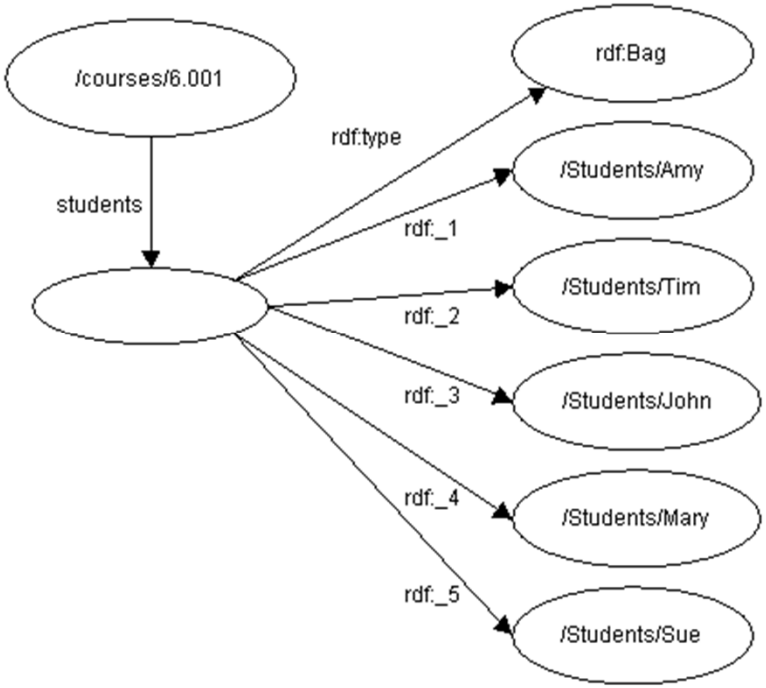
Bag Example (1)

- The students in course 6.001 are Amy, Tim, John, Mary, and Sue



Bag Example (2)

- The graph has eight nodes and seven arcs. The first node is the resource /courses/6.001. An arc labelled students connects this node to an unnamed node. An arc labelled rdf:type connects the unnamed node to a node labelled rdf:Bag. Five additional arcs labelled rdf:_1, rdf:_2, rdf:_3, rdf:_4, and rdf:_5 connect the unnamed node to nodes labelled, respectively, /Students/Amy, /Students/Tim, /Students/John, /Students/Mary, and /Students/Sue. All the nodes are represented as ovals.

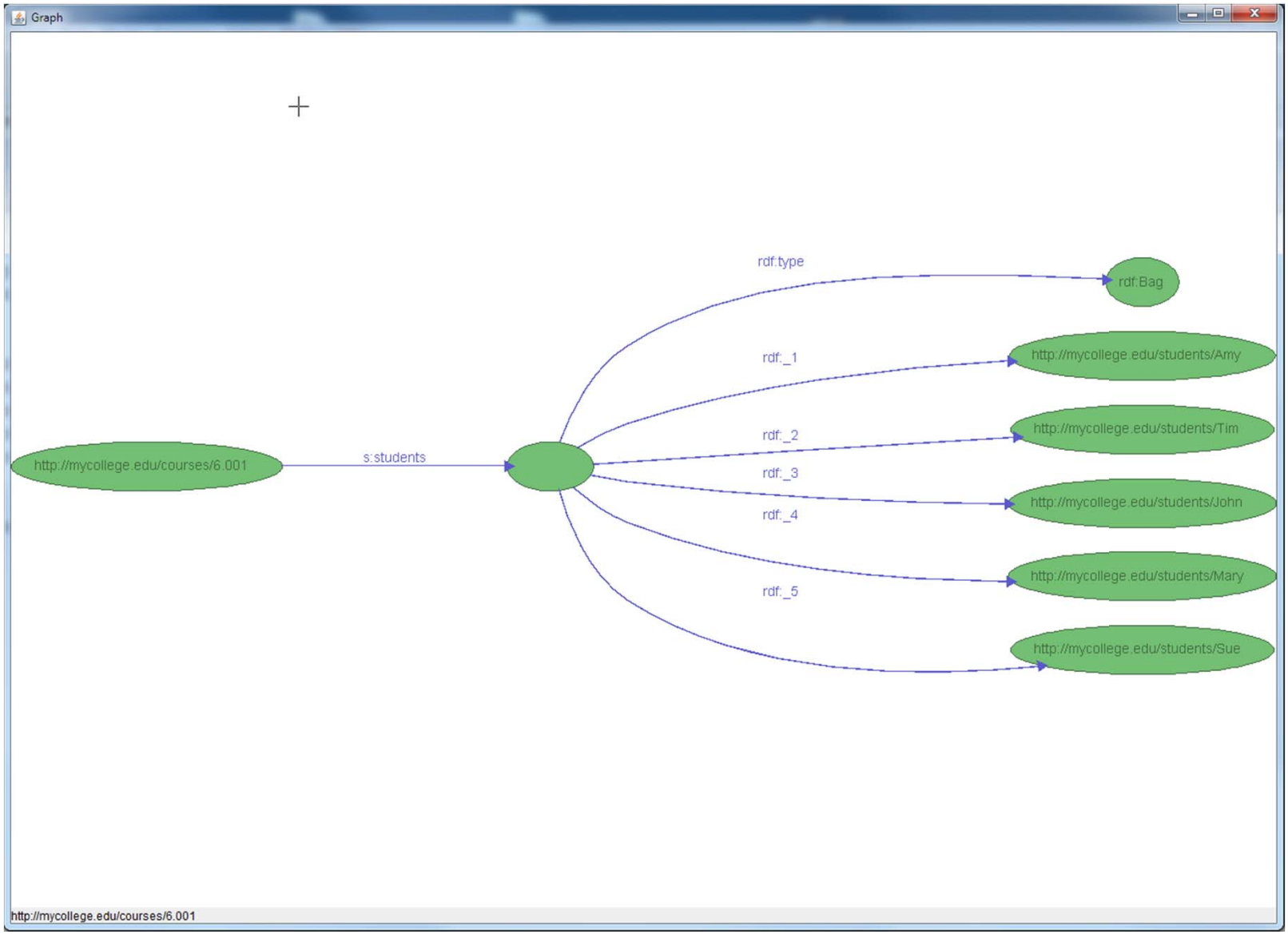


Bag Example (3)

- The students in course 6.001 are Amy, Tim, John, Mary, and Sue

```
<rdf:RDF>
  <rdf:Description about="http://mycollege.edu/courses/6.001">
  <s:students>
    <rdf:Bag>
      <rdf:li resource="http://mycollege.edu/students/Amy"/>
      <rdf:li resource="http://mycollege.edu/students/Tim"/>
      <rdf:li resource="http://mycollege.edu/students/John"/>
      <rdf:li resource="http://mycollege.edu/students/Mary"/>
      <rdf:li resource="http://mycollege.edu/students/Sue"/>
    </rdf:Bag>
  </s:students>
  </rdf:Description>
</rdf:RDF>
```

