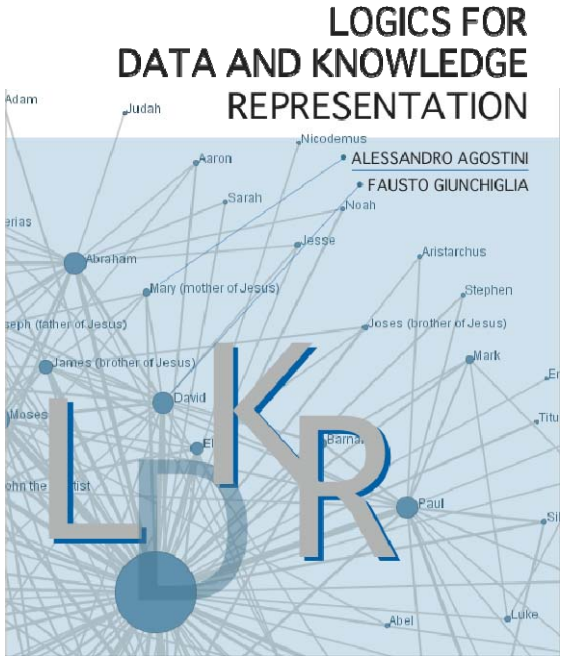


Logics for Data and Knowledge Representation

Resource Description Framework (RDF)

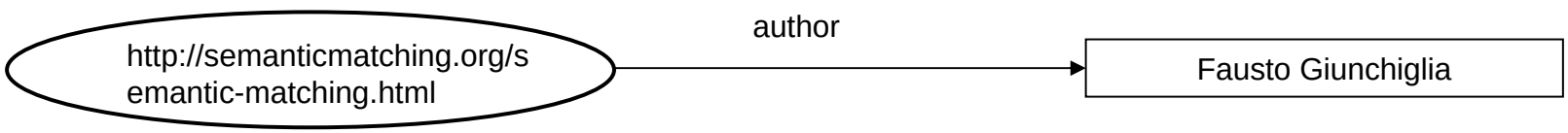
Fausto Giunchiglia and Biswanath Dutta

Outline

- Introduction
- Fundamentals of RDF
- Syntax
- Capabilities of RDF
  - Containers
  - Collections
  - Reification
- RDF Summary
- RDF Schema
  - RDF vs. RDFS
  - RDF/ RDFS: Core classes and Properties
- RDFS Summary

Introduction: What is RDF

- RDF is a data model
  - use in representing information about resources in the World Wide Web (WWW)
  - can be seen as directed graph with labeled nodes and arcs or as an object-oriented model (object/attribute/value)



Introduction: What is RDF

- domain, and application independent
- goal is to avail the information for applications to process, rather than only display to the human beings
- is based on the idea of identifying things using Web identifiers (i.e., *Uniform Resource Identifiers*, or *URIs*)

- RDF data model is an abstract, conceptual layer

Fundamentals of RDF

- Three fundamental concepts in RDF are:
  - Resources
  - Properties
  - Statements

