RDF and Semantic Mark-up of legislative texts

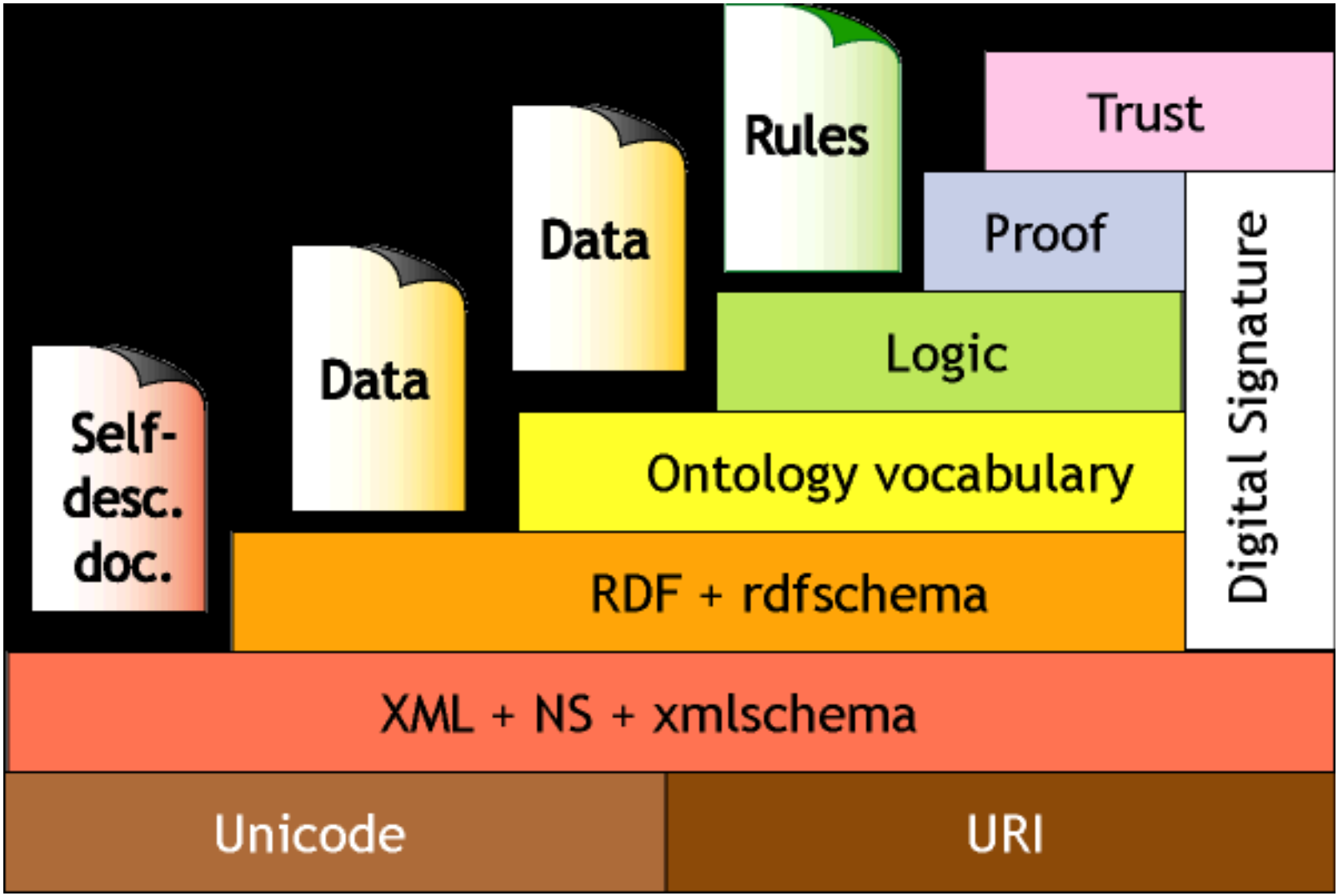
Enrico Francesconi

ITTIG-CNR
Institute of Legal Information Theory and Techniques
Italian National Research Council
http://www.ittig.cnr.it
francesconi@ittig.cnr.it

Overview

- Semantic Web and RDF
  - Why RDF and not just XML?
  - RDF Model, Graphs and Syntax (with examples)
  - RDF and RDF Schema (RDFS)
- Legal Semantic Web and RDF
  - A Semantic Model for Legislation (“Provision Model”)
  - A Provision Model in RDFS
  - Provisions Semantic Mark-up in RDF

The Semantic Web Layers



Why the Semantic Web?

- **Search on the Web**: problems due to the way in which information is stored
  - Problem 1:
    - distinction between information content and presentation
  - Problem 2:
    - different web documents may represent in different ways semantically related pieces of information
    - this leads to hard problems for “intelligent” information search on the Web

Separating content and presentation

- **Problem 1:** need to distinguish between information content and presentation; problem due to the HTML language
- Problem “solved” by current technology
  - XML
    - XML tags are used to express the “semantics” of the various pieces of information
  - XSL
    - stylesheets allow for separating formatting attributes from the information presented

XML Example

HTML

```
<H1>LeX Summer School 2012</H1>
  <UL>
    <LI>Location: Faculty of Law, Ravenna, University of Bologna</LI>
     <LI>Address: via Oberdan, Ravenna, Italy</LI>
  </UL>
```

XML

```
<course>
  <title>LeX Summer School 2012</title>
  <location>Faculty of Law, Ravenna, University of Bologna</location>
  <address>via Oberdan, Ravenna, Italy</address>
</course>
```

Why should I use RDF? Why not just XML?

- **Problem 2**: different web documents may represent in different ways semantically related pieces of information

- Such different XML documents might not share such common semantics

- RDF is a Mark-up language to express the “meaning” of such a piece of information in a shared way

A Statement and different XML Serializations

How would this information be typically represented in XML?

```
<author>
    <document>http://...</document>
    <name>John Smith</name>
</author>
```

or maybe

```
<document href="http://...">
  <author>John Smith</author>
</document>
```

or maybe

```
<document>
  <details>
   <uri>http://...</uri>
   <author>
      <name>John Smith</name>
   </author>
   </details>
</document>
```

or maybe

```
<document>
  <author>
   <uri>http://...</uri>
   <details>
      <name>John Smith</name>
   </details>
   </author>
</document>
```

or maybe

```
<document href="http://..." author="John Smith" />
```

RDF vs. XML

- Problems in querying the same semantics represented in different XML trees
- The query you write has to be independent of how the same semantics is represented
- Need of converting of all possible representations of a fact into one statement
- This is just what RDF does!
  - It gives you some standard ways of writing statements
  - However the same semantics occurs in an XML document, it is represented in the same way by using RDF statements
  - The same RDF tree results from many XML trees
  - Using RDF, the same parser can extract assertion components independently from the XML serialization

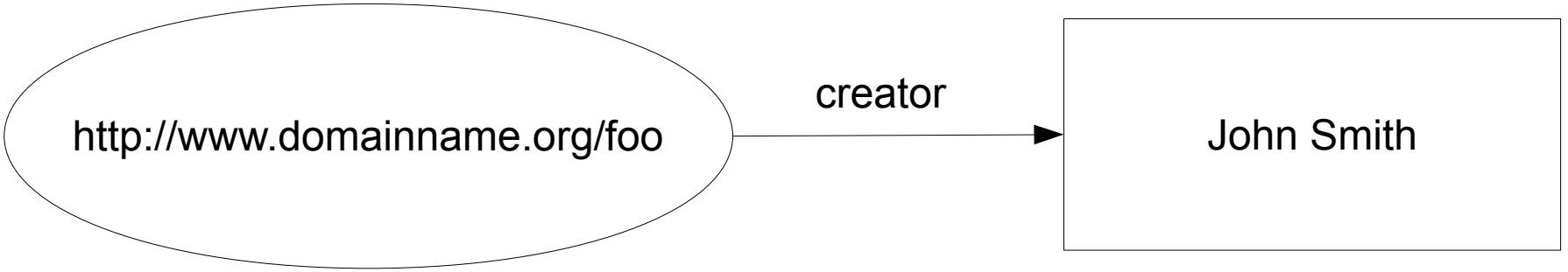
RDF Model and Syntax

- Language to describe resources
- It gives a way to describe resources using metadata (data about data)
- Provides interoperability between applications that exchange machine-understandable information on the Web
- Use XML as a syntax

Basic RDF Model

- Three object types:
  1. **Resources** - Things being described by RDF expressions, always named by URIs. Ex:
     - HTML Document
     - Specific XML element within the document source
     - Collection of pages
     - Books
     - etc
  2. **Properties** - Specific aspects, characteristics, attributes or relations used to describe a resource. (Ex: Creator, Title, Name, etc.)
  3. **Statements**
     - Resource (Subject) +
     - Property (Predicate) +
     - Property Value (Object)

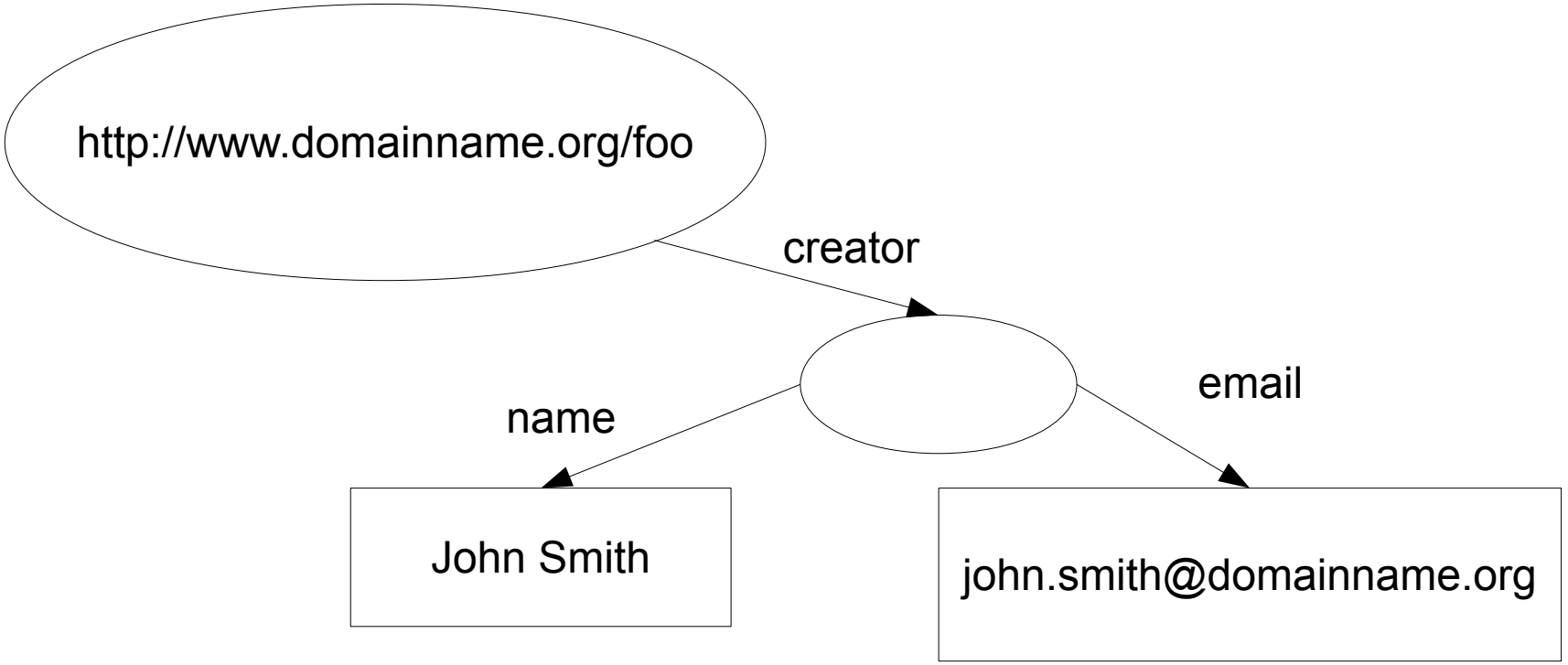
Statement Example (1)



John Smith is the **creator** of **http://www.domainname.org/foo**

- Subject (Resource) - http://www.domainname.org/foo
- Predicate (Property) - Creator
- Object (Property Value (Literal)) - “John Smith”

Statement Example (2)



- **John Smith** whose **e-mail** is **john.smith@domainname.org** is the **creator** of the web page **http://www.domainname.org/foo**

Basic XML Serialization Syntax

- The RDF root element and namespace:

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

The RDF **Description** Element

- Attributes of rdf:Description
  - about: refers to a URI of an existing resource
  - ID: signals the creation of a new resource