Graphical representation in IsaViz



Containers: Sequence

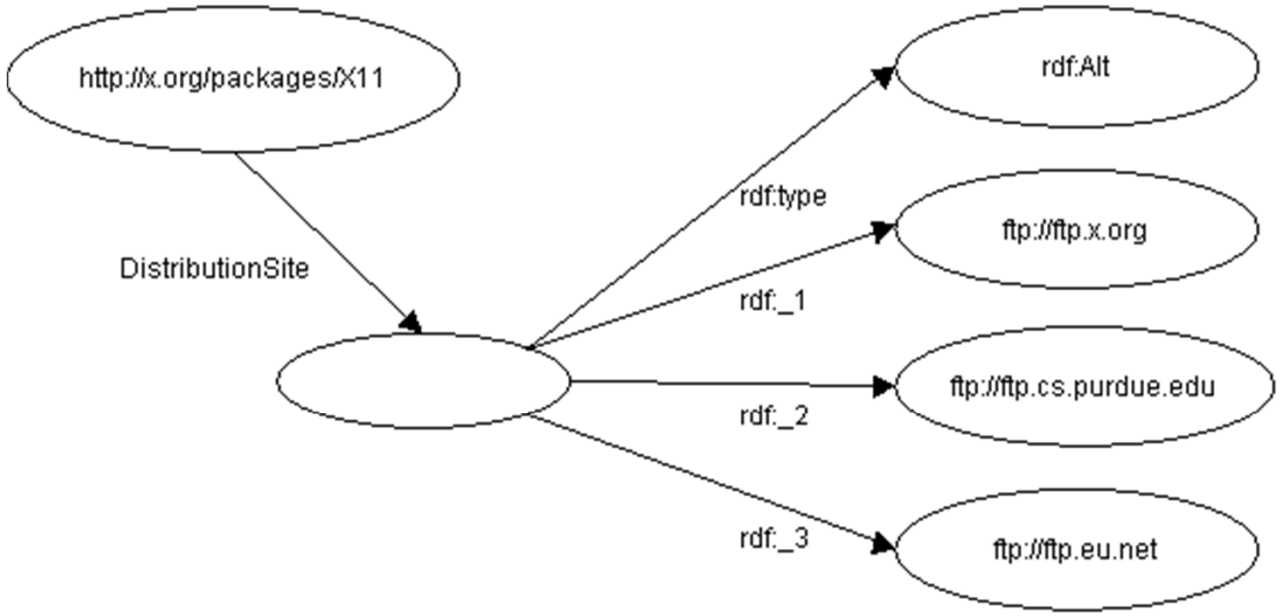
- An ordered list of resources or literals.
- Sequence is used to declare that a property has multiple values and that the order of the values is significant.
- Sequence might be used, for example, to preserve an alphabetical ordering of values.
- Duplicate values are permitted.

Containers: Alternative

- A list of resources or literals that represent alternatives for the (single) value of a property.
- An application using a property whose value is an Alternative collection is aware that it can choose any one of the items in the list as appropriate

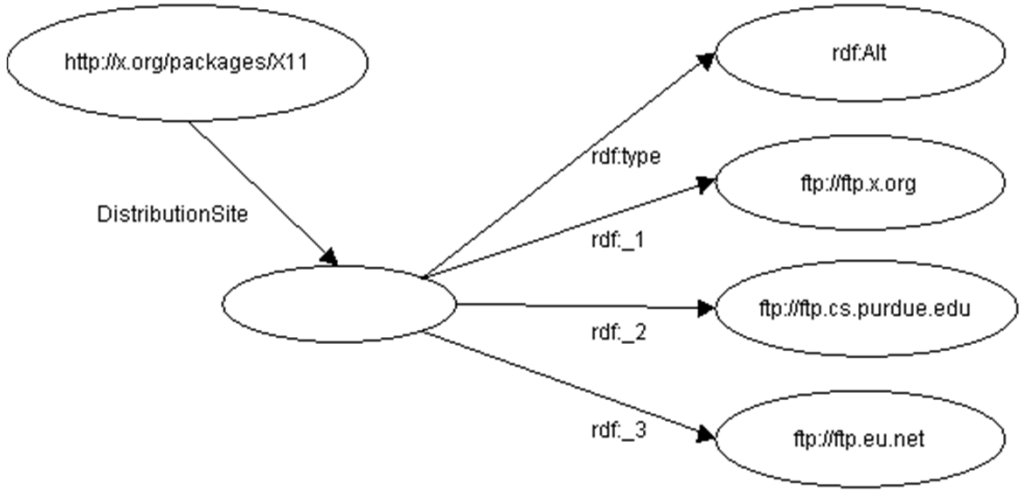
Alternative Example (1)

- The source code for X11 may be found at ftp.x.org, ftp.cs.purdue.edu, or ftp.eu.net



Alternative Example (2)

- The graph has six nodes and five arcs. The first node is the resource http://x.org/packages/X11. An arc labelled DistributionSite connects this node to an unnamed node. An arc labelled rdf:type connects the unnamed node to a node labelled rdf:Alt. Three additional arcs labelled rdf:_1, rdf:_2, and rdf:_3 connect the unnamed node to nodes labelled, respectively, ftp.x.org, ftp.cs.purdue.edu, and ftp.eu.net. All the nodes are represented as ovals



Alternative Example (3)

- The source code for X11 may be found at ftp.x.org, ftp.cs.purdue.edu, or ftp.eu.net.

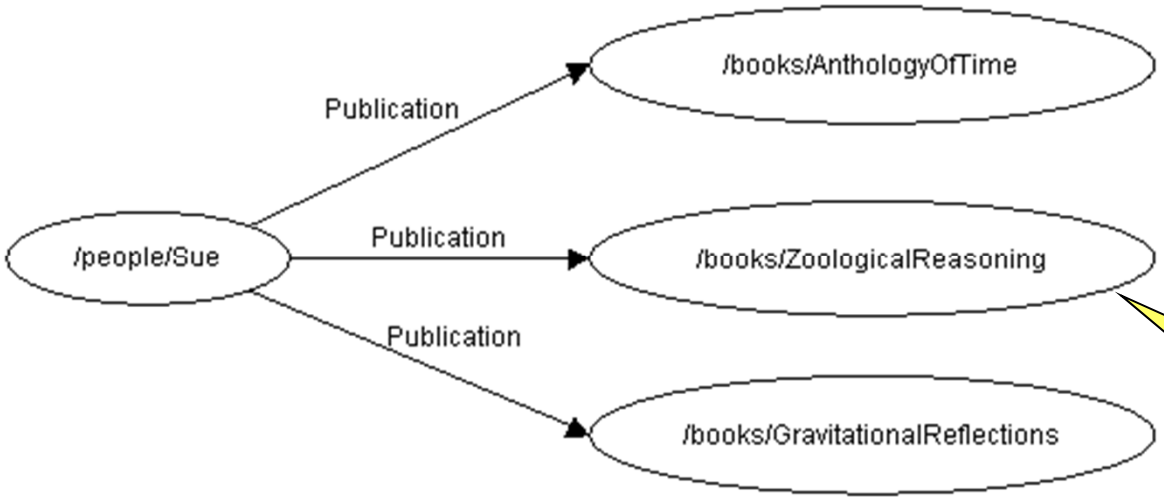
```
<rdf:RDF>
    <rdf:Description about="http://x.org/packages/X11">
      <s:DistributionSite>
        <rdf:Alt>
          <rdf:li resource="ftp://ftp.x.org"/>
          <rdf:li resource="ftp://ftp.cs.purdue.edu"/>
          <rdf:li resource="ftp://ftp.eu.net"/>
        </rdf:Alt>
      </s:DistributionSite>
    </rdf:Description>
</rdf:RDF>
```

Repeated Property Example

- Sue has written "Anthology of Time", "Zoological Reasoning", "Gravitational Reflections".