Fundamentals of RDF

Resources

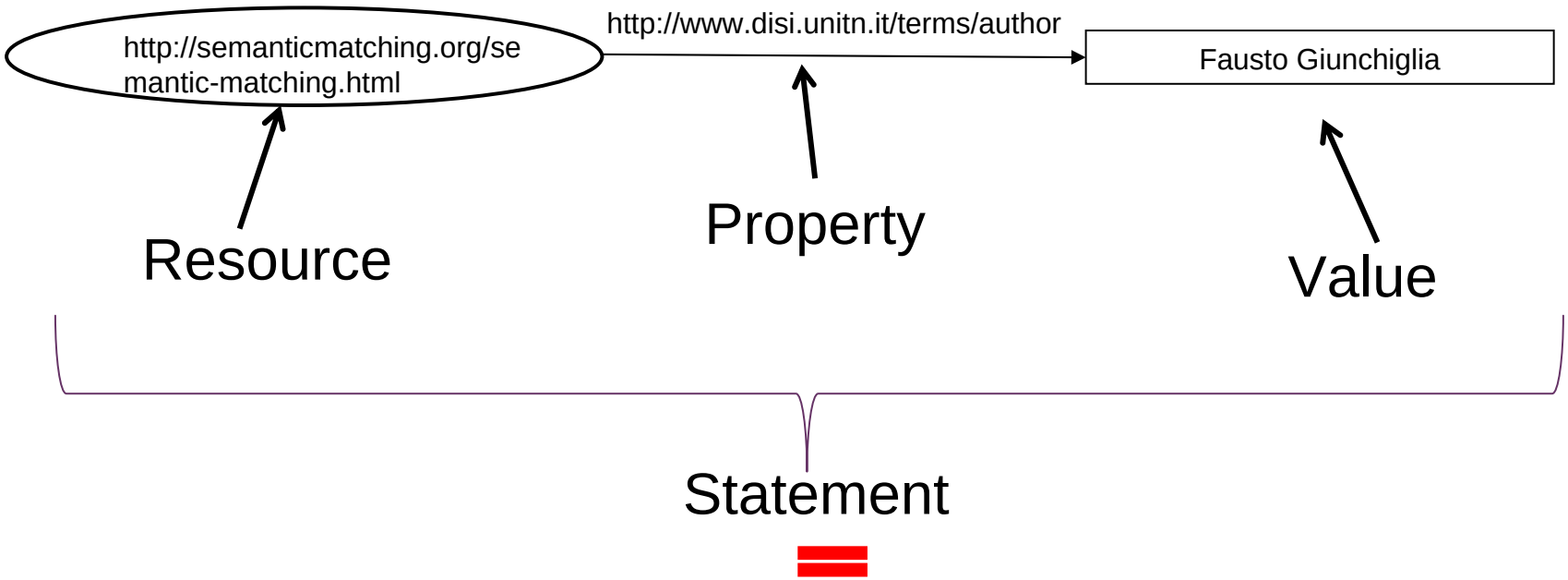
- Resource can be considered as an object, a "thing", we want to talk about
  - For example, web page, books, authors, publishers, people, organizations, places, etc.
- All resource has a URI (i.e., Universal Resource Identifier)
- A URI can be
  - a URL (Web address) or
  - some other kind of unique identifier

Fundamentals of RDF

Properties

- Properties are a special kind of resources
- They describe relations between resources
  - For example, “author”, “publisher”, “hasStudent”, “teach”, “age”, “title”, “name”, “locatedIn”, etc.
- Properties are also identified by URIs
  - Advantages of using URIs:
    - A global, worldwide, unique naming scheme
    - Reduces the homonym (e.g., title) problem of distributed data representation

Fundamentals of RDF



“http://semanticmatching.org/semantic-matching.html” has “http://www.disi.unitn.it/terms/author” Fausto Giunchiglia

Important: value can be another resource or literals (e.g., character strings such as “Fausto Giunchiglia", and values from other data types such as integers and dates, as the values of properties)

Fundamentals of RDF

Statements

- RDF statements consist of

  resources (= nodes) = subject
  which have properties = predicate
  which have values (= nodes, strings) = object

  predicate(subject, object)

- Statements assert the properties of resources

- A statement is a triple of *object-attribute-value*
  - consisting of a resource, a property, and a value