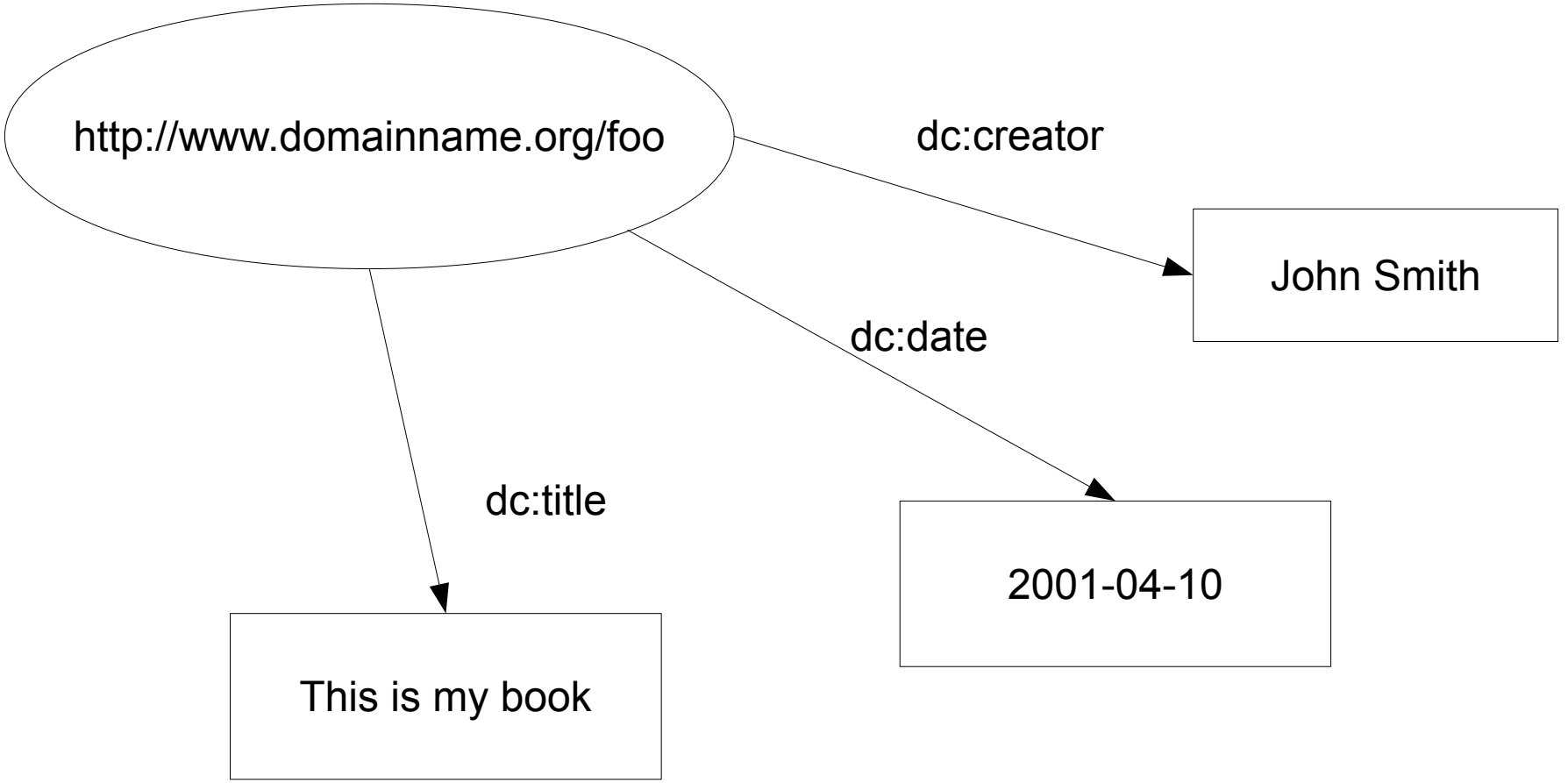
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="http://www.domainname.org/foo">
    ...
  </rdf:Description>
</rdf:RDF>
```

RDF **Properties**

- They describe the characteristics of a Resource
- Property names must be associated with a schema
  - Qualify property names with a namespace prefix
  - Associate a value to a Property
    <PropertyName> Value </PropertyName>
- Value: Description element or String

```
<?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/">
    <rdf:Description rdf:about="http://www.domainname.org/foo">
      <dc:creator>John Smith</dc:creator>
    </rdf:Description>
  </rdf:RDF>
```

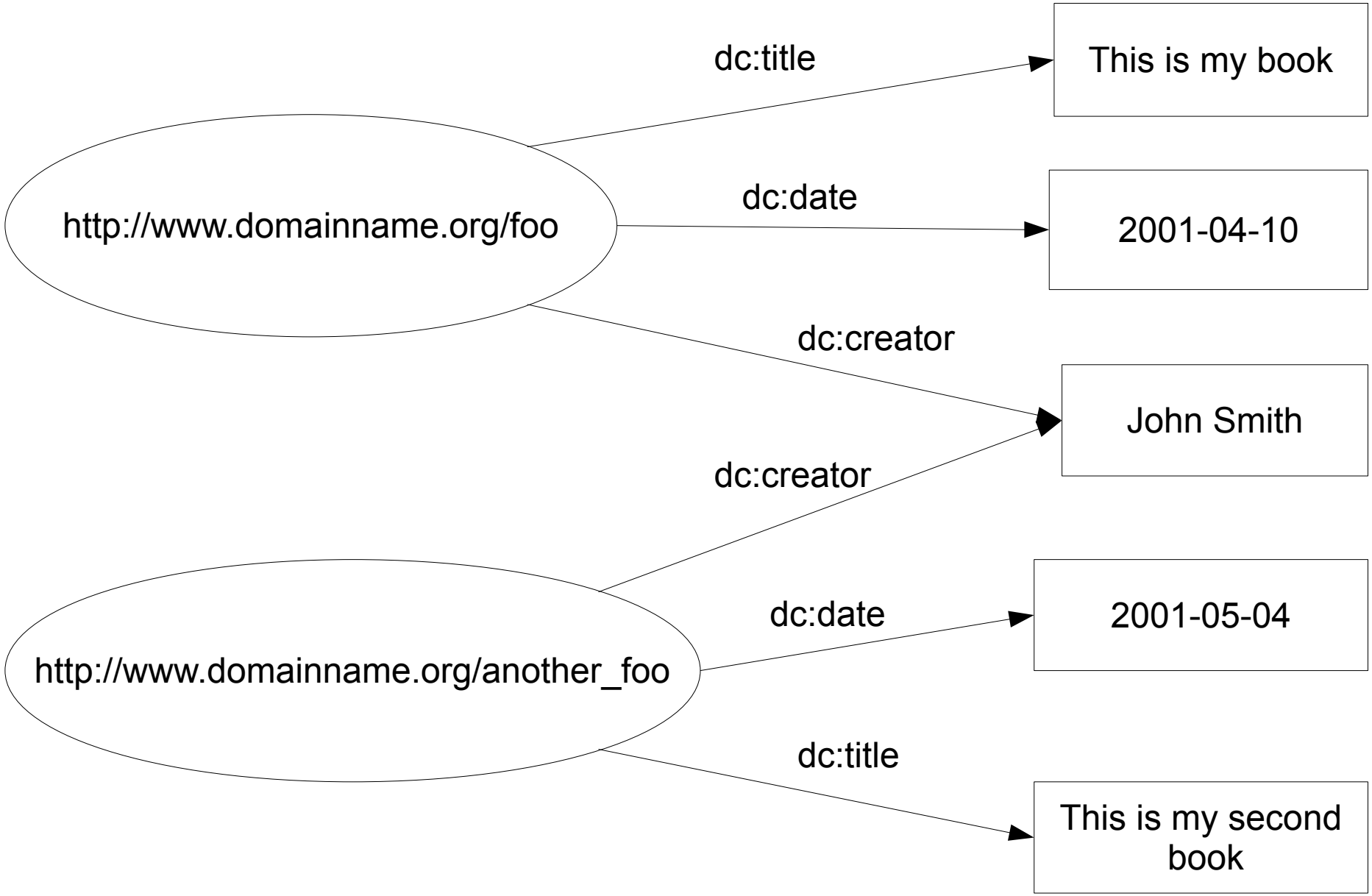
Describing Multiple Properties



Describing Multiple Properties

```
<?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/">
    <rdf:Description rdf:about="http://www.domainname.org/foo">
      <dc:creator>John Smith</dc:creator>
      <dc:title>This is my book</dc:title>
      <dc:date>2001-04-10</dc:date>
    </rdf:Description>
  </rdf:RDF>
```

Describing Multiple Resources



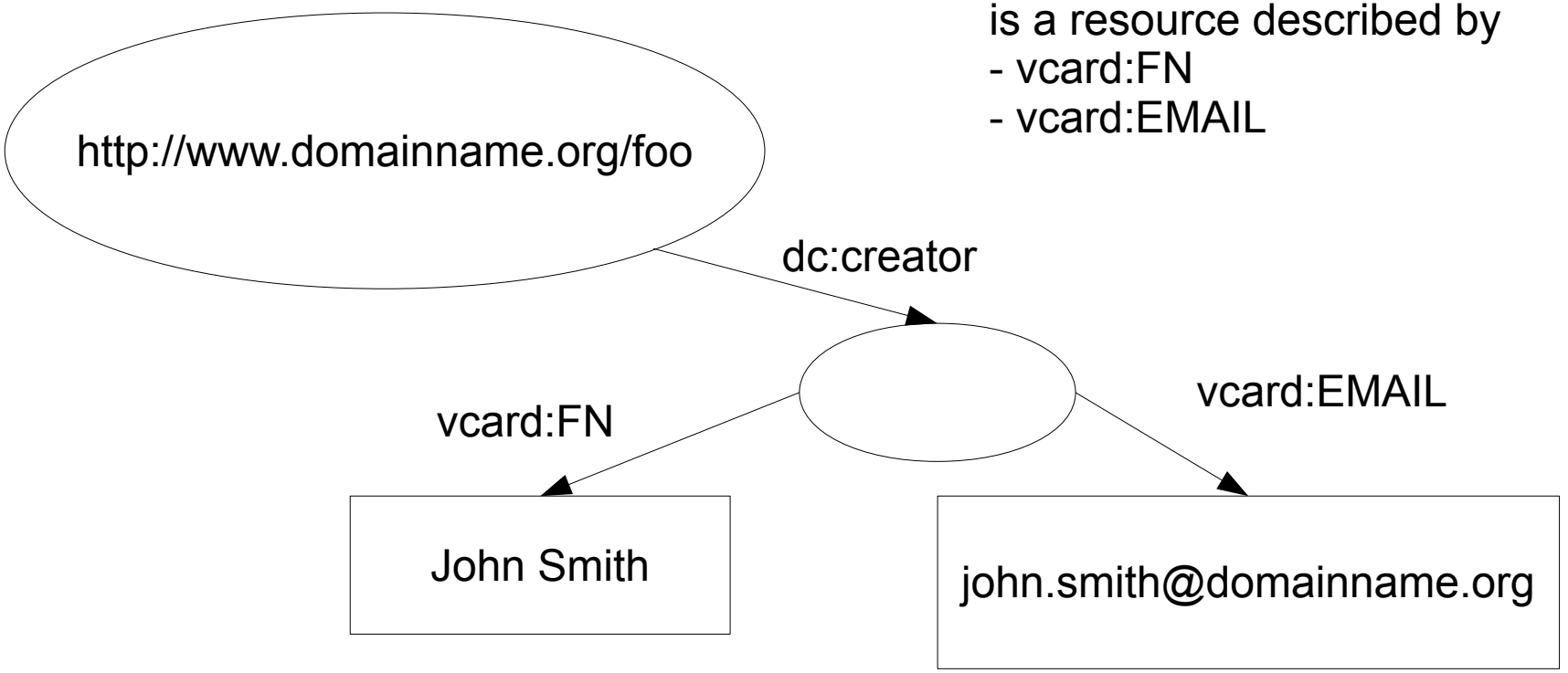
Describing Multiple Resources

- Use multiple Description elements

```
<?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/">

  <rdf:Description rdf:about="http://www.domainname.org/foo">
    <dc:creator>John Smith</dc:creator>
    <dc:title>This is my book</dc:title>
    <dc:date>2001-04-10</dc:date>
  </rdf:Description>

  <rdf:Description rdf:about="http://www.domainname.org/another_foo">
    <dc:creator>John Smith</dc:creator>
    <dc:title>This is my second book</dc:title>
    <dc:date>2001-05-04</dc:date>
  </rdf:Description>

</rdf:RDF>
```

Nesting Resources
(Property value as Description element)

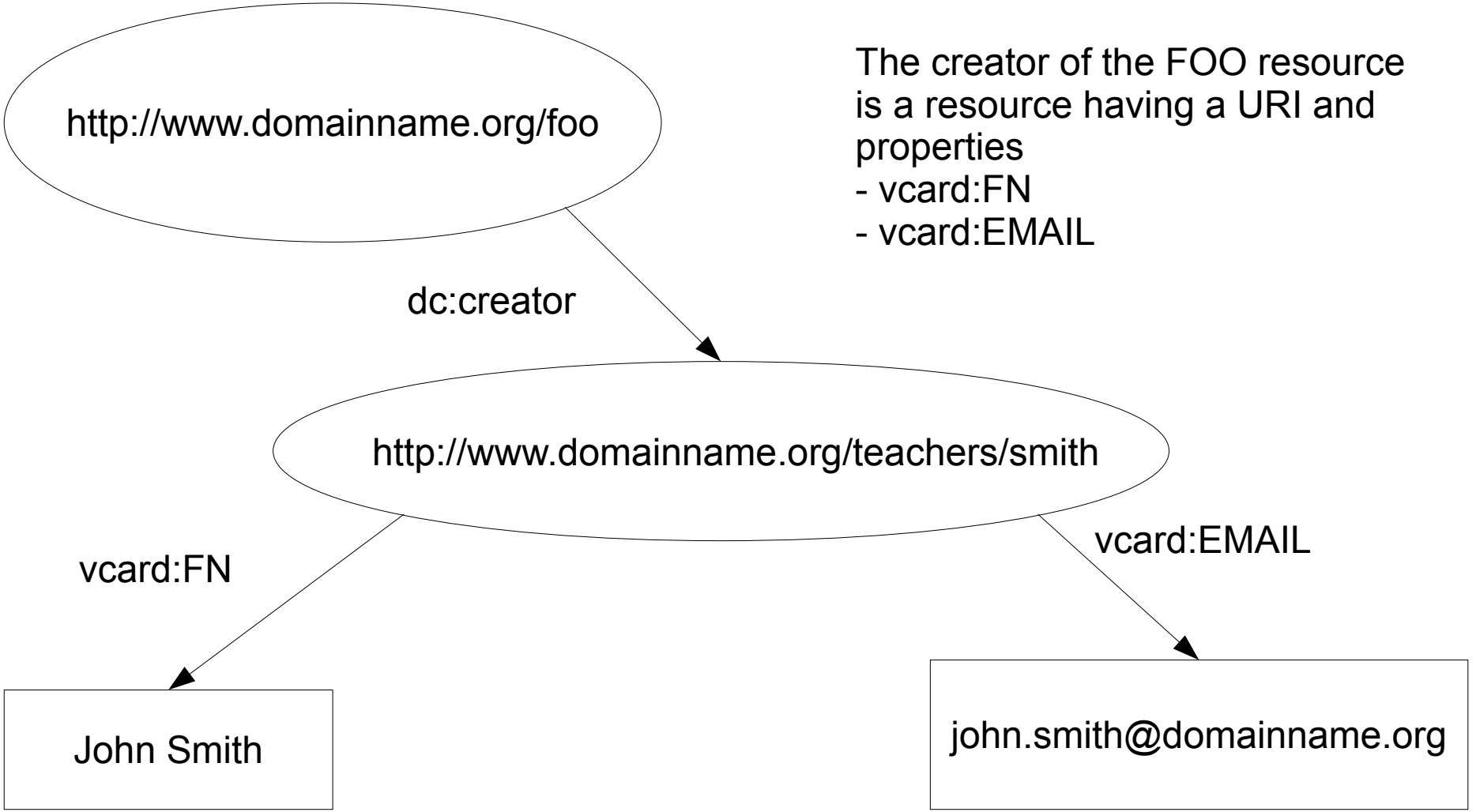
The creator of the FOO resource
is a resource described by
- vcard:FN
- vcard:EMAIL



Nesting Resources
(Property value as Description element)

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:vcard="http://imc.org/vCard/3.0#">

  <rdf:Description rdf:about="http://www.domainname.org/foo">

    <dc:creator>

      <rdf:Description>
        <vcard:FN>John Smith</vcard:FN>
        <vcard:EMAIL>john.smith@domainname.org</vcard:EMAIL>
      </rdf:Description>


    </dc:creator>

  </rdf:Description>
</rdf:RDF>
```