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:s="http://description.org/schema/">
  <rdf:Description rdf:about=
       "http://www.books.org/books/AnthologyOfTime">
    <dc:creator rdf:Resource="http://www.writers.org/people/Sue" />
    <dc:title>Anthology of Time</dc:title>
  </rdf:Description>
  <rdf:Description rdf:about=
       "http://www.books.org/books/ZoologicalReasoning">
    <dc:creator rdf:Resource="http://www.writers.org/people/Sue" />
    <dc:title>Zoological Reasoning</dc:title>
  </rdf:Description>
  <rdf:Description rdf:about=
       "http://www.books.org/books/GravitationalReflections">
    <dc:creator rdf:Resource="http://www.writers.org/people/Sue"/>
    <dc:title>Gravitational Reflections</dc:Title>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.writers.org/people/Sue">
    <s:Name>Sue</s:Name>
  </rdf:Description>
</rdf:RDF>
```

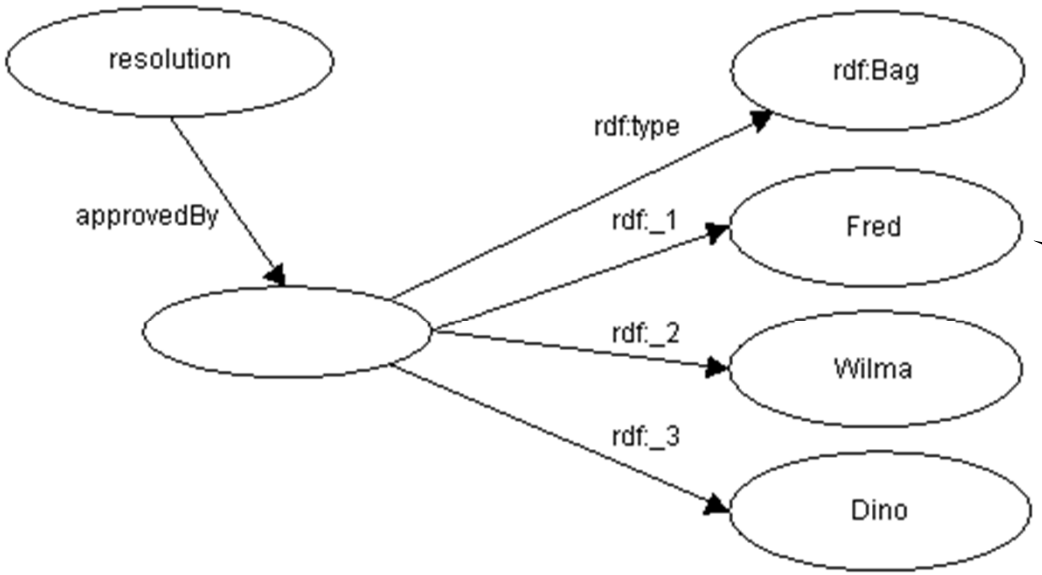
Containers vs. Repeated Properties (1)



There is no stated relationship between the publications other than that they were written by the same person

Containers vs. Repeated Properties (2)

- The committee of Fred, Wilma, and Dino approved the resolution.



three committee members as a whole voted in a certain manner; it does not necessarily state that each committee member voted in favour of the resolution.

Statements about Statements (1)

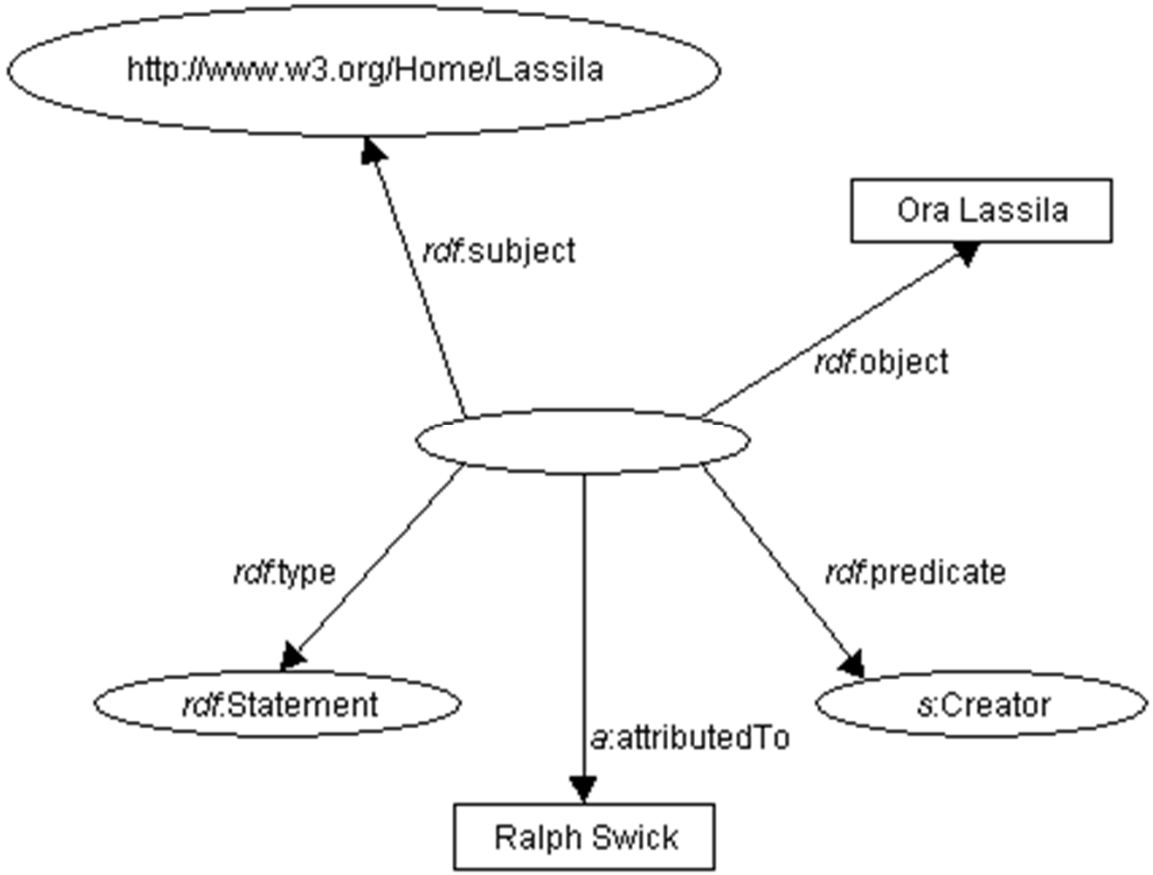
- For example, let us consider the sentence:
  - “Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila”.
- RDF would regard this sentence as a fact.
- If, instead, we write the sentence:
  - “Ralph Swick says that Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila”
- … we have said nothing about the resource http://www.w3.org/Home/Lassila; instead, we have expressed a fact about a statement Ralph has made.

Statements about Statements (2)

- To model statements, RDF defines the following properties:
  - subject
    - The subject property identifies the resource being described by the modelled statement; that is, the value of the subject property is the resource about which the original statement was made (e.g., http://www.w3.org/Home/Lassila).
  - predicate
    - The predicate property identifies the original property in the modelled statement. The value of the predicate property is a resource representing the specific property in the original statement (in our example, creator).
  - object
    - The object property identifies the property value in the modelled statement. The value of the object property is the object in the original statement (in our example, "Ora Lassila").
  - type
    - The value of the type property describes the type of the new resource. All reified statements are instances of RDF:Statement; that is, they have a type property whose object is RDF:Statement.

Statements about Statements (3)

- “Ralph Swick says that Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila”



Statements about Statements (4)

- “Ralph Swick says that Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila”

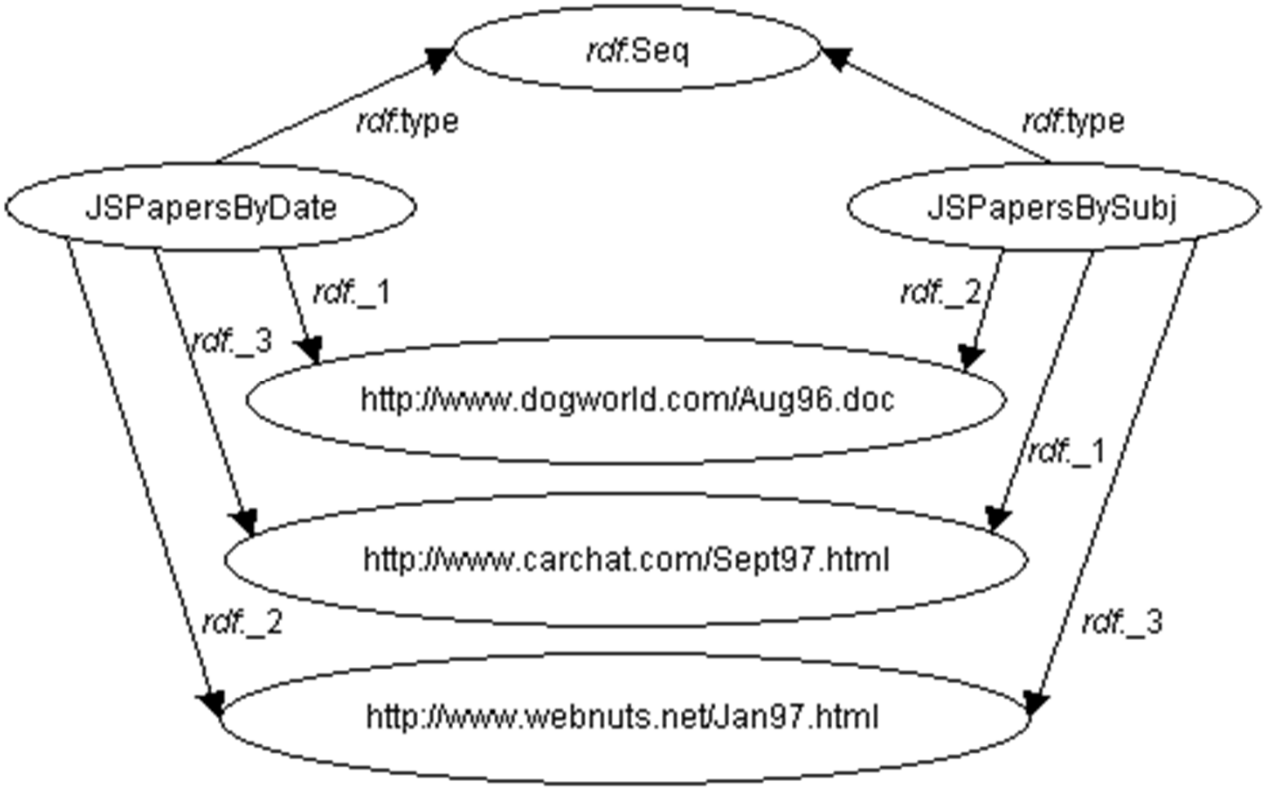
```
<rdf:RDF
 xmlns:rdf="http://w3.org/TR/1999/PR-rdf-syntax-19990105#"
 xmlns:a="http://description.org/schema/">
 <rdf:Description>
   <rdf:subject resource="http://www.w3.org/Home/Lassila" />
   <rdf:predicate resource="http://description.org/schema#Creator"/>
   <rdf:object>Ora Lassila</rdf:object>
   <rdf:type resource="http://w3.org/TR/1999/PR-rdf-syntax-
                                              19990105#Statement"/>
   <a:attributedTo>Ralph Swick</a:attributedTo>
 </rdf:Description>
</rdf:RDF>
```

Statements about containers