Fundamentals of RDF

Three views of a RDF statement

- A triple
- A piece of a graph
- A piece of XML code

Hence, a RDF document can be seen as,

- A set of triples
- A graph (semantic net)
- An XML document

Fundamentals of RDF

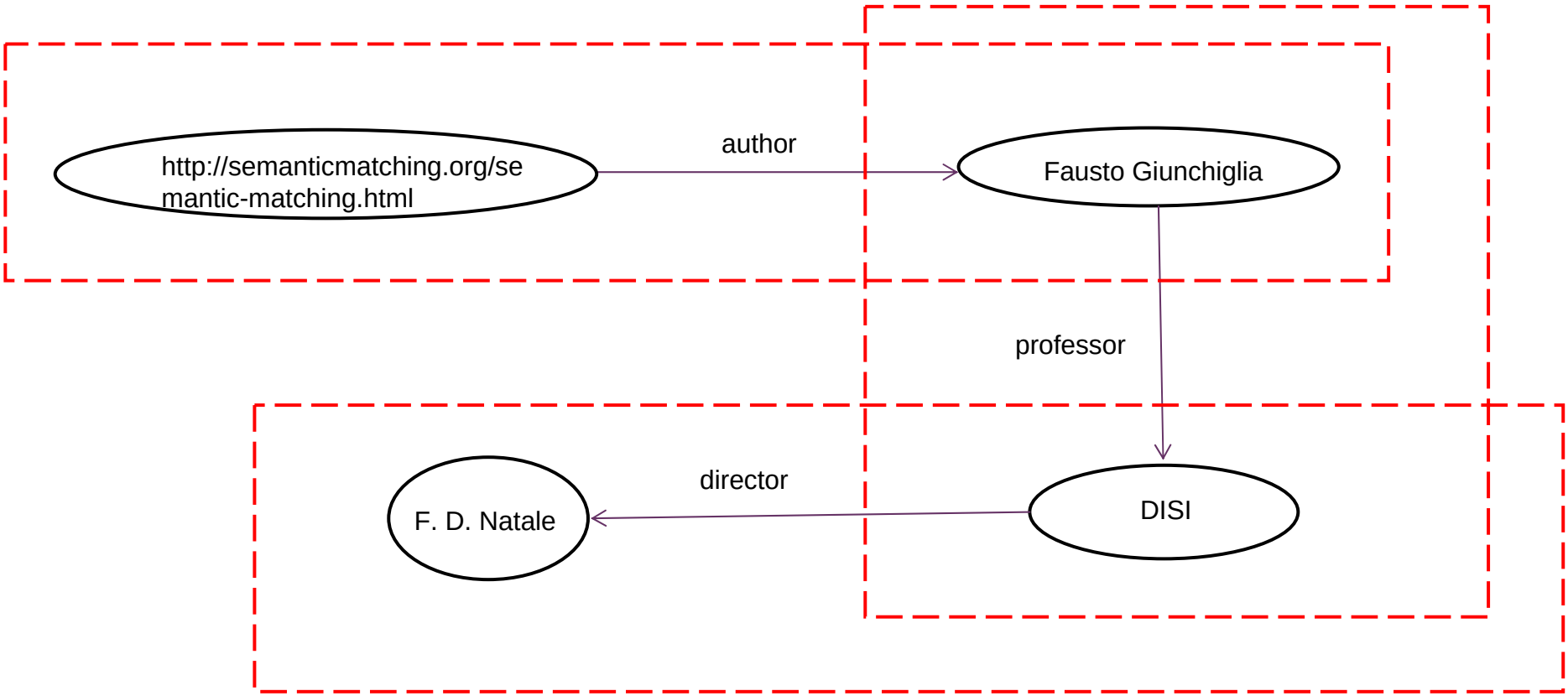
Statements as Triples

{http://semanticmatching.org/semantic-matching.html, http://disi.unitn.it/terms/author, Fausto Giunchiglia}

- Triple (x, P, y) can be considered as a logical formula P(x, y)
  - Binary predicate P relates object x to object y

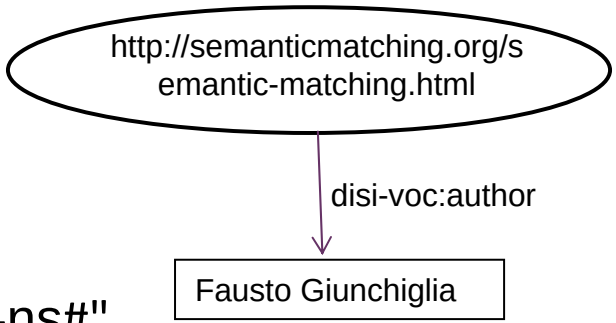
Fundamentals of RDF

A Set of Triples as a Semantic Net



Fundamentals of RDF

Statement in XML

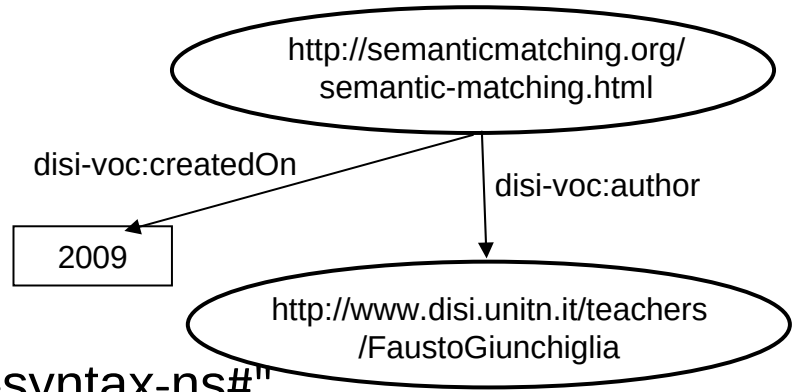



```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:disi-voc="http://www.disi.unitn.it/terms/">

     <rdf:Description
      rdf:about=" http://www.http://semanticmatching.org/semantic-matching.html">
          <disi-voc:author>Fausto Giunchiglia</disi-voc:author>
     </rdf:Description>
</rdf:RDF>
```

RDF Syntax

- The RDF graphs are useful tool for human understanding while
- The Semantic Web (SW) vision requires "machine accessible" and "machine processable" representations
- RDF uses eXtensible Markup Language (XML) where XML is used as a transfer syntax for RDF
  - Important: XML is not a part of the RDF data model
- RDF provides only binary predicates (properties)
  - E.g., P(x,y), here, binary predicate P relates object x to object y
- Property Names and Values are always unambiguous