Nesting Resources (Property value as URI)



Nesting Resources (Property value as URI)

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:vcard="http://imc.org/vCard/3.0#">

  <rdf:Description rdf:about="http://www.domainname.org/foo">
      <dc:creator rdf:resource="http://www.domainname.org/teachers/smith"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://www.domainname.org/teachers/smith">
      <vcard:FN>John Smith</vcard:FN>
      <vcard:EMAIL>john.smith@domainname.org</vcard:EMAIL>
  </rdf:Description>

</rdf:RDF>
```

Using Abbreviated RDF Syntax (1)

- Property elements converted to attributes

```
<?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/">


<!--ALTERNATIVE 1-->

<rdf:Description rdf:about="http://www.domainname.org/foo">
      <dc:creator>John Smith</dc:creator>
      <dc:title>This is my book</dc:title>
      <dc:date>2001-04-10</dc:date>
 </rdf:Description>



<!--ALTERNATIVE 2-->
  <rdf:Description rdf:about="http://www.domainname.org/foo"
        dc:creator="John Smith" dc:title="This is my book" dc:date="2001-04-10"/>


</rdf:RDF>
```

Using Abbreviated RDF Syntax (2)

- Inner Description and related properties as property attributes

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:vcard="http://imc.org/vCard/3.0#">

<!-- ALTERNATIVE 1-->
  <rdf:Description rdf:about="http://www.domainname.org/foo">

    <dc:creator>
      <rdf:Description rdf:about="http://www.domainname.org/teacher/smith">
        <vcard:FN>John Smith</vcard:FN>
        <vcard:EMAIL>john.smith@domainname.org</vcard:EMAIL>
      </rdf:Description>
    </dc:creator>

  </rdf:Description>

<!-- ALTERNATIVE 2-->
  <rdf:Description rdf:about="http://www.domainname.org/foo">

    <dc:creator rdf:resource="http://www.domainname.org/teacher/smith"
          vcard:FN="John Smith"
          vcard:EMAIL="john.smith@domainname.org"/>

  </rdf:Description>

</rdf:RDF>
```

Using Abbreviated RDF Syntax (3)

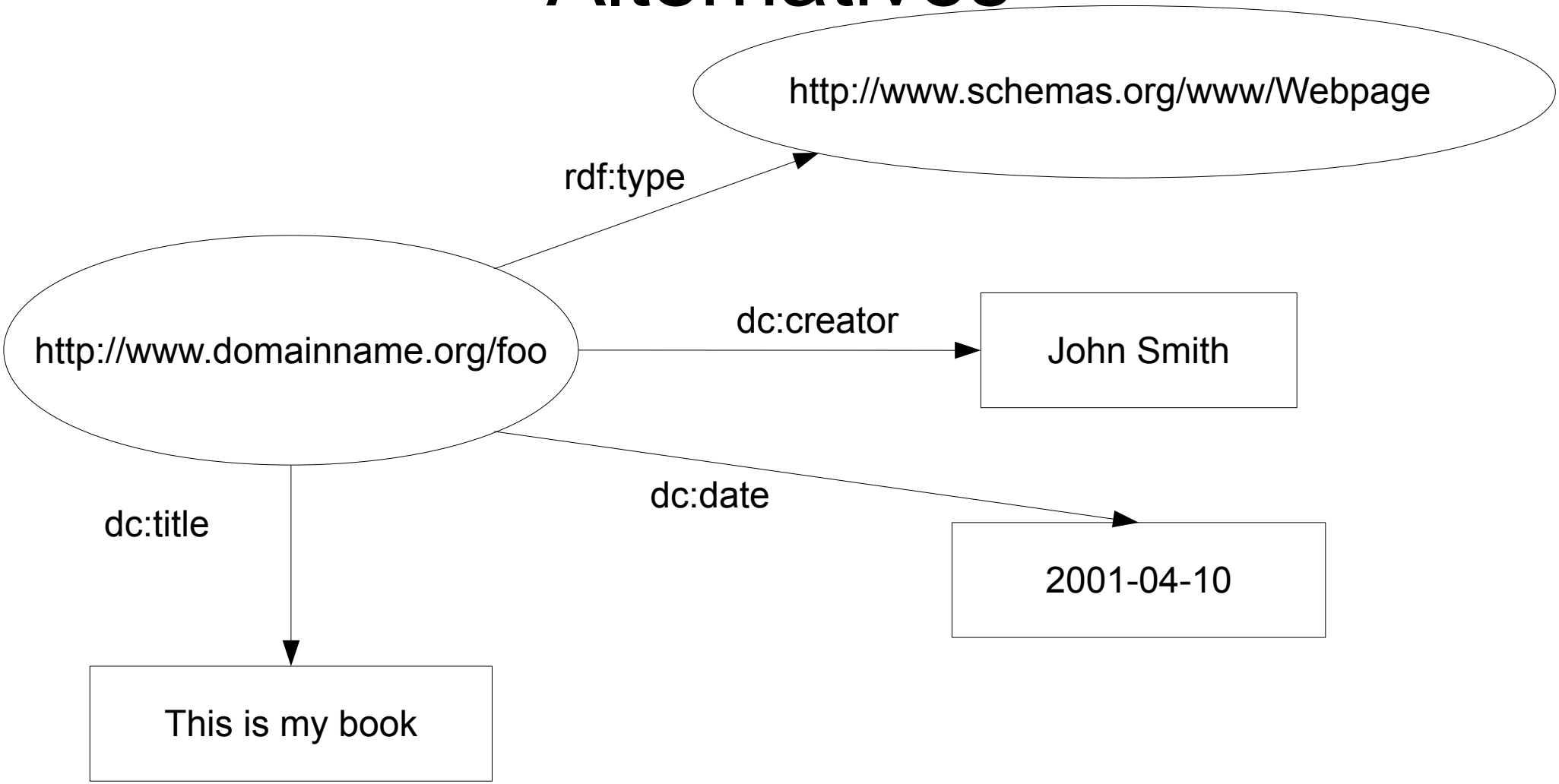
- **New fact:** http://www.domainname.org/foo is an instance of the class WebPage
- **rdf:type** element adds this new fact

```
<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://www.domainname.org/foo">
   <rdf:type rdf:resource="http://www.schemas.org/www/Webpage"/>
   <dc:creator>John Smith</dc:creator>
   <dc:title>This is my book</dc:title>
   <dc:date>2001-04-10</dc:date>
 </rdf:Description>
</rdf:RDF>
```

- Value of **rdf:type** attribute (http://www.schemas.org/www/Webpage) can be used directly as element name <s:Webpage> (“s:” is the namespace)

```
<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://www.schemas.org/www/">
   <s:WebPage rdf:about="http://www.domainname.org/foo/">
    <dc:creator>John Smith</dc:creator>
    <dc:title>This is my book</dc:title>
    <dc:date>2001-04-10</dc:date>
  </s:WebPage>
</rdf:RDF>
```

The Same Graph for the 2 Previous Alternatives



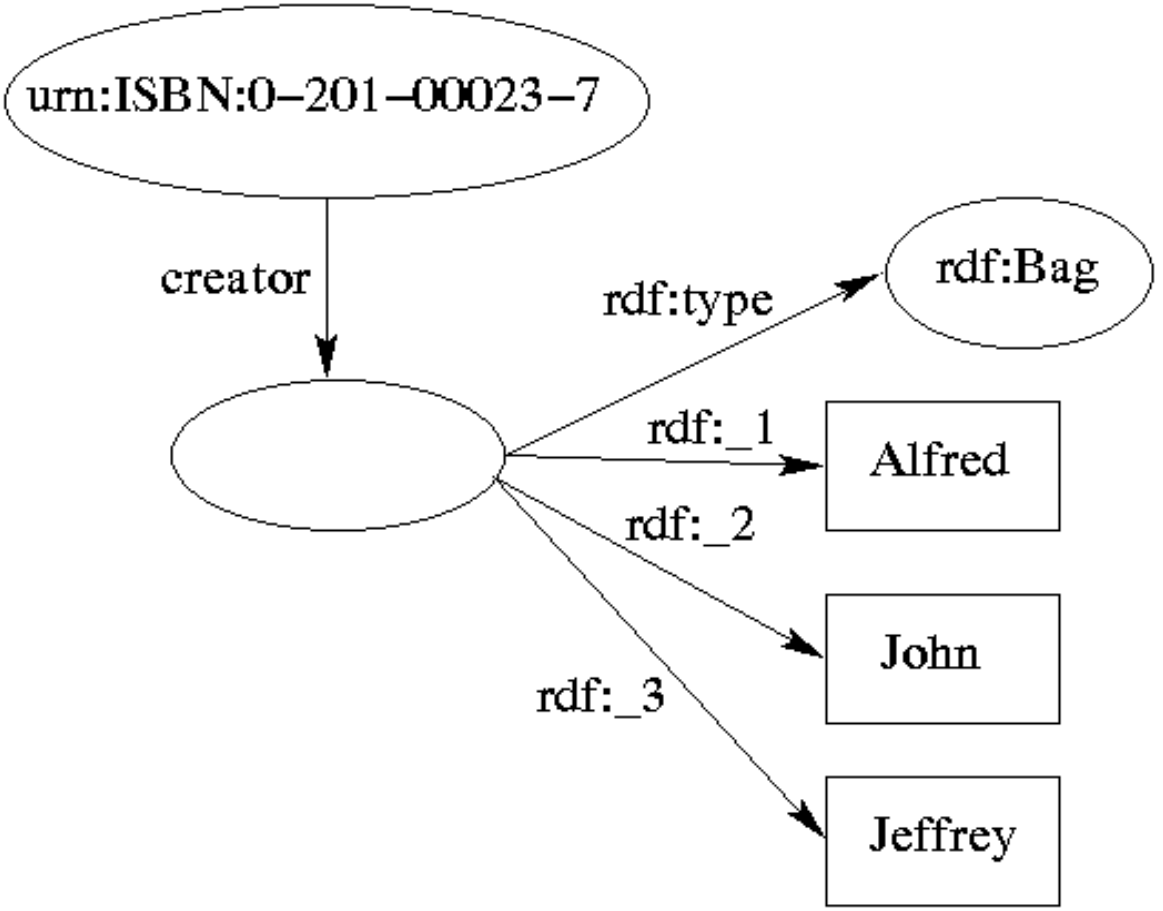
RDF Containers

- **Bag** - An **unordered** list of resources or literals
- **Sequence** - An **ordered** list of resources or literals
- **Alternative** - A list of resources or literals that represent alternatives for the value of a property

Using the Bag Container

- Statement:

  “The authors of the book 0201000237 are Alfred, John and Jeffrey”