```
<rdf:Description about="http://www.cs.jyu.fi/vagan/courses/ITKS544">
  <s:lectures>
    <rdf:Bag ID="pages">
      <rdf:li rdf:resource=
          "http://www.cs.jyu.fi/vagan/courses/ITKS544/Lecture_1.ppt">
      <rdf:li rdf:resource=
          "http://www.cs.jyu.fi/vagan/courses/ITKS544/Lecture_2.ppt">
      ...
      <rdf:li rdf:resource=
          "http://www.cs.jyu.fi/vagan/courses/ITKS544/Lecture_8.ppt">
    </rdf:Bag>
  </s:lectures>
</rdf:Description>

<rdf:Description about="#pages">
       <dc:creator>  Vagan Terziyan  </dc:creator>
</rdf:Description>
```

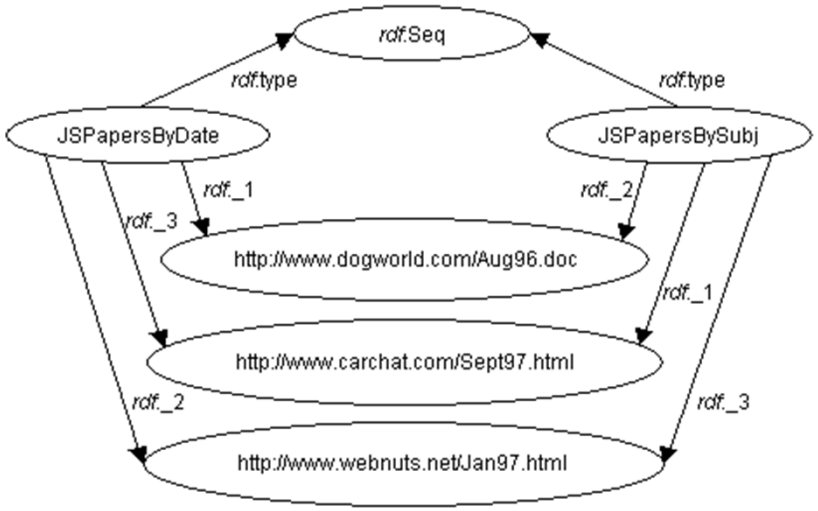
Sharing Values between Sequences (1)

- Consider the case of specifying 3 collected works of an author, sorted once by publication date and sorted again alphabetically by subject.



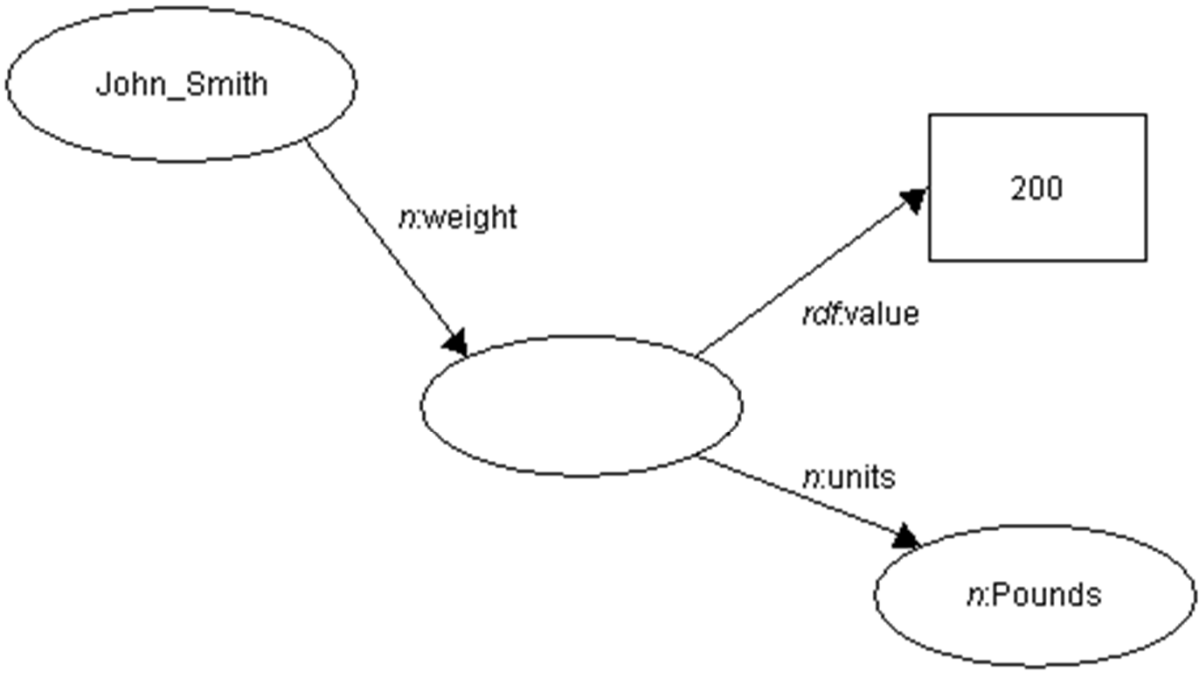
Sharing Values between Sequences (2)

```
<RDF xmlns="http://w3.org/TR/1999/PR-rdf-syntax-19990105#">
    <Seq ID="JSPapersByDate">
      <li resource="http://www.dogworld.com/Aug96.doc"/>
      <li resource="http://www.webnuts.net/Jan97.html"/>
      <li resource="http://www.carchat.com/Sept97.html"/>
    </Seq>
    <Seq ID="JSPapersBySubj">
      <li resource="http://www.carchat.com/Sept97.html"/>
      <li resource="http://www.dogworld.com/Aug96.doc"/>
      <li resource="http://www.webnuts.net/Jan97.html"/>
    </Seq>
</RDF>
```



Ternary Relation (1)

- John Smith's weight is 200 pounds



Ternary Relation (2)

- John Smith's weight is 200 pounds