RDF/XML




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

    <rdf:Description
      rdf:about="http://www.http://semanticmatching.org/semantic-
   matching.html">
         <disi-voc:author
   rdf:resource="http://www.disi.unitn.it/teachers/FaustoGiunchiglia"/>
         <disi-voc:createdOn>2009</disi-voc:createdOn>
    </rdf:Description>
</rdf:RDF>
```

RDF/XML

- An RDF document is represented by an XML element with the tag rdf:RDF
  - The content of this rdf:RDF element is a number of descriptions, which use rdf:Description tags.
- The rdf:Description element makes a statement about the resource http://www.http://semanticmatching.org/semantic-matching.html
- Within the description
  - the property “disi-voc:author” is used as a tag
    - the content “http//www.disi.unitn.it/teachers/FaustoGiunchiglia” is the value of the property “disi-voc:author”
  - the content of the property element “disi-voc:createdOn” is the object of the statement, the plain literal, 2009.

RDF/XML

- Every description makes a statement about a resource, identified in 3 ways:
  - an about attribute, referencing an existing resource
  - an ID attribute, creating a new resource
  - without a name, creating an anonymous resource

rdf:about vs. rdf:ID

- An element rdf:Description has
  - an rdf:about attribute indicates that the resource has been "defined" elsewhere (*refer slide 15*)
    - Assigns an absolute identifier in general
  - An rdf:ID attribute indicates that the resource is defined (*refer slide 22*)
    - Assigns a fragment identitfier (relative URIref)

- Sometimes it is good (for better organization and human readability) to have things defined in one location, while other location state "additional" properties

Data Types

- Unlike typical programming languages and database systems, RDF has no built-in set of data types of its own (e.g., integers, strings, dates)
- Basic XML Schema datatypes such as xsd:string, xsd:boolean, xsd:time, xsd:date, etc. are suitable for use in RDF
  - Important: some of the built-in XML Schema datatypes are not suitable for use in RDF (e.g., xsd:duration)
- RDF provides no mechanism for defining new datatypes
- But the use of any externally defined data typing scheme is allowed in RDF documents

Data Types

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

<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>

<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:disi-voc="http://www.disi.unitn.it/terms/">


     <rdf:Description rdf:about="http://www.http://semanticmatching.org/semantic-
    matching.html">
            <disi-voc:author>Fausto Giunchiglia</disi-voc:author>
            <disi-voc:title>Professor</disi-voc:title>
            <disi-voc:age rdf:datatype="&xsd:integer”>55</disi-voc:age>
     </rdf:Description>
</rdf:RDF>
```

attribute rdf:datatype="&xsd:integer”, a typed literal is used to indicate the datatype of the value of the property “age”

rdf:type

- Similar to the programming languages, concept of objects having different *types* or *classes*, RDF also supports this concept by providing a predefined property, rdf:type