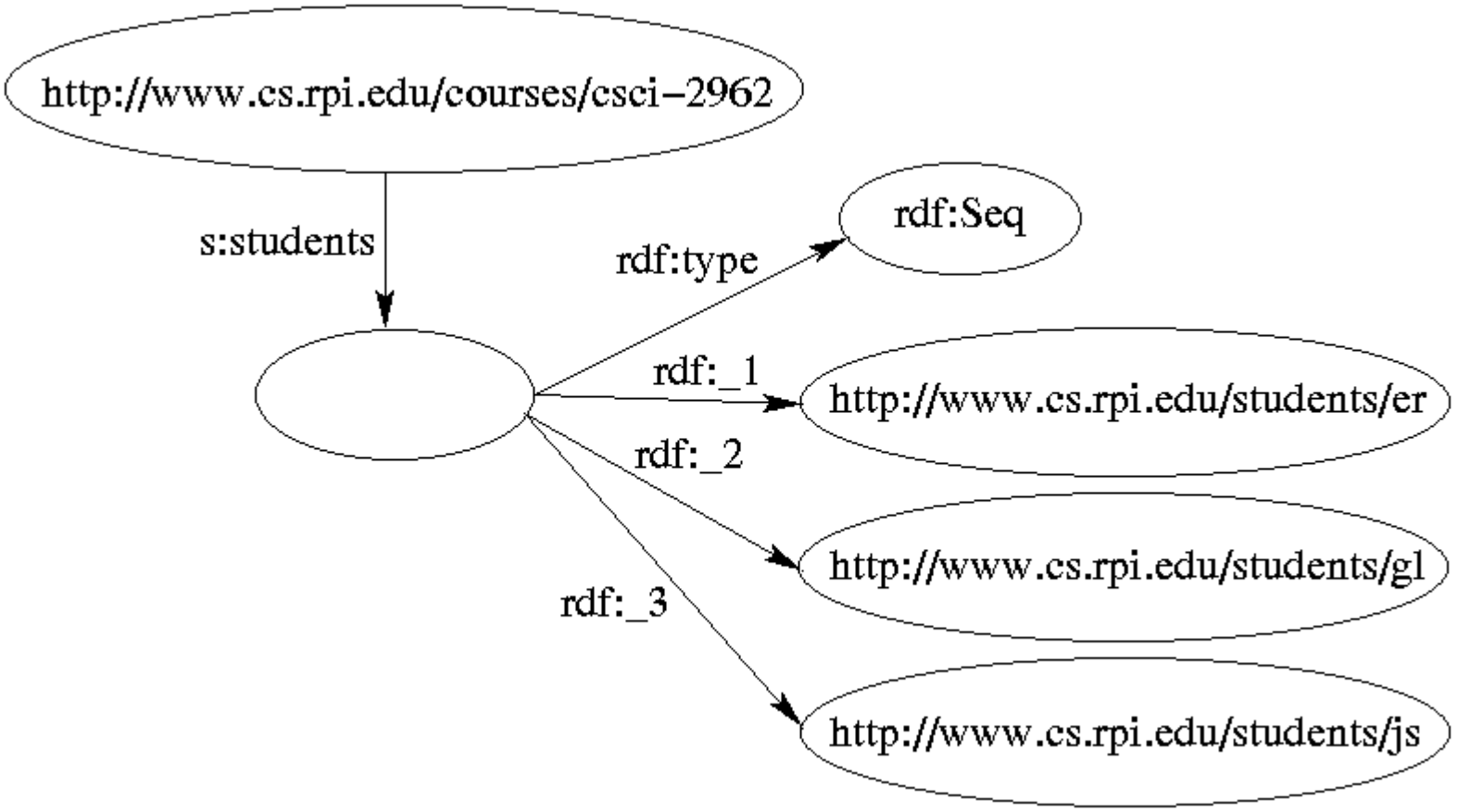
Using the Bag Container

```
<?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/">

    <rdf:Description rdf:about="urn:ISBN:0-201-00023-7">

     <dc:creator>
         <rdf:Bag>
                <rdf:li>Alfred</rdf:li>
                <rdf:li>John</rdf:li>
                <rdf:li>Jeffrey</rdf:li>
         </rdf:Bag>
      </dc:creator>

    </rdf:Description>

 </rdf:RDF>
```

Using the Seq Container

- Statement:

  “The students of the course csci-2962 in alphabetical order are Elizabeth, George and John.”



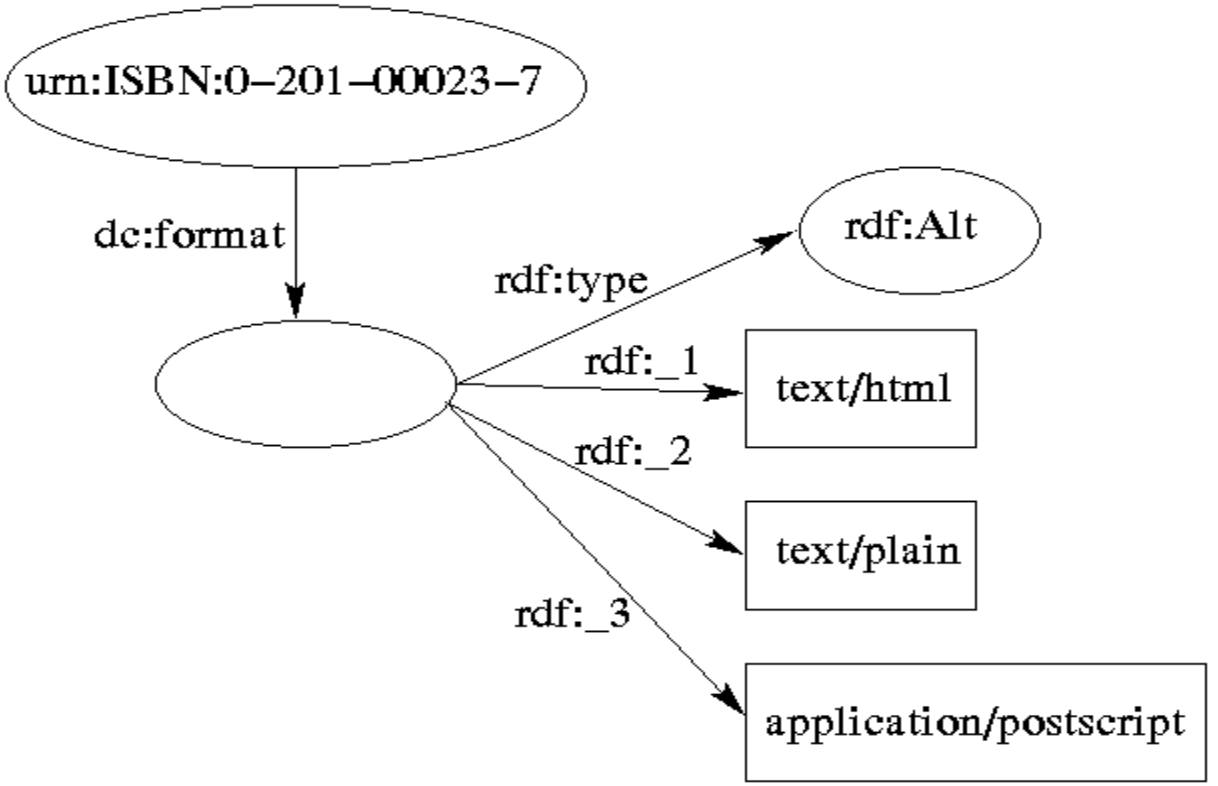
Using the Seq Container

```
<?xml version="1.0"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:s="http://www.schemas.org/Course/">

   <rdf:Description rdf:about="http://www.cs.rpi.edu/courses/csci-2962">
     <s:students>
        <rdf:Seq>
              <rdf:li rdf:resource="http://www.cs.rpi.edu/students/er"/>
              <rdf:li rdf:resource="http://www.cs.rpi.edu/students/gl"/>
              <rdf:li rdf:resource="http://www.cs.rpi.edu/students/js"/>
        </rdf:Seq>
     </s:students>
    </rdf:Description>

  </rdf:RDF>
```

Using the Alt Container

- Statement:

  “The format of the book 0-201-00023-7 can be plain text or html or postscript.”



Using the Alt Container

```
<?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/">

    <rdf:Description rdf:about="urn:ISBN:0-201-00023-7">
      <dc:format>
        <rdf:Alt>
            <rdf:li>text/html</rdf:li>
            <rdf:li>text/plain</rdf:li>
            <rdf:li>application/postscript</rdf:li>
        </rdf:Alt>
      </dc:format>
    </rdf:Description>

  </rdf:RDF>
```

Making Statements about Containers

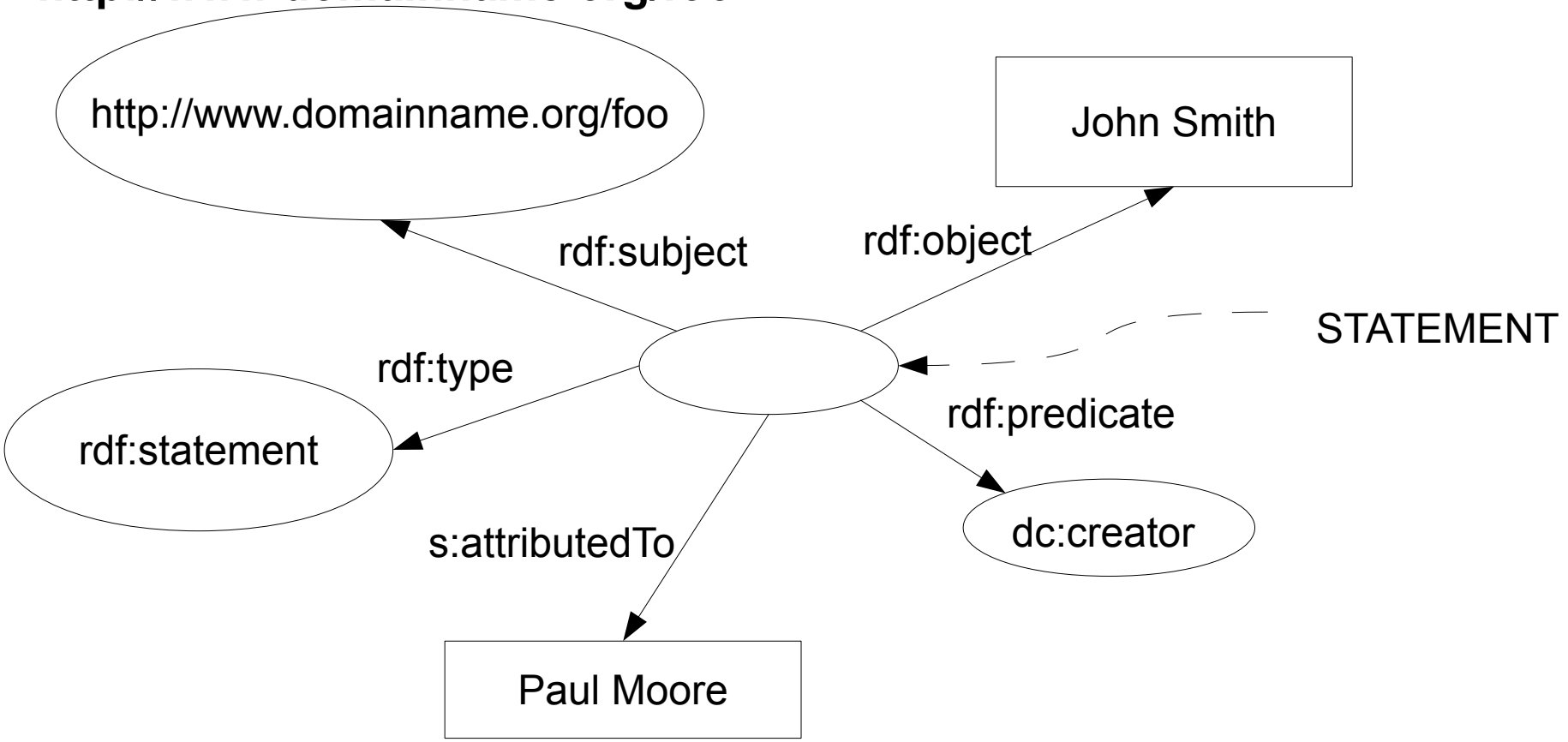
```
<?xml version="1.0"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:s="http://www.schemas.org/Course/"
   xmlns:dc="http://purl.org/dc/elements/1.1/">


    <rdf:Description rdf:about="http://www.domainname.org/foo">
      <s:homeworks>
         <rdf:Bag ID="pages">
               <rdf:li rdf:resource="http://www.domainname.org/foo/hw1.html">
               <rdf:li rdf:resource="http://www.domainname.org/foo/hw2.html">
         </rdf:Bag>
      </s:homeworks>
    </rdf:Description>

    <rdf:Description rdf:about="#pages">
       <dc:creator>John Smith</dc:creator>
    </rdf:Description>

  </rdf:RDF>
```

Making Statements about the Items in a Container

```
<?xml version="1.0"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:s="http://www.schemas.org/Course/"
   xmlns:dc="http://purl.org/dc/elements/1.1/">

    <rdf:Description rdf:about="http://domainname.org/foo">
      <s:homeworks>
          <rdf:Bag ID="pages">
              <rdf:li rdf:resource="http://www.domainname.org/foo/hw1.html"/>
              <rdf:li rdf:resource="http://www.domainname.org/foo/hw2.html"/>
          </rdf:Bag>
      </s:homeworks>
    </rdf:Description>

    <rdf:Description rdf:aboutEach="#pages">
       <dc:creator>John Smith</dc:creator>
    </rdf:Description>

  </rdf:RDF>
```

Statements about Statements: Using Reification

- A statement can be assigned a URI and treated as a resource
- RDF defines the following properties:
  - rdf:subject - The resource being described
  - rdf:predicate - The original property
  - rdf:object - The property value in the statement
  - rdf:type – rdf:Statement
- "**Paul Moore** says that **John Smith** is the **creator** of the web page **http://www.domainname.org/foo"**