```
<RDF
  xmlns="http://w3.org/TR/1999/PR-rdf-syntax-19990105#"
  xmlns:rdf="http://w3.org/TR/1999/PR-rdf-syntax-19990105#"
  xmlns:n="http://www.nist.gov/units/">
  <Description about="John_Smith">
    <n:weight rdf:parseType="Resource">
      <rdf:value>200</rdf:value>
      <n:units rdf:resource="http://www.nist.gov/units/Pounds"/>
    </n:weight>
  </Description>
</RDF>
```

Why RDFS ?

- Resource description communities require the ability to describe the organization of certain kinds of resources.
- 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 but may also define the kinds of resources being described.
- RDF schema lets developers define a particular vocabulary for RDF data and specify the kinds of object to which these attributes can be applied.

What is RDFS ?

- The RDF Schema is a collection of RDF resources that can be used to describe the schema of other RDF resources.
- The core schema vocabulary is defined in a namespace informally called 'rdfs', and identified by the URI reference:
  - http://www.w3.org/2000/01/rdf-schema#.
- Specification also uses the prefix 'rdf' to refer to the core RDF namespace:
  - http://www.w3.org/1999/02/22-rdf-syntax-ns#.
- Latest RDFS description (15 December 2003)
  - http://www.w3.org/TR/2003/PR-rdf-schema-20031215/

What is RDFS ?

- RDF Schema
  - Describe the organization of RDF data
  - Defines vocabulary for RDF
  - Organizes this vocabulary in a typed hierarchy
    - Class, subClassOf, type, Property, subPropertyOf
- Rich, web-based publication format for declaring semantics (XML for exchange)
- Capability to explicitly declare semantic relations between vocabulary terms

RDF Schemas

- Semantic network on the Web
- Nodes are identified by URIs
- Main constructs:
  - rdfs:Class
  - rdfs:Property
  - rdf:type

RDF Classes

- Are collections of similar Web resources
- Have URIs to identify them
- The special class “rdfs:Literal” consists of all possible RDF string values

Property-centric classes

- In typical OO classes, each class specifies completely what properties it has and what their types are
- In RDF classes, each property specifies what classes of subjects and objects it relates
- Therefore, new properties can be added to a class without modifying the class

Specifying classes

- To specify a class, create an RDF resource of type rdf:Class, for example:

```
<rdf:Class id="MyClass">
   <rdfs:label>My Class</rdfs:label>
   <rdfs:comment>Vagan Terziyan's demonstration
                        class</rdfs:comment>
</rdf:Class>
```

Specifying properties

- To specify a property, create an RDF resource of type rdf:Property, for example:

```
<rdf:Property id="myProperty">
   <rdfs:comment>Vagan Terziyan's demo
                              property</rdfs:comment>
   <rdfs:domain resource="#MyClass"/>
   <rdfs:range resource=
      "http://www.w3.org/2000/01/rdf-schema#Literal"/>
<rdf:Property>
```

Domain and Range of a Property

- "rdfs:domain" specifies the domain of a property (the class of its subjects); if unknown, anything can be a subject
  - More formally: if a pair (x,y) is an instance of P, then x is an instance of the class domain.
- rdfs:range specifies the range of a property (the single class of its objects); if unknown, anything can be an object
  - More formally: a pair (x,y) can only be an instance of P if y is an instance of the calss range

Other RDF constructs (1)

- “rdfs:subClassOf” relates a subclass to its superclass
- “rdfs:subPropertyOf” relates a subproperty to its superproperty
- “rdfs:seeAlso” relates a resource to another resource explaining it
- “rdfs:isDefinedBy” the definition of the subject resource

Other RDF constructs (2)

- “rdf:subject” is the property relating a reified statement to its subject (resource)
- “rdf:predicate” is the property relating a reified statement to its predicate (property)
- “rdf:object” is the property relating a reified statement to its object (value)

Other RDF constructs (3)

- "rdfs:label" specifies a human-readable name for this Class, Property, or whatever
- "rdfs:comment" specifies human-readable documentation

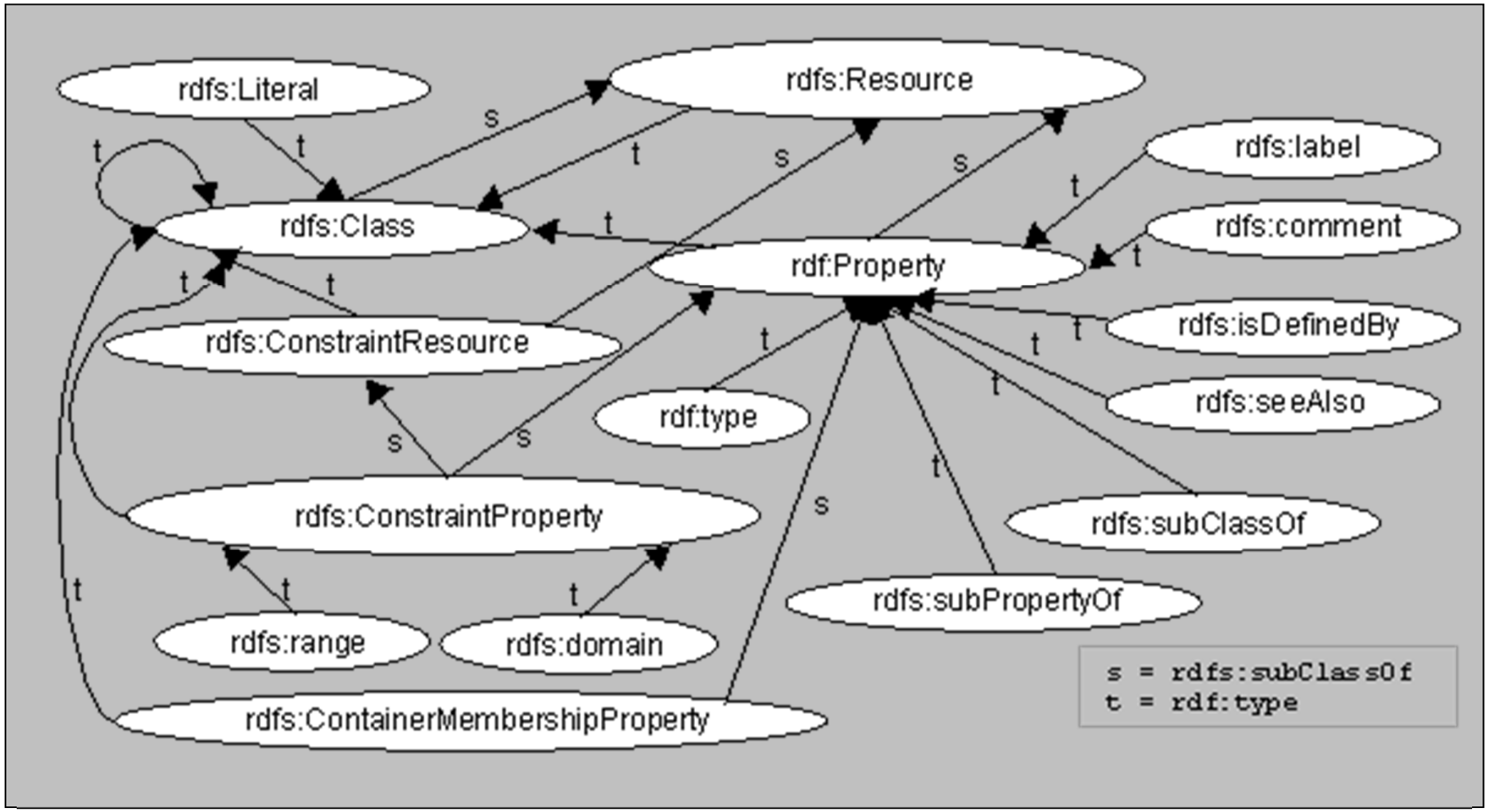
Predefined classes (1)

- “rdfs:Resource” is the class of all resources
- “rdfs:Literal” is the class of all strings
- “rdfs:Class” is the class of all classes
- “rdfs:Property” is the class of all properties
- “rdf:Statement” is the class of all asserted RDF statements

Predefined classes (2)

- “rdfs:Container” is the superclass of all container classes
- “rdf:Bag”, “rdf:Seq”, “rdf:Alt” are the classes of Bags, Seqs, and Alts
- (Any other class that is a subclass of “rdfs:Container” can be used in RDF syntax in place of a standard container)

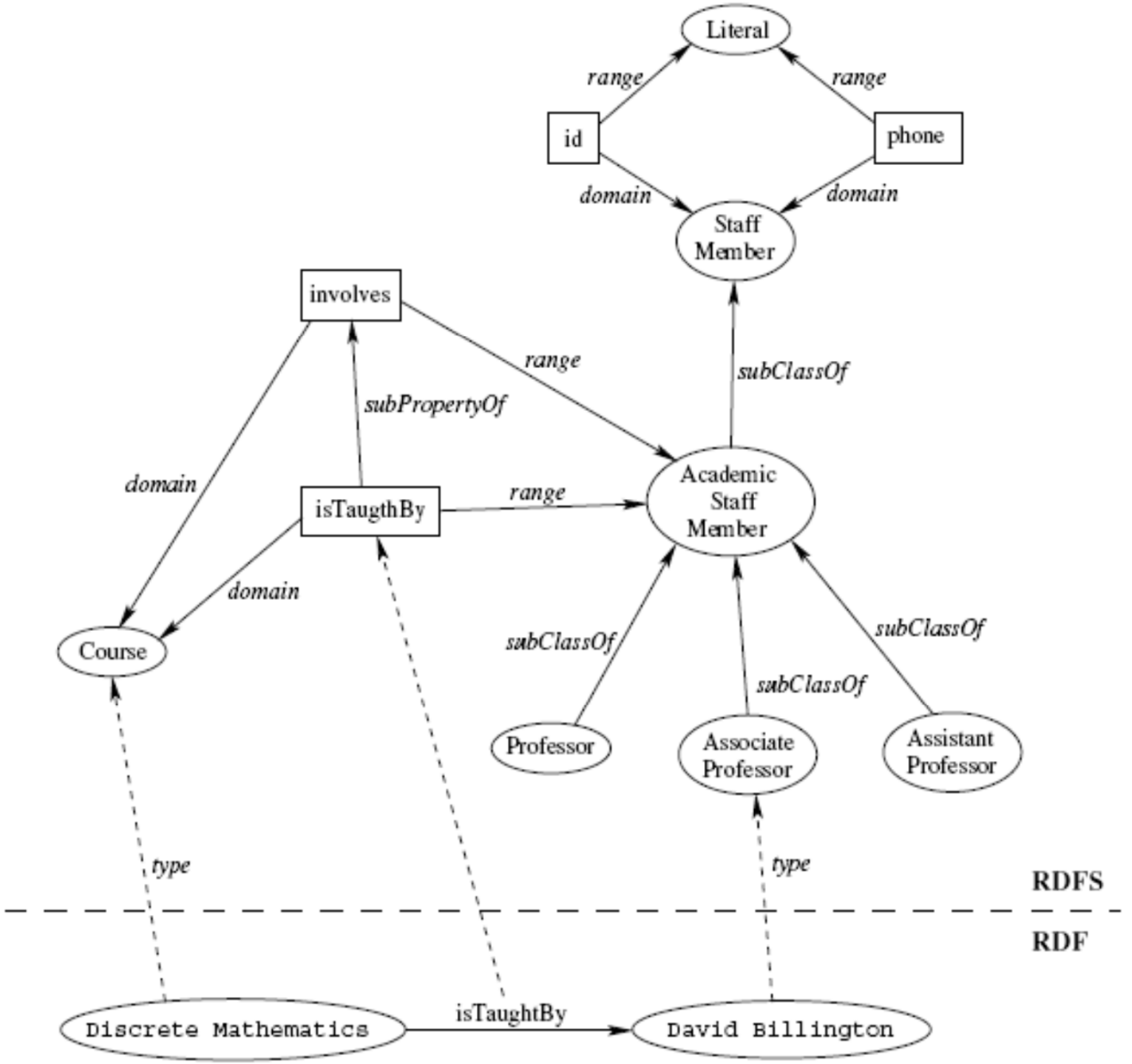
Predefined class hierarchy of RDFS (1)



Predefined class hierarchy of RDFS (2)

- Class hierarchy is shown using a "nodes and arcs" graph representation of the RDF data model.
- If one 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.
- If a resource is an instance of a class, then there is an rdf:type arc from the resource to the node representing the class.

RDF & RDFS example (by Emily Chen)



Dublin Core

- A set of fifteen basic properties for describing generalised Web resources
- ISO Standard 15836-2003 (February 2003): http://www.niso.org/international/SC4/n515.pdf
- The Dublin Core Metadata Initiative is an open forum engaged in the development of interoperable online metadata standards that support a broad range of purposes and business models: http://dublincore.org/

Dublin Core (1): Title

- Label: Title
- Definition: A name given to the resource.
- Comment: Typically, Title will be a name by which the resource is formally known.

Dublin Core (2): Creator

- Label: Creator
- Definition: An entity primarily responsible for making the content of the resource.
- Comment: Examples of Creator include a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.

Dublin Core (3): Subject

- Label: Subject and Keywords
- Definition: A topic of the content of the resource.
- Comment: Typically, Subject will be expressed as keywords, key phrases or classification codes that describe a topic of the resource. Recommended best practice is to select a value from a controlled vocabulary or formal classification scheme.

Dublin Core (4): Description

- Label: Description
- Definition: An account of the content of the resource.
- Comment: Examples of Description include, but is not limited to: an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content.

Dublin Core (5): Publisher

- Label: Publisher
- Definition: An entity responsible for making the resource available.
- Comment: Examples of Publisher include a person, an organization, or a service. Typically, the name of a Publisher should be used to indicate the entity.

Dublin Core (6): Contributor

- Label: Contributor
- Definition: An entity responsible for making contributions to the content of the resource.
- Comment: Examples of Contributor include a person, an organization, or a service. Typically, the name of a Contributor should be used to indicate the entity.

Dublin Core (7): Date

- Label: Date
- Definition: A date of an event in the lifecycle of the resource.
- Comment: Typically, Date will be associated with the creation or availability of the resource. Recommended best practice for encoding the date value is defined in a profile of ISO 8601 [http://dublincore.org/documents/dces/#w3cdtf] and includes (among others) dates of the form YYYY-MM-DD.

Dublin Core (8): Type

- Label: Resource Type
- Definition: The nature or genre of the content of the resource.