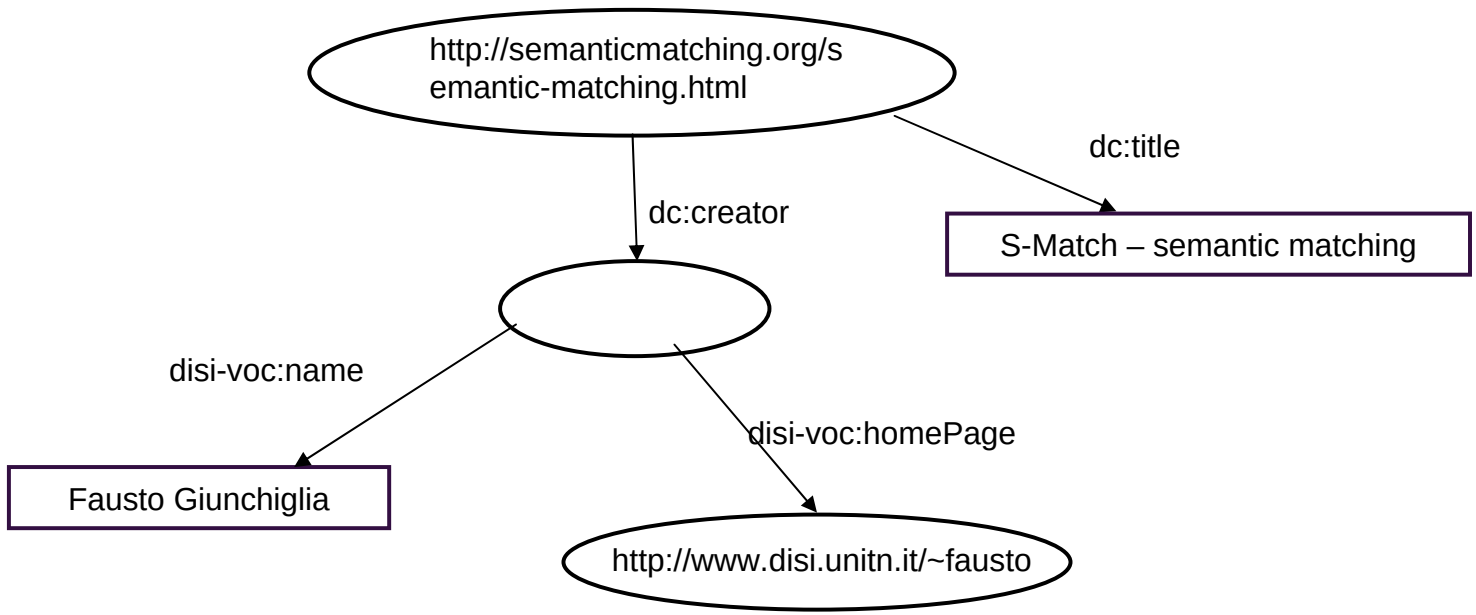
- When an RDF resource is described with an rdf:type property, *the value of that property is considered to be a resource* that represents a category or *class* of things, and *the subject of that property is considered to be an instance of that category or class*

rdf:type

```
<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:disi-voc="http://www.disi.unitn.it/terms/">

<rdf:Description rdf:ID=”ICT001">
        <rdf:type rdf:resource="http://www.disi.unitn.it/course"/>
        <disi-voc:courseName>LDKR</disi-voc:courseName>
        <disi-voc:isTaughtBy rdf:resource=”DISI111"/>

<rdf:Description rdf:ID=”DISI111">
        <rdf:type rdf:resource="http://www.disi.unitn.it/lecurer"/>
        <disi-voc:name>Fausto Giunchiglia</disi-voc:name>
        <disi-voc:title>Professor</disi-voc:title>
        <disi-voc:age rdf:datatype="&xsd:integer”>55</disi-voc:age>
    </rdf:Description>
</rdf:RDF>
```

Blank Node

RDF/XML allows representation of graphs that include nodes without any URIrefs, i.e., the blank nodes



Blank Node: RDF/XML

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:disi-voc="http://www.disi.unitn.it/terms/">

<rdf:Description rdf:about="http://www.http://semanticmatching.org/semantic-
    matching.html">
    <dc:title>S-Match – semantic matching</dc:title>
    <dc:creator rdf:nodeID="abc"/>
</rdf:Description>