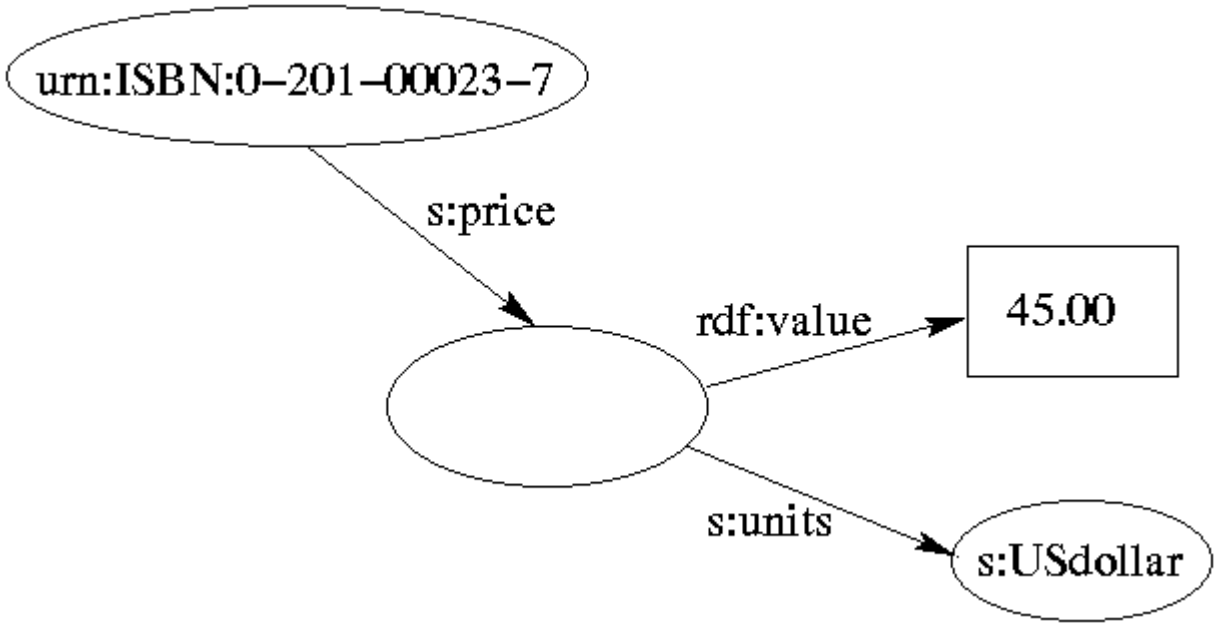


Statements about Statements: Using Reification

```
<?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://www.schemas.org/schema/">

   <rdf:Description>


     <rdf:subject rdf:resource="http://www.domainname.org/foo"/>
     <rdf:predicate rdf:resource="http://purl.org/dc/elements/1.1/creator"/>
     <rdf:object>John Smith</rdf:object>

                          <!-- The following means that the resource is of type “Statement” -->

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


     <s:attributedTo>Paul Moore</s:attributedTo>

    </rdf:Description>
  </rdf:RDF>
```

Non-Binary Relations

- RDF data model only supports binary relations (relation between two resources)
- Solution: Use an intermediate resource with additional properties
- Ex. statement: “The cost of the book 0-201-00023-7 is $45.00”



Non-Binary Relations

```
<?xml version="1.0"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:s="http://www.schemas.org/Units/">

    <rdf:Description rdf:about="urn:ISBN:0-201-00023-7">

      <s:price rdf:parseType="Resource"> <!--Blank node (not RDF URI reference nodes)-->
        <rdf:value>45.00</rdf:value>
        <s:units rdf:resource="http://www.schemas.org/Units/USdollar"/>
      </s:price>

    </rdf:Description>

  </rdf:RDF>
```

RDF and RDFS

- RDF needs a way to define application-specific classes and properties.
- Application-specific classes and properties must be defined using extensions to RDF.
- One such extension is RDF Schema (RDFS)
- RDF Schema does not provide actual application-specific classes and properties.
- RDF Schema gives you a way (framework) to define application-specific classes and properties.
- Classes in RDF Schema is much like classes in object oriented programming languages.
    - Resources as instances of classes, and subclasses of classes.

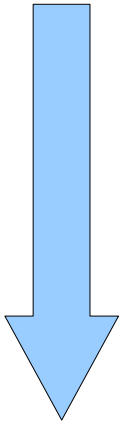
RDF Schema

- RDFS defines Classes and Properties for application-specific semantic model (to be applied to documents using RDF)

- As
  XML Schema gives specific constraints on the structure of an XML document (you can define your application-specific document structure)

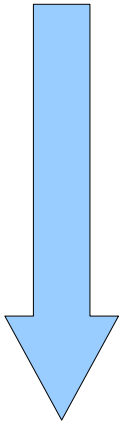
  Similarly
  RDF Schema gives the “meaning” of the RDF statements (you can define your application-specific classes and properties)

XMLSchema



XML

RDFSchema



RDF

RDFS Classes and Properties

- Namespace prefix:'rdfs'

  'http://www.w3.org/2000/01/rdf-schema#'

- RDFS
  - Defines Classes which the Resources (to be described in RDF) belong to
  - Defines Properties, and in particular
    - the possible Classes which a Property applies to (*rdfs:domain*)
    - the Class of property values (*rdfs:range*)

RDFS Core Classes

- rdfs:Resource
  - resources are instances of this class
- rdf:Property
  - properties are instances of this class
- rdfs:Class
  - similar to a Class in object-oriented programming language

RDF Core Properties

- rdf:type
  - indicates that a resource is a member of a class
- rdfs:subClassOf
  - specifies subset/superset relation between classes
- rdfs:subPropertyOf
  - specifies that a property is a specialization of another
- rdfs:seeAlso
  - specifies a resource that provides additional information
- rdfs:isDefinedBy
  - indicates the resource defining the subject resource

RDF Constraints

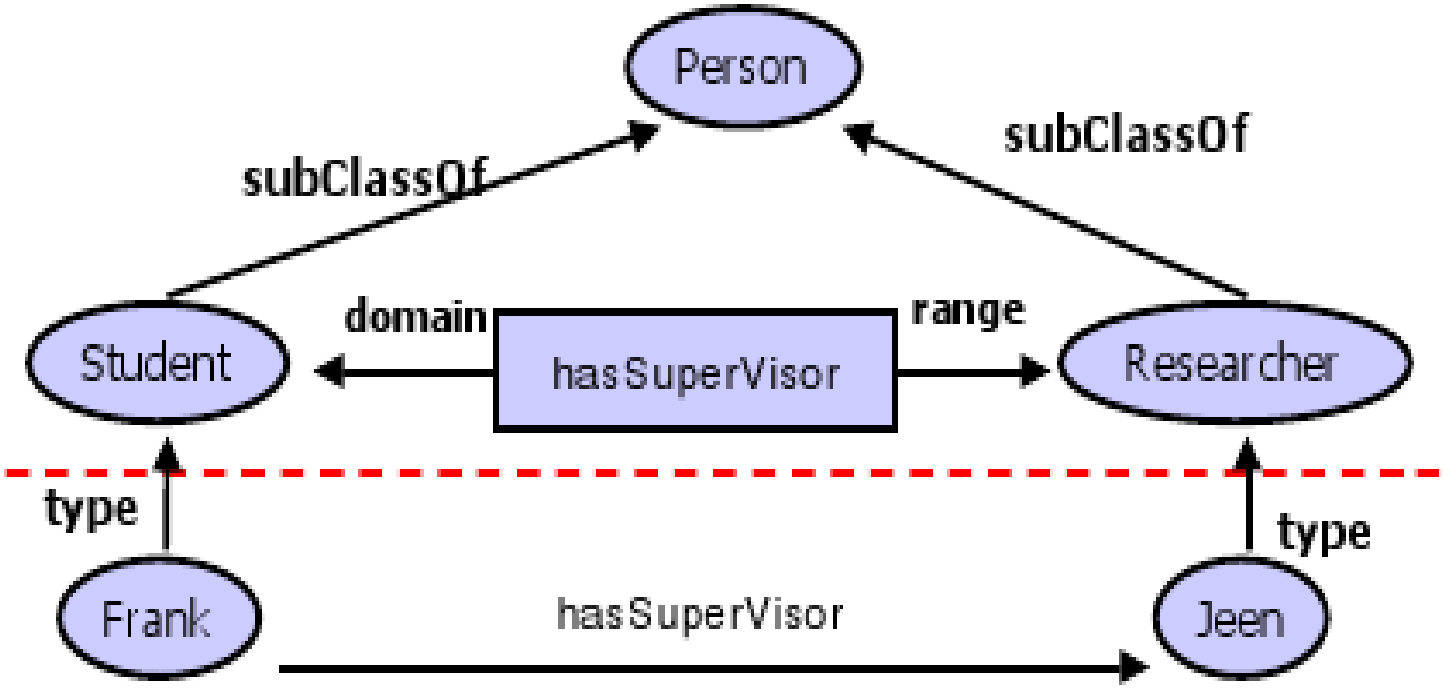
- rdfs:domain
  - Indicates the possible Classes which a property applies to
  - No domain property means that the property may be used with any resource
- rdfs:range
  - Indicates the classes of property values

RDF Documentation

- rdfs:comment
  - human-readable description of a resource
- rdfs:label
  - human-readable version of a resource name

RDF-RDFS Example

RDFS



RDF

RDFS Serialization

```
<?xml version="1.0"?>

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

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

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

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

</rdf:RDF>
```

RDF Abbreviated Example

- A class (ex. “Person”) is a Resource of type “Class”
- So we can use <rdfs:Class> instead of <rdf:Description> and <rdf:type>
-

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

can be written as:

```
<rdfs:Class rdf:ID="Person" />
```

-

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

can be written as:

```
<rdfs:Class rdf:ID="Student">
    <rdfs:subClassOf rdf:resource="#Person"/>
</rdfs:Class>
```

RDFS Example (Classes)

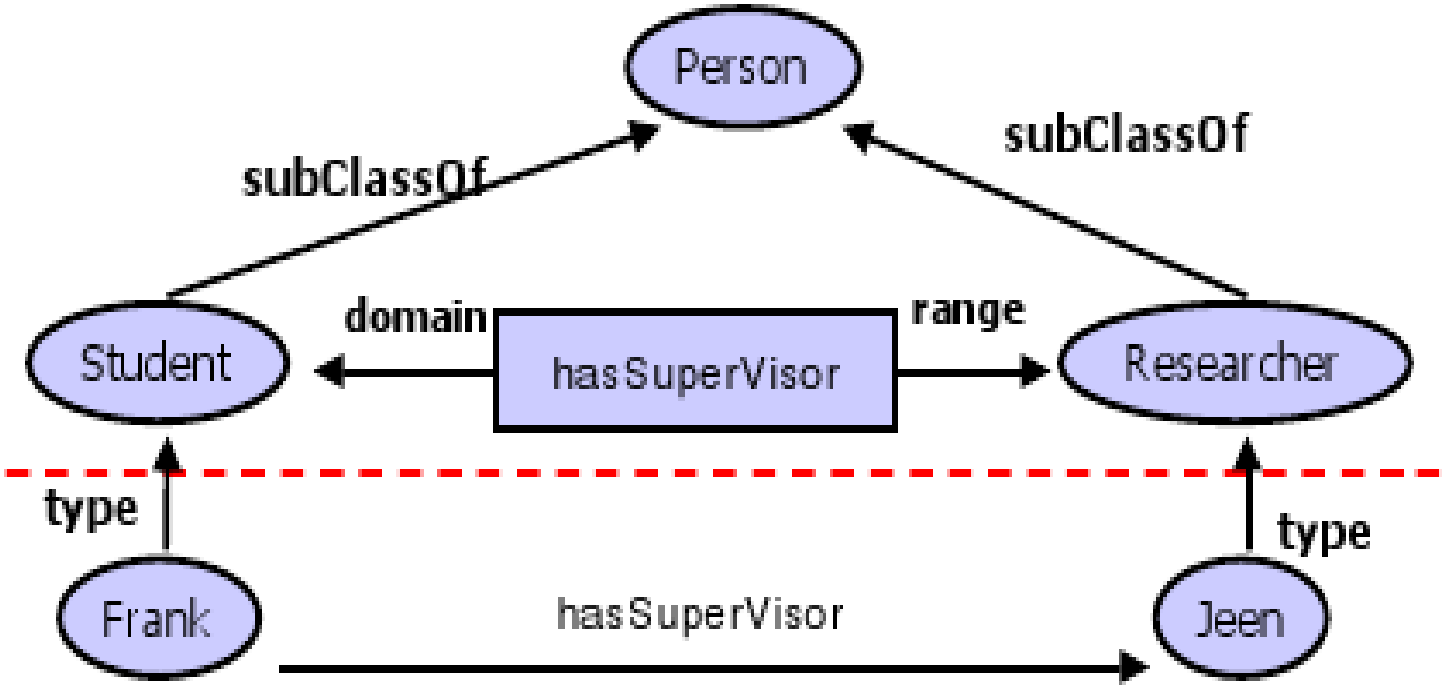
```
<?xml version="1.0"?>

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

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

<rdfs:Class rdf:ID="Student">
        <rdfs:subClassOf rdf:resource="#Person">
</rdfs:Class>


<rdfs:Class rdf:ID="Researcher">
        <rdfs:subClassOf rdf:resource="#Person">
</rdfs:Class>


...
```

RDFS Example (Properties)

```
...
<rdf:Property rdf:ID="hasSuperVisor">
    <rdfs:comment>   Definition of hasSuperVisor property   </rdfs:comment>
    <rdfs:domain rdf:resource="#Student"/>
    <rdfs:range  rdf:resource="#Researcher"/>
</rdf:Property>
</rdf:RDF>
```

RDF-RDFS Example

RDFS



RDF

RDF Instance (1)

```
<?xml version="1.0"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns:uni="http://www.university.org#">

    <rdf:Description rdf:about="http://www.domainname.org#Frank">

     <rdf:type rdf:resource="http://www.university.org#Student"/>

      <uni:hasSuperVisor>

          <rdf:Description rdf:about=”http://www.domainname.org#Jeen”>
                <rdf:type rdf:resource="http://www.university.org#Researcher"/>
          </rdf:Description>

       </uni:hasSuperVisor>

   </rdf:Description>

</rdf:RDF>
```

RDF Instance (2)

```
<?xml version="1.0"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns:uni="http://www.university.org#">


    <rdf:Description rdf:about="http://www.domainname.org#Frank">
            <rdf:type rdf:resource="http://www.university.org#Student"/>
            <uni:hasSuperVisor rdf:resource="http://www.domainname.org#Jeen"/>
     </rdf:Description>

    <rdf:Description rdf:about="http://www.domainname.org#Jeen”>
            <rdf:type rdf:resource="http://www.university.org#Researcher"/>
    </rdf:Description>

</rdf:RDF>
```

RDF Instance (3)

```
<?xml version="1.0"?>
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns:uni="http://www.university.org#">


    <uni:Student rdf:about="http://www.domainname.org#Frank">

        <uni:hasSuperVisor>
            <uni:Reseacher rdf:about="http://www.domainname.org#Jeen"/>
        </uni:hasSuperVisor>

     </uni:Student>

</rdf:RDF>
```