- Comment: Type includes terms describing general categories, functions, genres, or aggregation levels for content. Recommended best practice is to select a value from a controlled vocabulary (for example, the DCMI Type Vocabulary [http://dublincore.org/documents/dces/#dct1]). To describe the physical or digital manifestation of the resource, use the FORMAT element.

Dublin Core (9): Format

- Label: Format
- Definition: The physical or digital manifestation of the resource.
- Comment: Typically, Format may include the media-type or dimensions of the resource. Format may be used to identify the software, hardware, or other equipment needed to display or operate the resource. Examples of dimensions include size and duration. Recommended best practice is to select a value from a controlled vocabulary (for example, the list of Internet Media Types [http://dublincore.org/documents/dces/#mime] defining computer media formats).

Dublin Core (10): Identifier

- Label: Resource Identifier
- Definition: An unambiguous reference to the resource within a given context.
- Comment: Recommended best practice is to identify the resource by means of a string or number conforming to a formal identification system. Formal identification systems include but are not limited to the Uniform Resource Identifier (URI) (including the Uniform Resource Locator (URL)), the Digital Object Identifier (DOI) and the International Standard Book Number (ISBN).

Dublin Core (11): Source

- Label: Source
- Definition: A Reference to a resource from which the present resource is derived.
- Comment: The present resource may be derived from the Source resource in whole or in part. Recommended best practice is to identify the referenced resource by means of a string or number conforming to a formal identification system.

Dublin Core (12): Language

- Label: Language
- Definition: A language of the intellectual content of the resource.
- Comment: Recommended best practice is to use RFC 3066 [http://dublincore.org/documents/dces/#rfc3066] which, in conjunction with ISO639 [http://dublincore.org/documents/dces/#iso639]), defines two- and three-letter primary language tags with optional subtags. Examples include "en" or "eng" for English, "akk" for Akkadian", and "en-GB" for English used in the United Kingdom.

Dublin Core (13): Relation

- Label: Relation
- Definition: A reference to a related resource.
- Comment: Recommended best practice is to identify the referenced resource by means of a string or number conforming to a formal identification system.

Dublin Core (14): Coverage

- Label: Coverage
- Definition: The extent or scope of the content of the resource.
- Comment: Typically, Coverage will include spatial location (a place name or geographic coordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity). Recommended best practice is to select a value from a controlled vocabulary (for example, the Thesaurus of Geographic Names [http://www.getty.edu/research/tools/vocabulary/tgn/index.html]) and to use, where appropriate, named places or time periods in preference to numeric identifiers such as sets of coordinates or date ranges.

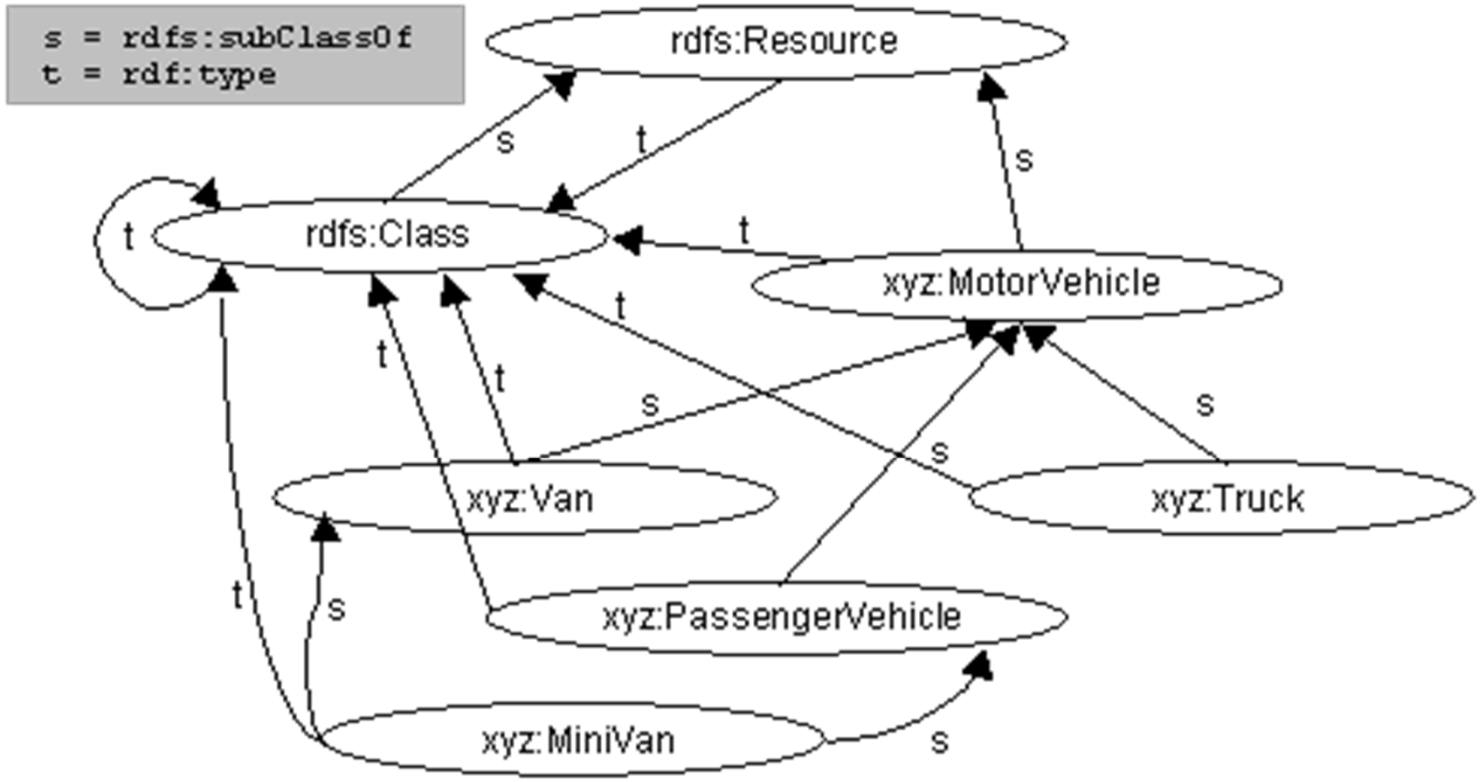
Dublin Core (15): Rights

- Label: Rights Management
- Definition: Information about rights held in and over the resource.
- Comment: Typically, Rights will contain a rights management statement for the resource, or reference a service providing such information. Rights information often encompasses Intellectual Property Rights (IPR), Copyright, and various Property Rights. If the Rights element is absent, no assumptions may be made about any rights held in or over the resource.

Dublin Core Example

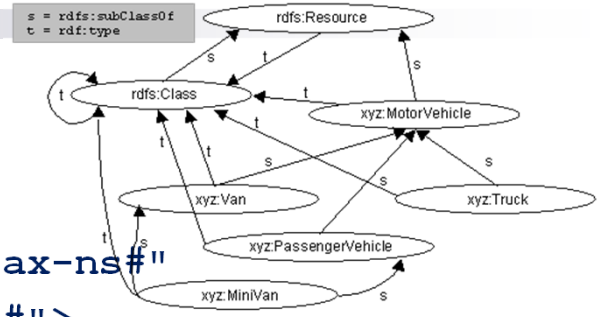
```
<?xml version="1.0"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:dc="http://purl.org/dc/elements/1.0/">
  <rdf:Description rdf:about="http://www.ukoln.ac.uk/metadata/
                              resources/dc/datamodel/WD-dc-rdf/">
       <dc:title> Guidance on expressing the Dublin Core within the
                  Resource Description Framework (RDF) </dc:title>
       <dc:creator> Eric Miller </dc:creator>
       <dc:creator> Paul Miller </dc:creator>
       <dc:creator> Dan Brickley </dc:creator>
       <dc:subject> Dublin Core; Resource Description Framework; RDF;
                  eXtensible Markup Language; XML </dc:subject>
       <dc:publisher> Dublin Core Metadata Initiative </dc:publisher>
       <dc:contributor> Dublin Core Data Model Working
                  Group </dc:contributor>
       <dc:date> 1999-07-01 </dc:date>
       <dc:format> text/html </dc:format>
       <dc:language> en </dc:language>
  </rdf:Description>
</rdf:RDF>
```

First example (1)



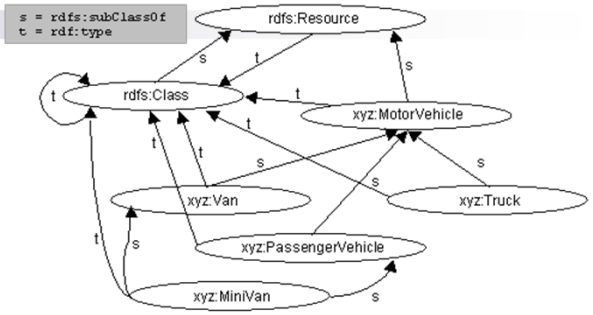
Example expresses the following class hierarchy. We first define a class MotorVehicle. We then define three subclasses of MotorVehicle, namely PassengerVehicle, Truck and Van. We then define a class Minivan which is a subclass of both Van and PassengerVehicle.

First example (2)



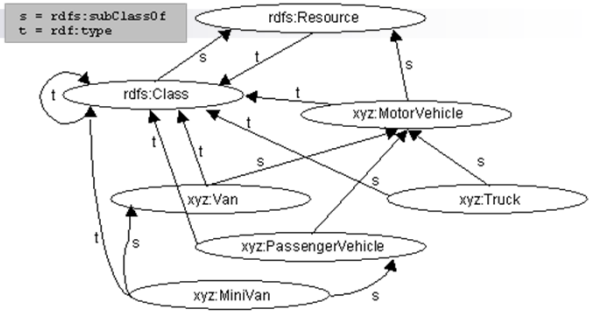