 <rdf:Description rdf:nodeID="abc">
    <disi-voc:name>Fausto Giunchiglia</disi-voc:name>
    <disi-voc:homePage rdf:resource="http://disi.unitn.it/~fausto"/>
</rdf:Description>
</rdf:RDF>
```

Container

- A container is a resource that contains things
- Allow grouping of resources (including blank nodes) or literals values
  - about which we want to make statements as a whole
- The contained things are called members
- A typical use of a container is to indicate that the value of a property is a group of things
  - For example, we may wish to talk about a list of students taking a particular course, or, we may wish to talk about a list of courses offered by a particular lecturer, and so on.

Container

- The content of container elements (i.e., members) are named rdf:_1, rdf:_2, etc.
  - Alternatively rdf:li
  - Important: RDF/XML provides rdf:li as a convenience element to avoid having to explicitly number each membership property

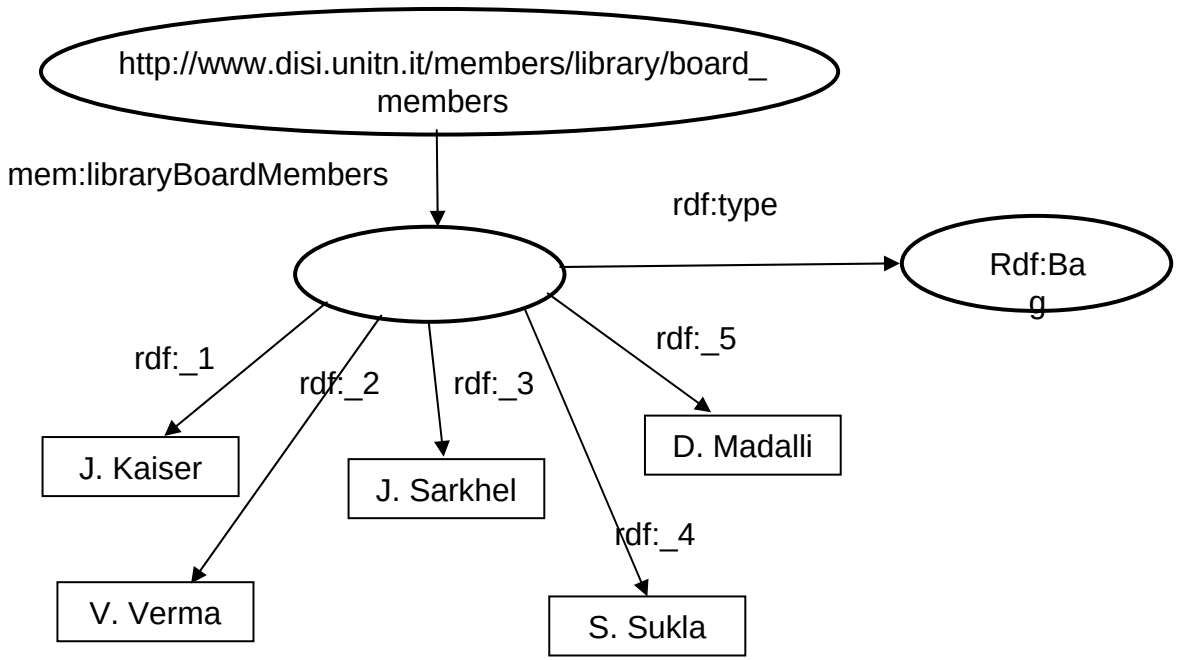
Container

RDF defines three types of containers:

- rdf:Bag an unordered container
  - E.g. members of the university library board, documents in a folder
- rdf:Seq an ordered container
  - E.g. modules of a course, items on an agenda, an alphabetized list of staff members (order is imposed)
- rdf:Alt a set of alternatives
  - E.g., alternative (language) translations for the title of a book, or describing a list of alternative Internet sites at which a resource might be found,
  - Important: an application using a property whose value is an Alt container should be aware that it can choose any one of the members of the group as appropriate

- Important: describing a resource as being one of these types of containers, the resource is given an rdf:type property whose value is one of the predefined resources rdf:Bag, rdf:Seq, or rdf:Alt (whichever is appropriate)

Container



Triples

{http://disi.unitn.it/members/library/board_members,mem:libraryBoardMemebers, x}
{x, rdf:_1, “J. kaiser”}
{x, rdf:_2, “V. Verma”}
{x, rdf:_3, “J. Sarkhel”}
{x, rdf:_4, “S. Sukla”}
{x, rdf:_5, “D. Madalli”}
{x, rdf:type, rdf:bag}

Container (Bag): RDF/XML

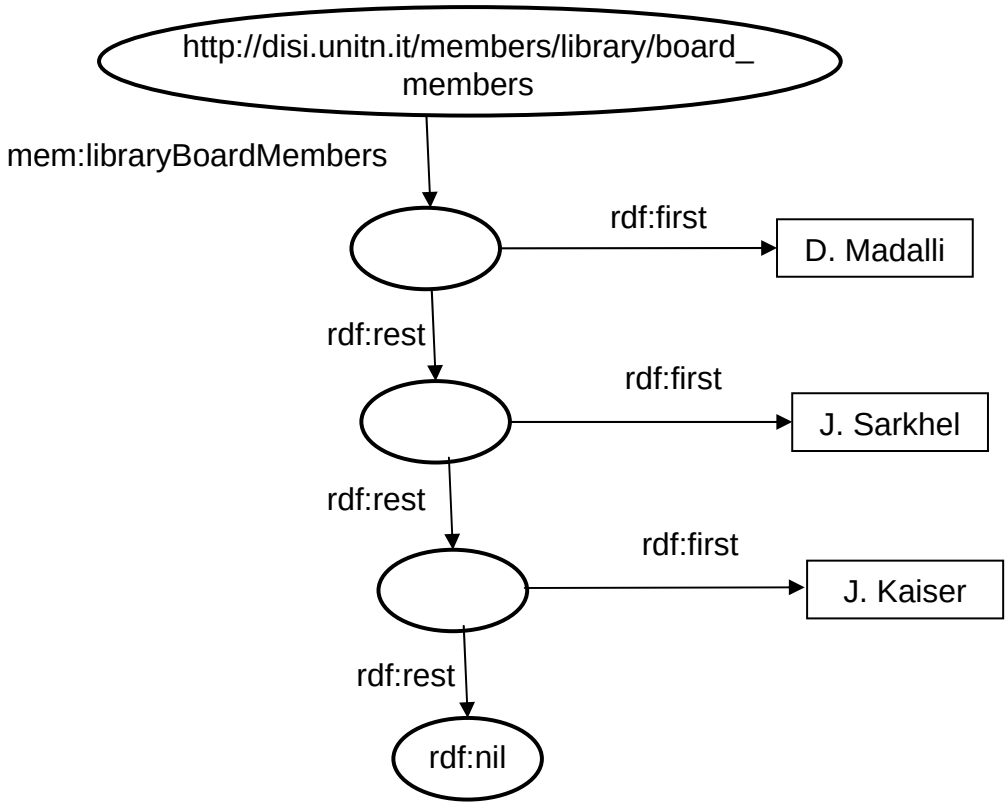
```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:mem="http://www.disi.unitn.it/members/vocabularies/">