The same Graph for (1)(2)(3)

Software for Managing Semantic Web standards

- **Jena**: an open source Java framework for storing, querying and reasoning with RDF and RDF Schema
  - A RDF API
  - Reading and writing RDF in RDF/XML, N3 and N-Triples
  - An OWL API
  - In-memory and persistent storage
  - SPARQL query engine
- **Sesame**
- Reasoners
  - Pellet
  - Racer

A Semantic Model for Legislation

Overview

- From Structure to Semantics of Legislative texts

- A Semantic Model for Legislation
  - *Model of Provisions* (analytical metadata)
  - Strategies and tools to implement Analytical Semantic Mark-up

From Structural to Semantic Mark-up of Legislative Texts

- Legal documents (texts) show peculiarities with respect to other documents in terms of their fruition
- Legal Information Retrieval on legislative collections is mainly focused to identify rules rather than law texts
- Semantic mark-up is needed to obtain this result

From Structural to Semantic Mark-up of Legislative Texts

- The semantics of texts can be perceived mainly according to 3 meaningful levels:
  - Bottom level:
    - atomic components (simple and complex terms)
  - Middle level:
    - aggregations of such components (sentences or partitions)
  - Top level:
    - whole text

From Structural to Semantic Mark-up of Legislative Texts

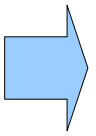
- The semantics of legislative texts can be described according to 3 meaningful levels:
  - Bottom level
    - terms
  - Middle level
    - provisions
  - Top level
    - acts

How to describe the Semantics of Legislative Texts

- Terms (Bottom level)
  - Controlled vocabularies, Thesauri, Domain Ontologies;
- Provisions (Middle level)
  - Metadata model to describe fragment of legislative texts

Law as a set of Partitions

(Structure)



Law as a set of Provisions

(Semantics)

- Acts (Top level)
  - General metadata of a whole act

Semantic models (Legal Concepts) in Literature

- *"The entire body of laws and regulations may be seen as a set of provisions, carried by speech acts, namely sentences endowed with meaning"* [Raz, 1977]
- Models of legal concepts in literature
  - Hohfeld [Hohfeld, 1913a], [Hohfeld, 1913b]
  - Rawls [Rawls, 1955]
  - Hart [Hart, 1961]
  - Ross [Ross, 1968]
  - Bentham [Bentham, 1872]
  - Kelsen [Kelsen, 1991]
  - Biagioli's "Model of Provisions" [Biagioli, 1997]

Biagioli's model

- Provision types and related Attributes
- Provision types: taxonomy containing
  - Rules (deontic concepts (duty, permission, prohibition, etc.), penalties, right, etc.)
  - Amendments (insertion, abrogation, substitution, etc.)
- Attributes
  - Entities which provisions apply to

Rules

- Rules are provisions which regulate a reality considered by the act.

- Adopting a typical law theory distinction, well expressed by Rawls, they consist in:
  - **constitutive rules**: rules on entities
    - Of the game (Ricciardi): they introduce entities
    - In the game (Ricciardi): they assign a juridical profile to the entities ("empowering norms");
  - **regulative rules**
    - Rules on Actions (deontic concepts)
    - Remedies: they discipline the substantial and procedural defaults

Provision Types and Arguments

- Rules and Amendments include several Sub-classes or *Provision Types*

- Each Provision Type is described by a set of specific *Attributes*

Constitutive Provisions

Class	Sub-class	Attributes			
definition	term	definiendum	definiens		
	procedure	addressee	counterpart	action	object
creation	establishment	addressee			
	organization	addressee			
attribution	power	addressee	counterpart	activity	object
	liability	addressee	counterpart	activity	object
	status	addressee			object

(Biagioli and Grossi, Jurix 2008)

Regulative Provisions

Class	Sub-class	Attributes			
action	right	bearer	counterpart	action	object
	duty	bearer	counterpart	action	object
	prohibition	bearer	counterpart	action	object
	permission	bearer	counterpart	action	object
remedy	redress	bearer	counterpart	effect	action
	violation	bearer	counterpart	penalty	action

(Biagioli and Grossi, Jurix 2008)

Amendments

- “Rules on rules”
- Technical provisions which affect other provisions
- They work over the legal order providing changes

Amendments

Class	Sub-Class		Arguments	
Contents amendments	Repeal	Norm		
	Substitution	Norm	NewText	OldText
	Insertion	Norm	NewText	
Temporal amendments	Suspension	Norm	Date	
	Transitional nature	Norm	Date	
	Prorogation	Norm	Date	
	Revival	Norm	Date	
	Retroactivity	Norm	Date	
	Ultra-activity	Norm	Date	
Extension amendments	Extension	Norm	Case	
	Derogation	Norm	Case	

Stuctural and Semantic mark-up

- A legislative text can be percieved according to
  - a Structural view
    - Law as a set of Partitions
  - a Semantic view
    - Law as a set of Provisions
- Text fragments are, at the same time, paragraphs and provisions, according to whether they are seen from a structural or semantic view-point.

A fragment of a law text perceived according to a structural or semantic view

Article 7

(Notification)

1. *A controller intending to process personal data falling within the scope of application of this Act shall have to notify the Garante thereof…*

2. *The notification shall have to be given…*

Structural view

Division **(or *Paragraph*)**

Semantic view

***Type of provision*:**
"Duty"

Attributes:

Bearer:
"Controller"

Object:
"Process personal data"

Action:
"Notification"

Counterpart:
"Garante"

A fragment of a law text perceived according to the structural or semantic view

1. In article 46, paragraph 1, of the legislative decree 30 July 1999, n. 300, letters a) and b) are abrogated.

Structural view

Division **(or *Paragraph*)**

Semantic view

***Type of provision*:**
"Repeal"

Attributes:

Norm to be repealed:
legislative decree 30 July 1999, n. 300

Partition to be repealed:
article 46, paragraph 1, letter a)
article 46, paragraph 1, letter b)

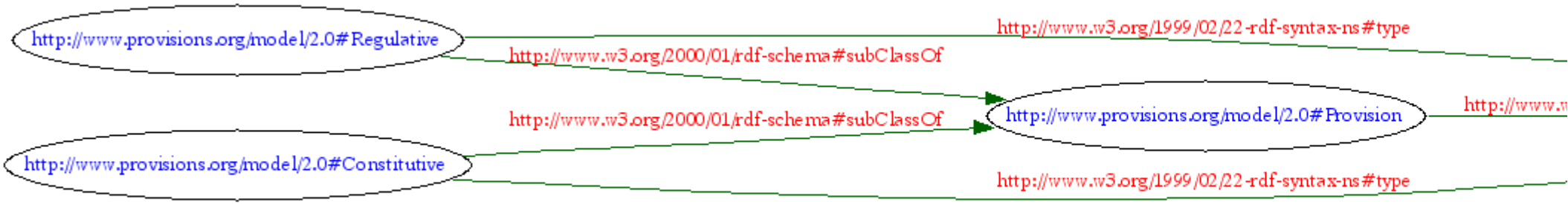
Use of the Provision Model

- Semantic Mark-up
- Metadata-based Legal Information Retrieval
  - to query a legal information system according to a semantic point of view;
- Consolidation (amendments)

Provision Model in RDFS (Top Classes)

```
<?xml version="1.0"?>
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xml:base="http://www.provisions.org/model/2.0">


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



<rdfs:Class rdf:ID="Regulative">
   <rdfs:subClassOf rdf:resource="#Provision"/>
</rdfs:Class>



<rdfs:Class rdf:ID="Constitutive">
   <rdfs:subClassOf rdf:resource="#Provision"/>
</rdfs:Class>


.............

</rdf:RDF>
```

Graph of Provision Model (Top Classes) (1)



Graph of Provision Model (Top Classes) (2)

Constitutives Provisions types in RDFS

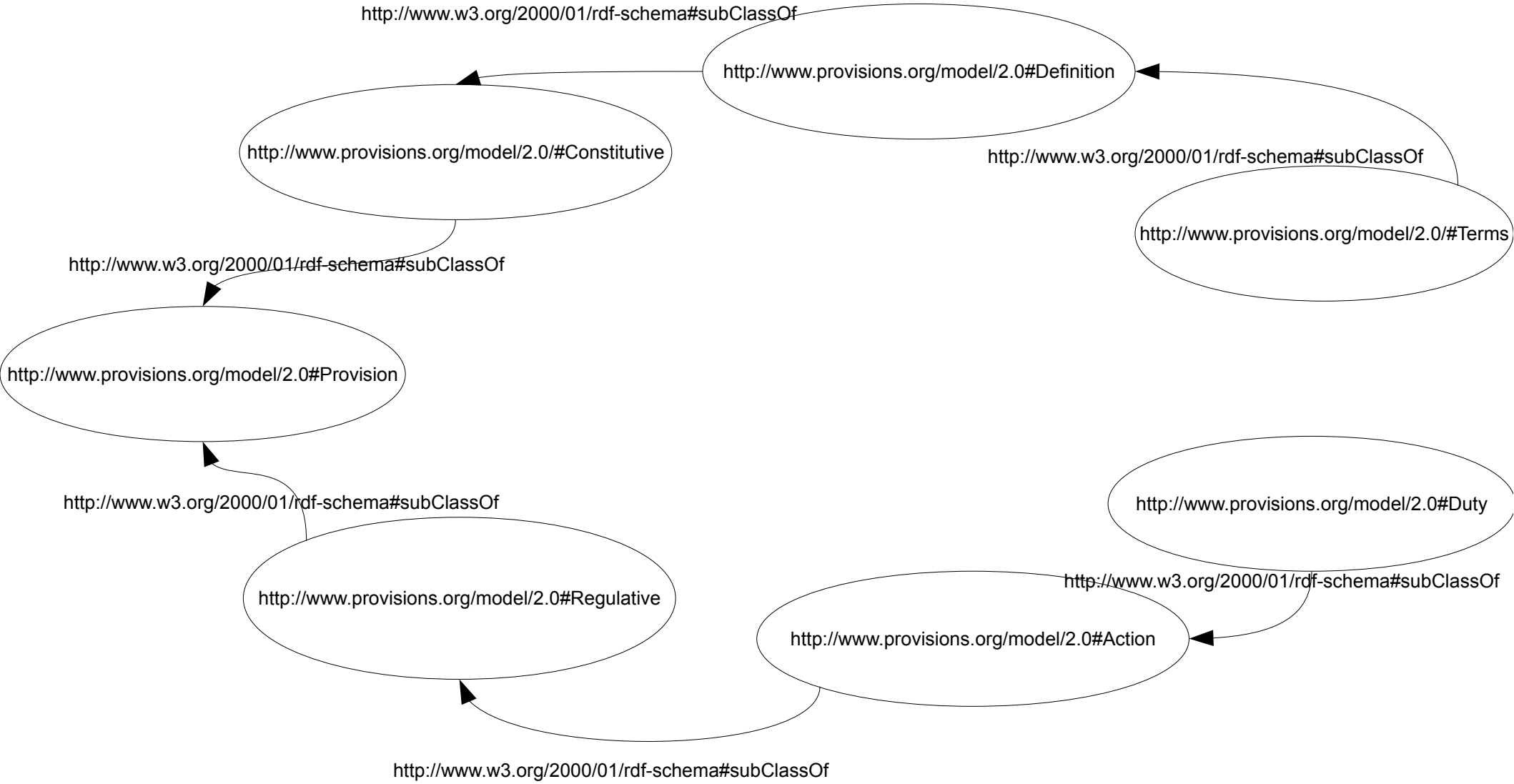
```
...
<rdfs:Class rdf:ID="Definition">
   <rdfs:subClassOf rdf:resource="#Constitutive"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Creation">
   <rdfs:subClassOf rdf:resource="#Constitutive"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Attribution">
   <rdfs:subClassOf rdf:resource="#Constitutive"/>
</rdfs:Class>

<!-- Definition -->
<rdfs:Class rdf:ID="Term">
   <rdfs:subClassOf rdf:resource="#Definition"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Procedure">
   <rdfs:subClassOf rdf:resource="#Definition"/>
</rdfs:Class>

<!-- Creation -->
<rdfs:Class rdf:ID="Establishment">
   <rdfs:subClassOf rdf:resource="#Creation"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Organization">
   <rdfs:subClassOf rdf:resource="#Creation"/>
</rdfs:Class>

<!-- Attribution -->
<rdfs:Class rdf:ID="Power">
   <rdfs:subClassOf rdf:resource="#Attribution"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Liability">
   <rdfs:subClassOf rdf:resource="#Attribution"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Status">
   <rdfs:subClassOf rdf:resource="#Attribution"/>
</rdfs:Class>

...
```

Regulatives Provisions types in RDFS

```
...
<rdfs:Class rdf:ID="Action">
   <rdfs:subClassOf rdf:resource="#Regulative"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Remedy">
   <rdfs:subClassOf rdf:resource="#Regulative"/>
</rdfs:Class>

<!-- Action -->
<rdfs:Class rdf:ID="Right">
   <rdfs:subClassOf rdf:resource="#Action"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Duty">
   <rdfs:subClassOf rdf:resource="#Action"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Prohibition">
   <rdfs:subClassOf rdf:resource="#Action"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Permission">
   <rdfs:subClassOf rdf:resource="#Action"/>
</rdfs:Class>

...
```

```
<!-- Remedy -->
<rdfs:Class rdf:ID="Redress">
   <rdfs:subClassOf rdf:resource="#Remedy"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Violation">
   <rdfs:subClassOf rdf:resource="#Remedy"/>
</rdfs:Class>

...
```

Provisions Taxonomy (excerpt)