```
<rdf:RDF xml:lang="en"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
   <!-- Note: this RDF schema would typically be used in RDF instance
        data by referencing it with an XML namespace declaration, for
        example xmlns:xyz="http://www.w3.org/2000/03/example/vehicles#".
        This allows us to use abbreviations such as xyz:MotorVehicle to
        refer unambiguously to the RDF class 'MotorVehicle'. -->

   <rdf: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>

...
```

First example (3)




```
...
   <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>

</rdf:RDF>
```

First example (4)



Alternative notation:

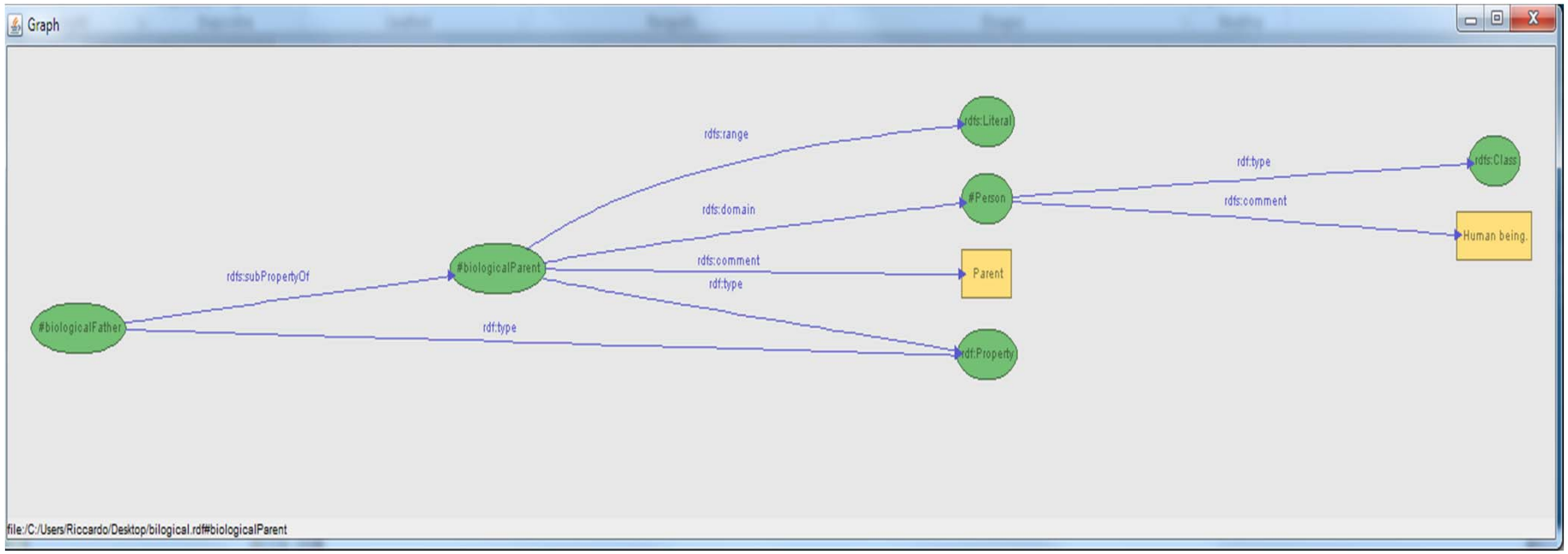
```
<rdf:RDF xml:lang="en"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

    <rdfs:Class rdf:ID="MotorVehicle">
        <rdfs:comment>Vehicle with a motor.</rdfs:comment>
    </rdfs:Class>

    <rdfs:Class rdf:ID="PassengerVehicle">
        <rdfs:comment>Vehicle with passagers.</rdfs:comment>
        <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
    </rdfs:Class>

    <rdfs:Class rdf:ID="Truck">
        <rdfs:comment>Truck.</rdfs:comment>
        <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
    </rdfs:Class>

    <rdfs:Class rdf:ID="Van">
        <rdfs:comment>Van.</rdfs:comment>
        <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
    </rdfs:Class>

    <rdfs:Class rdf:ID="MiniVan">
        <rdfs:comment>Little van.</rdfs:comment>
       <rdfs:subClassOf rdf:resource="#Van"/>
        <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
    </rdfs:Class>

</rdf:RDF>
```

Example of sub-property (1)

If the property biologicalFather is a subproperty 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/2000/01/rdf-schema#">

    <rdfs:Class rdf:ID="Person">
        <rdfs:comment>Human being.</rdfs:comment>
    </rdfs:Class>

    <rdf:Property rdf:ID="biologicalParent">
        <rdfs:domain rdf:resource="#Person"/>
        <rdfs:range rdf:resource="rdfs:Literal"/>
    </rdf:Property>

    <rdf:Property rdf: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>
```

Example of sub-property (2)



Another Example (1)

In this example, Person is a class with a corresponding human-readable description of "The class of people". Animal is a class presumed to be defined in another schema. All persons are animals, so we declare that Person is a subclass 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. Person's marital status is one of {Single, Married, Divorced, Widowed}. This is achieved through use of the rdfs: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.

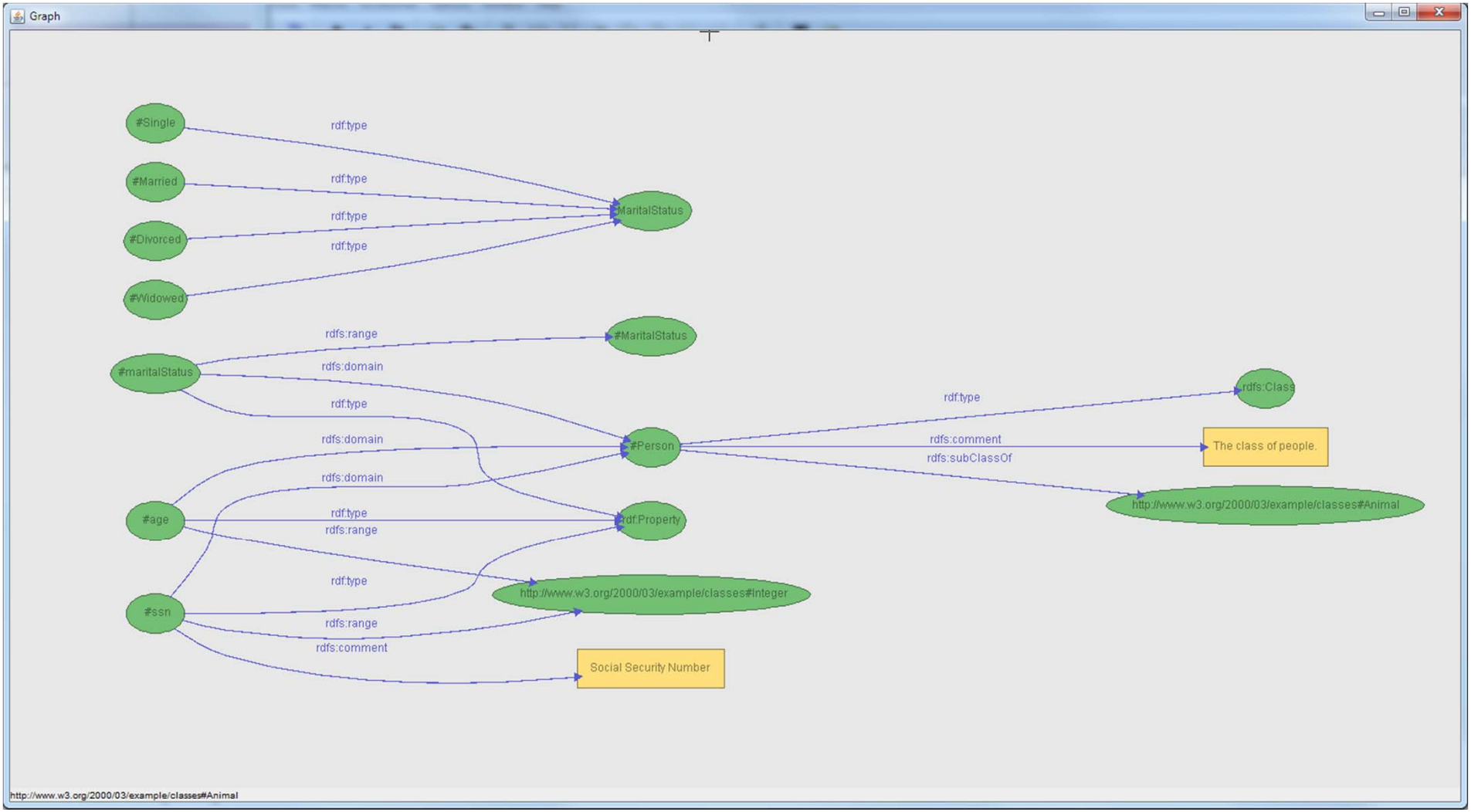
Another Example (2)

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

...
```

Another Example (3)

```
...
    <rdf:Property rdf: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 rdf:ID="age">
        <rdfs:range rdf:resource=
            "http://www.w3.org/2000/03/example/classes#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>
```

Another Example



Essential Online Resources

- http://www.w3.org/RDF/
- http://www.w3.org/TR/rdf-schema/