<rdf:Description rdf:about="http://disi.unitn.it/members/library/board_members">

  <mem:libraryBoardMembers>
       <rdf:Bag>
            <rdf:li> J. Kaiser</rdf:li>
            <rdf:li> V. Verma</rdf:li>
            <rdf:li>J. Sarkhel</rdf:li>
            <rdf:li> S. Sukla</rdf:li>
            <rdf:li>D. Madalli</rdf:li>
        </rdf:Bag>
   </mem:libraryBoardMembers>

 </rdf:Description>

</rdf:RDF>
```

Important: RDF/XML provides syntactic shorthand, similar like HTML lists

RDF Collections

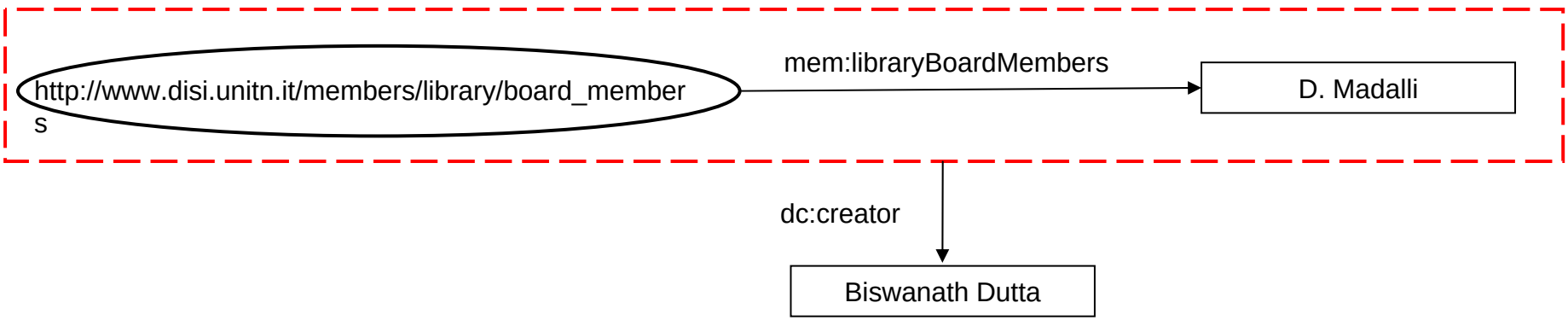
- Limitation of those containers is that there is no way to close them
  - E.g., “these are all the members of the container”
  - There is no mechanism enforcing the unique value constraints
- RDF provides support for describing groups containing only the specified members, in the form of RDF collections
  - list structure in the RDF graph constructed using a predefined collection vocabulary: rdf:List, rdf:first, rdf:rest and rdf:nil

RDF Collections



Reification

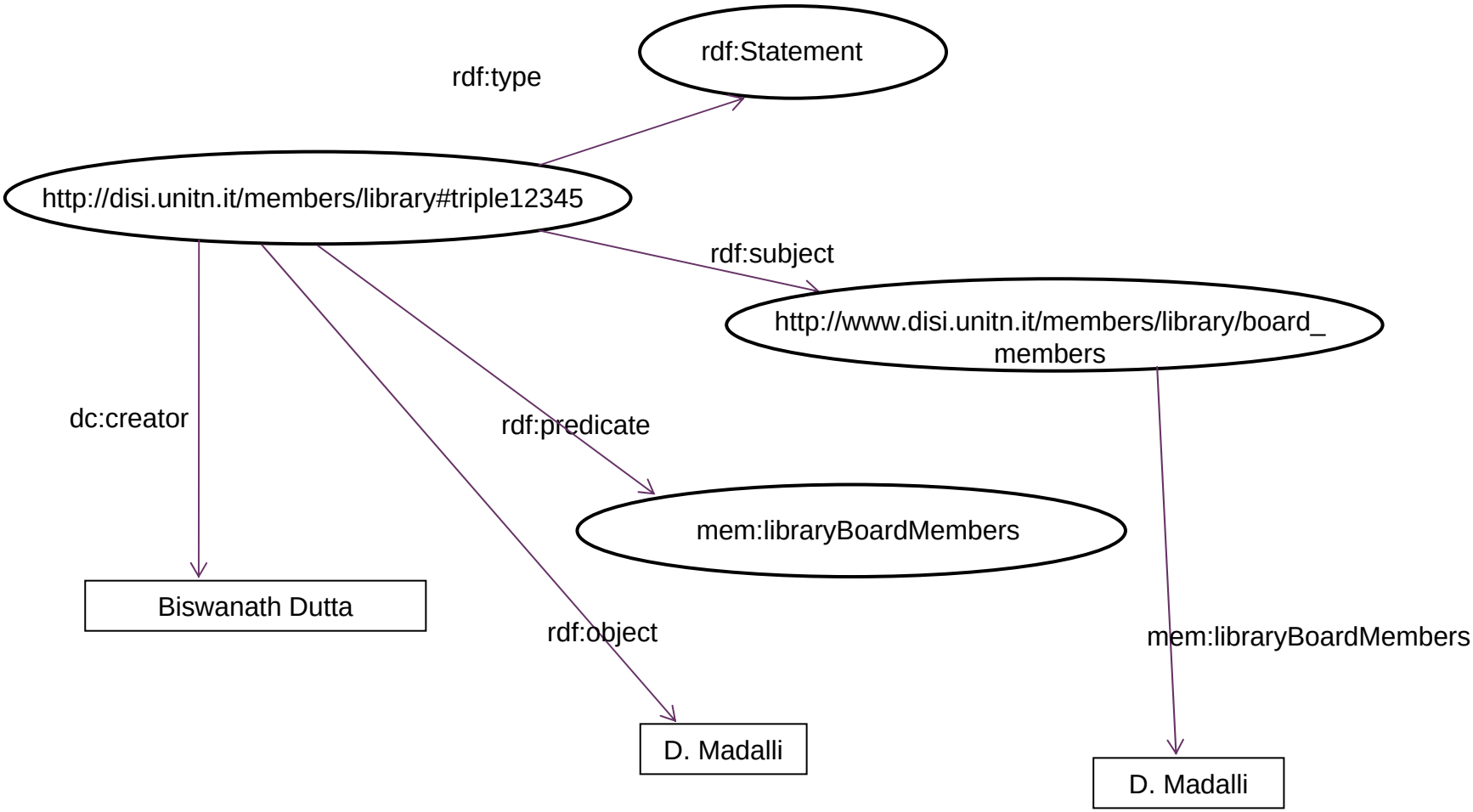
- In RDF, it is possible to make statements about statements
- Such statement can be used in building trust
- Can be referred as provenance information (like, who made, where, when made)
- Important: solution is to assign a unique identifier to each statement



Reification