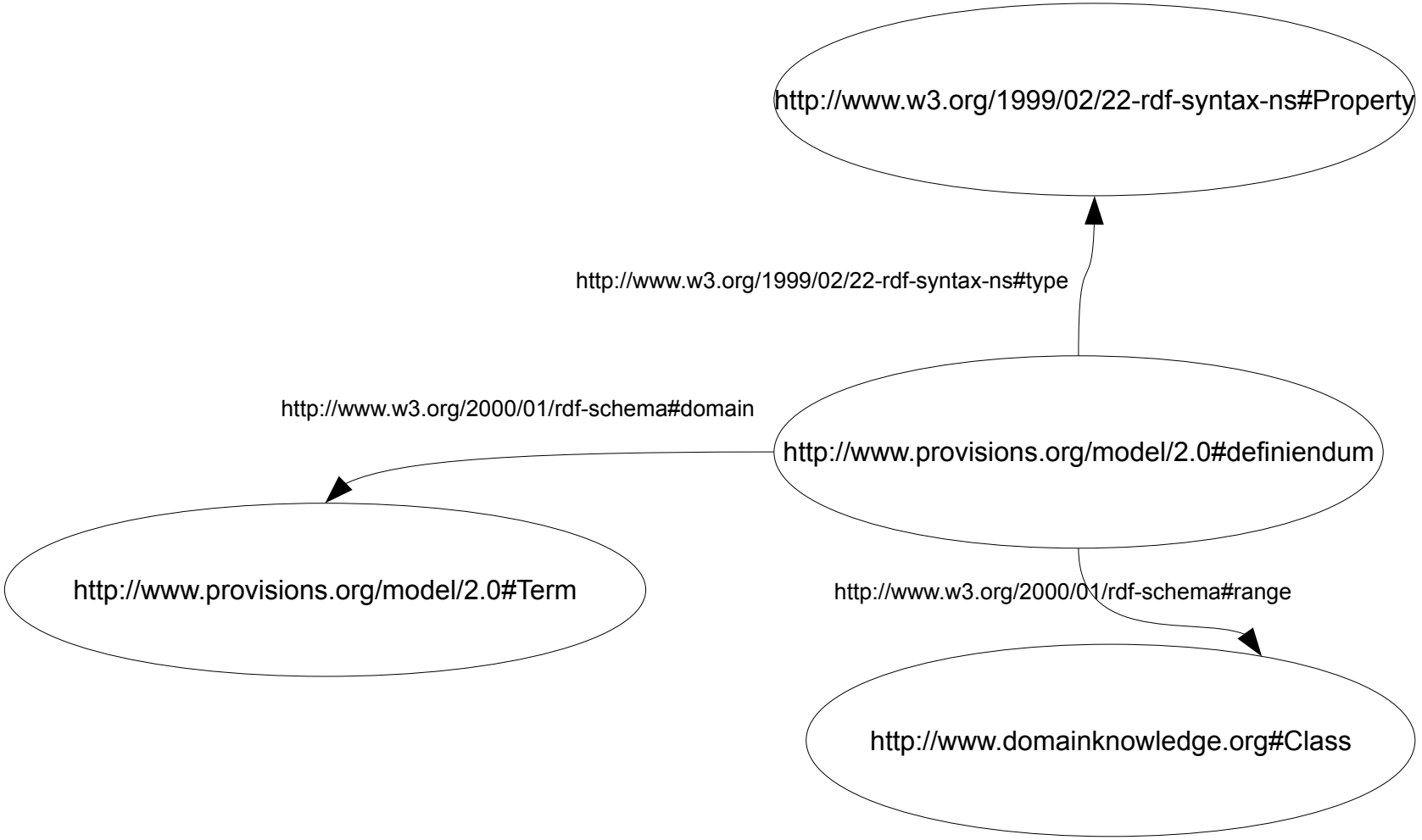
Provision Arguments in RDFS

Definition of Term
Definiendum: [Class]
Definiens: [Class]

...

```
<rdf:Property rdf:ID="definiendum">
    <rdfs:comment>What is defined by Definition of Terms</rdfs:comment>
    <rdfs:domain rdf:resource="#Term"/>
    <rdfs:range  rdf:resource="http://www.domainknowledge.org#Class"/>
</rdf:Property>


<rdf:Property rdf:ID="definiens">
    <rdfs:comment>The definition of a term</rdfs:comment>
    <rdfs:domain rdf:resource="#Term"/>
    <rdfs:range  rdf:resource="http://www.domainknowledge.org#Class"/>
</rdf:Property>
```

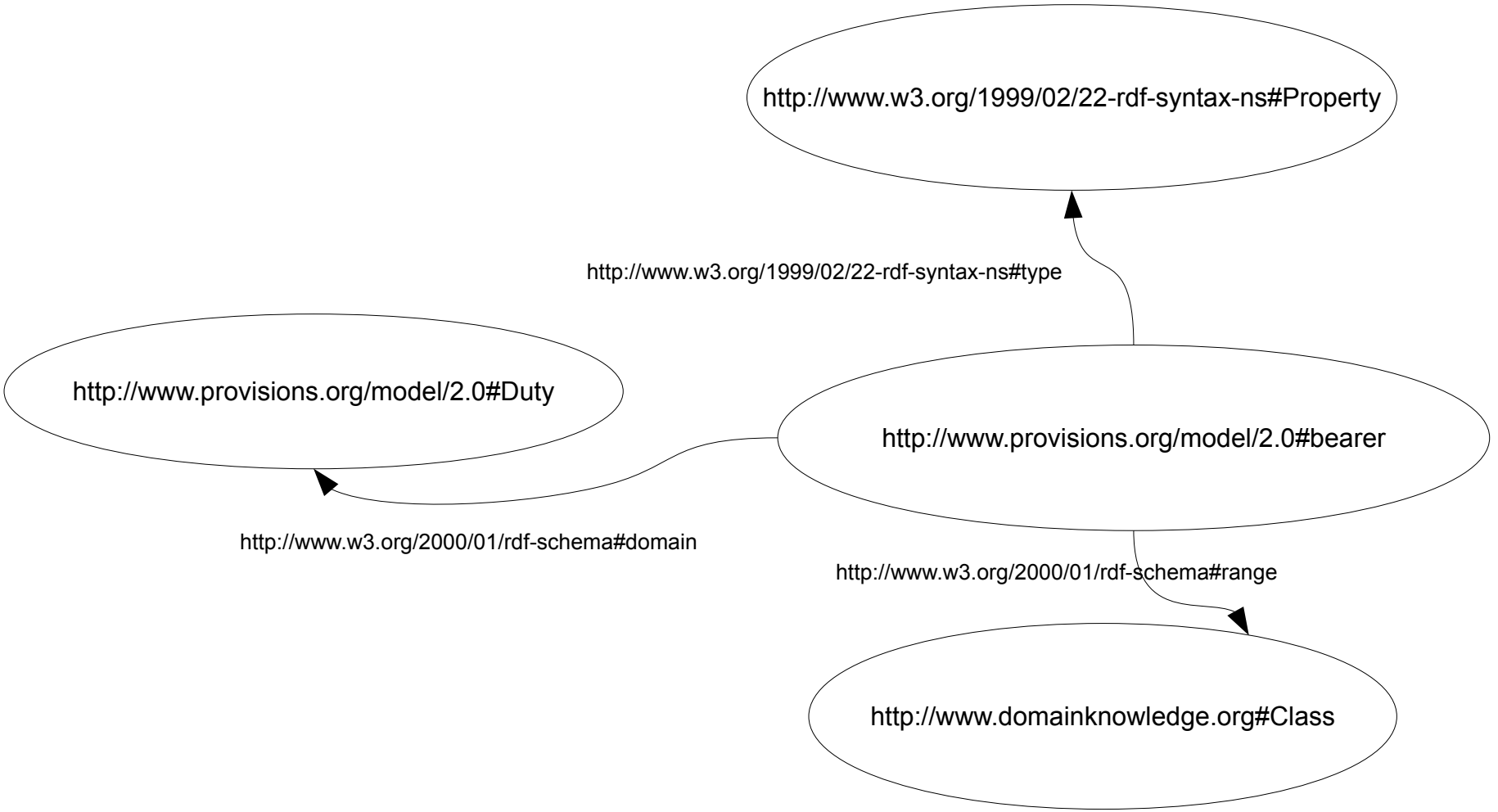
Provision Attributes as RDF Resource Property



Provision Arguments in RDFS

```
...

<rdf:Property rdf:ID="bearer">

    <rdfs:comment>Bearer of a provision type</rdfs:comment>

    <rdfs:domain rdf:resource="#Right"/>
    <rdfs:domain rdf:resource="#Duty"/>
    <rdfs:domain rdf:resource="#Prohibition"/>
    <rdfs:domain rdf:resource="#Permission"/>
    <rdfs:domain rdf:resource="#Redress"/>
    <rdfs:domain rdf:resource="#Violation"/>

    <rdfs:range rdf:resource="http://www.domainknowledge.org#Class"/>

</rdf:Property>

...
```

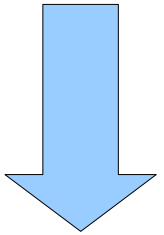
Provision Arguments as RDF Resource Property



Semantic Mark-up Example: a “Term” provision

2. For the purposes of this Act:

a) "data bank" shall mean any set of personal data, divided into one or more units located in one or more places, organized according to specific criteria such as to facilitate their processing



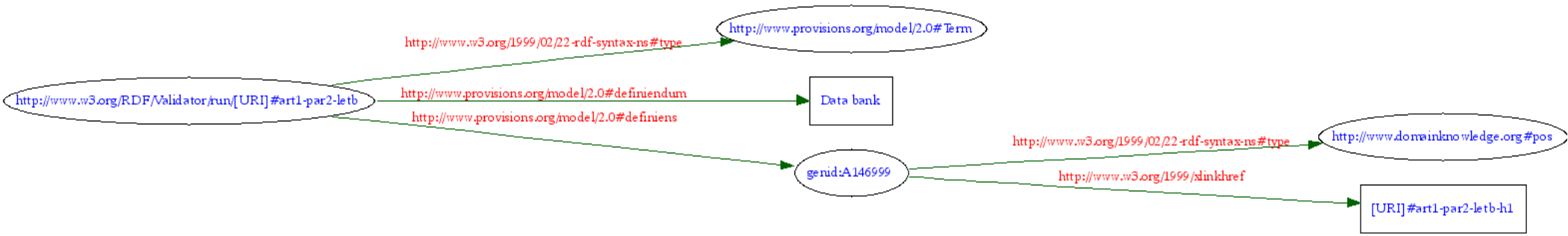
XML Representation

```
<paragraph id="art1-com2-letb">

<num>2.</num> <clause>For the purposes of this Act:</clause>

<letter id="art1-par1-leta"><num>a)</num>"data bank" shall mean <h:span id="art1-par2-letb-h1">any
    set of personal data, divided into one or more units located in one or more places, organized
    according to specific criteria such as to facilitate their processing</h:span>;

</paragraph>
```

“Term” provision instance (an attribute is a reference to a text portion)

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:h="http://www.w3.org/HTML/1998/html4"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prv="http://www.provisions.org/model/2.0#"
xmlns:dk =”http://www.domainknowledge.org#”>


  <rdf:Description rdf:about="[URI]#art1-par2-letb">


     <rdf:type rdf:resource="http://www.provisions.org/model/2.0#Term"/>

     <prv:definiendum>Data bank</prv:definiendum>

     <prv:definiens><dk:pos xlink:href='[URI]#art1-par2-letb-h1'/></prv:definiens>

</rdf:Description>

</rdf:RDF>
```

Graph of a “Term” provision instance

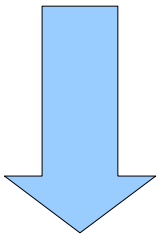


Semantic Mark-up Example: a “Duty”

CHAPTER II - OBLIGATIONS RELATING TO THE CONTROLLER

Article 7 (Notification)

1. A controller intending to process personal data falling within the scope of application of this Act shall have to notify the Garante thereof, exclusively in the cases and manner set out in the regulations as per Article 33(3), if the processing is liable to adversely affect the data subject's rights and freedoms on account of either the relevant mechanisms or the nature of the personal data



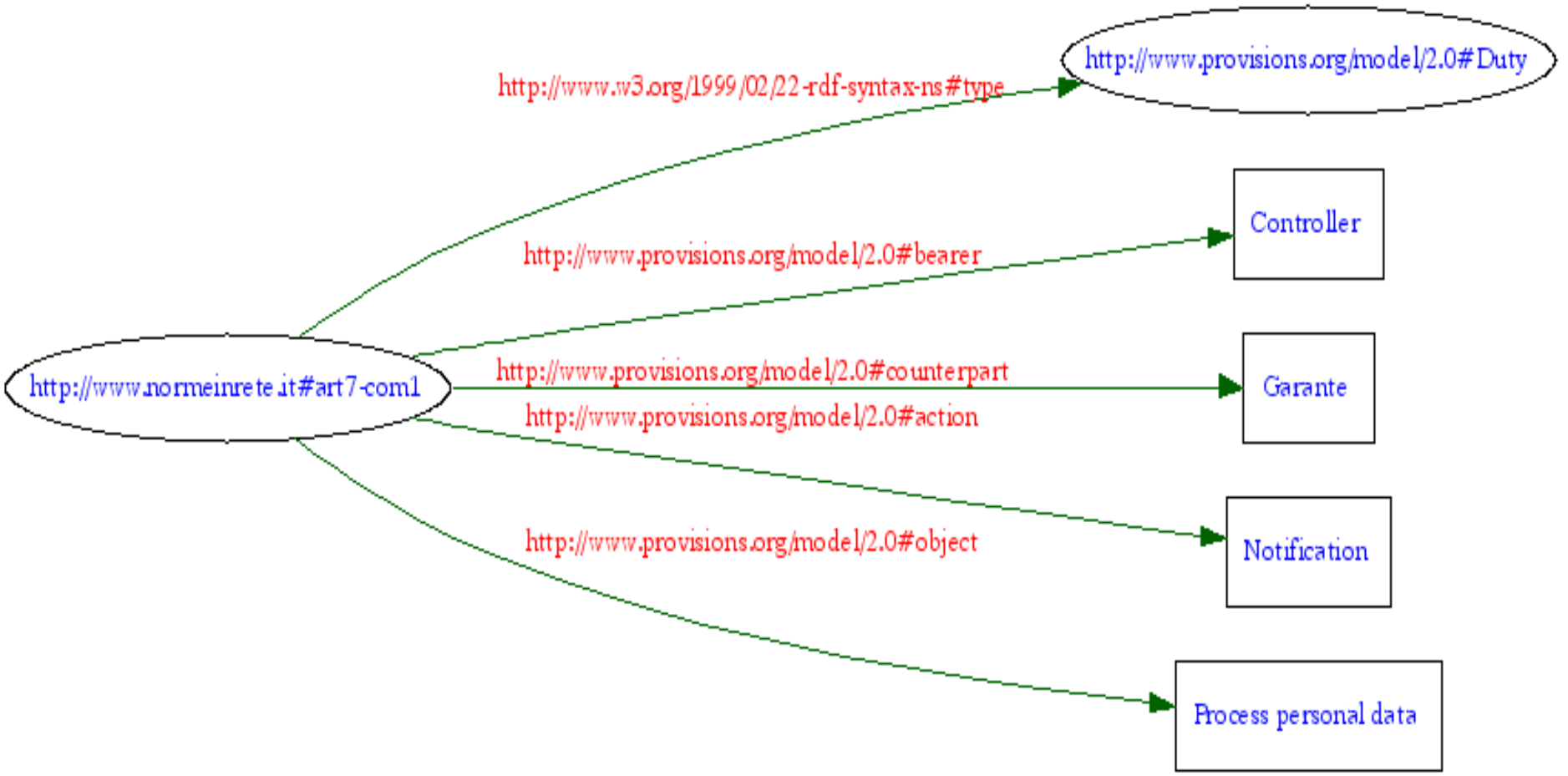
XML Representation

```
<paragraph id="art7-par1"><num>1.</num>A controller intending ... shall have to notify the
    Garante thereof, ... if the processing is liable to adversely affect the data subject's
    rights and freedoms on account of either the relevant mechanisms or the nature of the
    personal data.
</paragraph>
```

“Duty” instance (Arguments are Literals)

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prv="http://www.provisions.org/model/2.0#">

<rdf:Description rdf:about="[URI]#art7-par1">

    <rdf:type rdf:resource="http://www.provisions.org/model/2.0#Duty"/>

    <prv:bearer>Controller</prv:bearer>

    <prv:counterpart>Garante</prv:counterpart>

    <prv:action>Notification</prv:action>

    <prv:object>Process personal data</prv:object>

</rdf:Description>

</rdf:RDF>
```

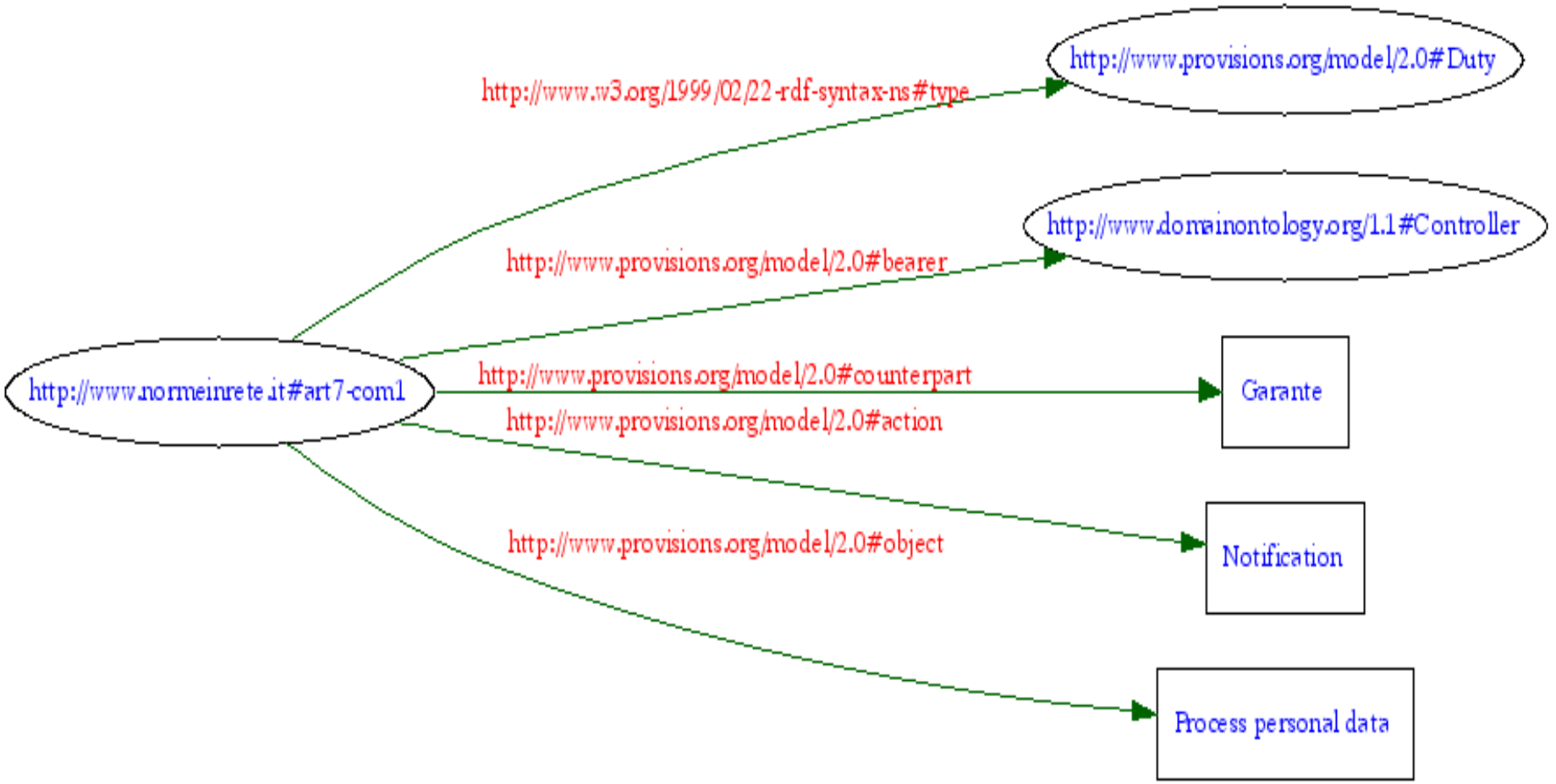
Graph of a “Duty” instance



“Duty” instance (an attribute is an external ontology concept)

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prv="http://www.provisions.org/model/2.0#">

<rdf:Description rdf:about="[URI]#art7-par1">

    <rdf:type rdf:resource="http://www.provisions.org/model/2.0#Duty"/>

    <prv:bearer rdf:resource=“http://www.domainknowledge.org/1.1#Controller”>

    <prv:counterpart>Garante</prv:counterpart>

    <prv:action>Notification</prv:action>

    <prv:object>Process personal data</prv:object>

</rdf:Description>

</rdf:RDF>
```

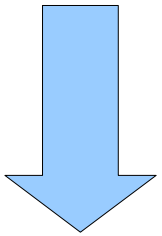
Graph of a “Duty” Instance
An argument is external ontology concept



Semantic Mark-up Example: a “Violation”

Art 34.

1. Whoever fails to promptly submit the notification required under Articles 7, 16(1) and 28 or provides incomplete information in a notification, in breach of his/her duties, shall be the subject of an administrative sanction entailing payment of an amount ranging between Lit 10 million (i.e., euro 5.164,6) and Lit 60 million (i.e., euro 30.987,4), as well as of the additional sanction consisting in publication of the relevant injunction/order.



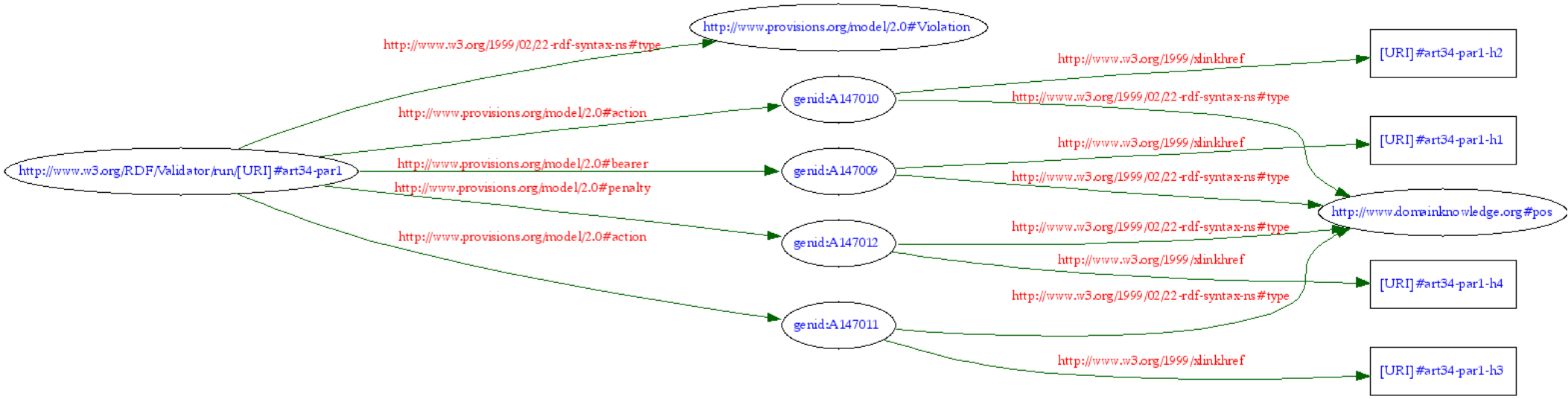
XML Representation

```
<paragraph id="art34-par1">

<num>1.</num> <h:span id="art34-com1-h1">Whoever</h:span> fails to promptly  <h:span id="art34-par1-h2">submit the notification required under Articles 7, 16(1) and 28 </h:span>  or <h:span id="art34-par1-h3"> provides incomplete information in a notification </h:span>, in breach of his/her duties, shall be the subject of an <h:span id="art34-par1-h4">administrative sanction entailing payment of an amount ranging between Lit 10 million (i.e., euro 5.164,6) and Lit 60 million (i.e., euro 30.987,4), as well as of the additional sanction consisting in publication of the relevant injunction/order </h:span>

</paragraph>.
```

RDF representation of a “Violation” Instance (arguments are referred portions of text)

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:h="http://www.w3.org/HTML/1998/html4"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:prv="http://www.provisions.org/model/2.0#"
xmlns:dk=”http://www.domainknowledge.org#”>

<rdf:Description rdf:about="[URI]#art34-par1">

    <rdf:type rdf:resource="http://www.provisions.org/model/2.0#Violation"/>

    <prv:bearer><dk:pos xlink:href=“[URI]#art34-par1-h1”/></prv:bearer>

    <prv:action><dk:pos xlink:href=“[URI]#art34-par1-h2”/></prv:action>

    <prv:action><dk:pos xlink:href=“[URI]#art34-par1-h3”/></prv:action>

    <prv:penalty><dk:pos xlink:href=“[URI]#art34-par1-h4”/></prv:penalty>

</rdf:Description>

</rdf:RDF>
```

Graph of a “Violation” Instance



References

- Resource Description Framework (RDF) Model and Syntax Specification
  - http://www.w3.org/TR/REC-rdf-syntax/
- Resource Description Framework (RDF) Schema Specification 1.0
  - http://www.w3.org/TR/2000/CR-rdf-schema-20000327/
- http://www.w3.org/RDF/Validator/

RDFS / RDF Classes

Element	Class of	Subclass of
rdfs:Class	All classes	
rdfs:Datatype	Data types	Class
rdfs:Resource	All resources	Class
rdfs:Container	Containers	Resource
rdfs:Literal	Literal values (text and numbers)	Resource
rdf:List	Lists	Resource
rdf:Property	Properties	Resource
rdf:Statement	Statements	Resource
rdf:Alt	Containers of alternatives	Container
rdf:Bag	Unordered containers	Container
rdf:Seq	Ordered containers	Container
rdfs:ContainerMembershipProperty	Container membership properties	Property
rdf:XMLLiteral	XML literal values	Literal

RDFS / RDF Properties

Element	Domain	Range	Description
rdfs:domain	Property	Class	The domain of the resource
rdfs:range	Property	Class	The range of the resource
rdfs:subPropertyOf	Property	Property	The property is a sub property of a property
rdfs:subClassOf	Class	Class	The resource is a subclass of a class
rdfs:comment	Resource	Literal	The human readable description of the resource
rdfs:label	Resource	Literal	The human readable label (name) of the resource
rdfs:isDefinedBy	Resource	Resource	The definition of the resource
rdfs:seeAlso	Resource	Resource	The additional information about the resource
rdfs:member	Resource	Resource	The member of the resource
rdf:first	List	Resource	
rdf:rest	List	List	
rdf:subject	Statement	Resource	The subject of the resource in an RDF Statement
rdf:predicate	Statement	Resource	The predicate of the resource in an RDF Statement
rdf:object	Statement	Resource	The object of the resource in an RDF Statement
rdf:value	Resource	Resource	The property used for values
rdf:type	Resource	Class	The resource is an instance of a class

RDF Attributes

Element	Domain	Range	Description
rdf:about			Defines the resource being described
rdf:Description			Container for the description of a resource
rdf:resource			Defines a resource to identify a property
rdf:datatype			Defines the data type of an element
rdf:ID			Defines the ID of an element
rdf:li			Defines a list
rdf:_n			Defines a node
rdf:nodeID			Defines the ID of an element node
rdf:parseType			Defines how an element should be parsed
rdf:RDF			The root of an RDF document
xml:base			Defines the XML base
xml:lang			Defines the language of the element content
rdf:aboutEach			(removed)
rdf:aboutEachPrefix			(removed)
rdf:bagID			(removed)