- RDF provides built-in vocabularies for describing RDF statements, such as,
  - type: rdf:Statement, and
  - properties: rdf:Subject, rdf:Predicate and rdf:Object
- A description of a statement using these vocabulary is called a reification of the statement.

Reification



Reification: RDF/XML

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:mem="http://www.disi.unitn.it/members/vocabulary/">

<rdf:description rdf:about="http://www.disi.unitn.it/members/library/board_members">
    <mem:libraryBoardMembers>D. Madalli</mem:libraryBoardMembers>
</rdf:description>

  <rdf:Statement rdf:about="http://disi.unitn.it/members/library#triple12345">
    <rdf:subject rdf:resource="http://www.disi.unitn.it/members/library/board_members"/>
    <rdf:predicate rdf:resource="mem:libraryBoardMembers"/>
    <rdf:object>D. Madalli</rdf:object>
    <dc:creator>Biswanath Dutta</dc:creator>
  </rdf:Statement>
</rdf:RDF>
```

RDF: Summary

- Even though RDF has its peculiarities
  - For example, syntax is hard
- Is not an optimal modeling language (!!!) but
  - It is already a de facto standard
- It has sufficient expressive power
- Allows mapping of information unambiguously to a model
  - Standardise the syntax and abstract semantics
- Providing a standard way of defining standard vocabularies (but without defining any)
  - RDF Schema

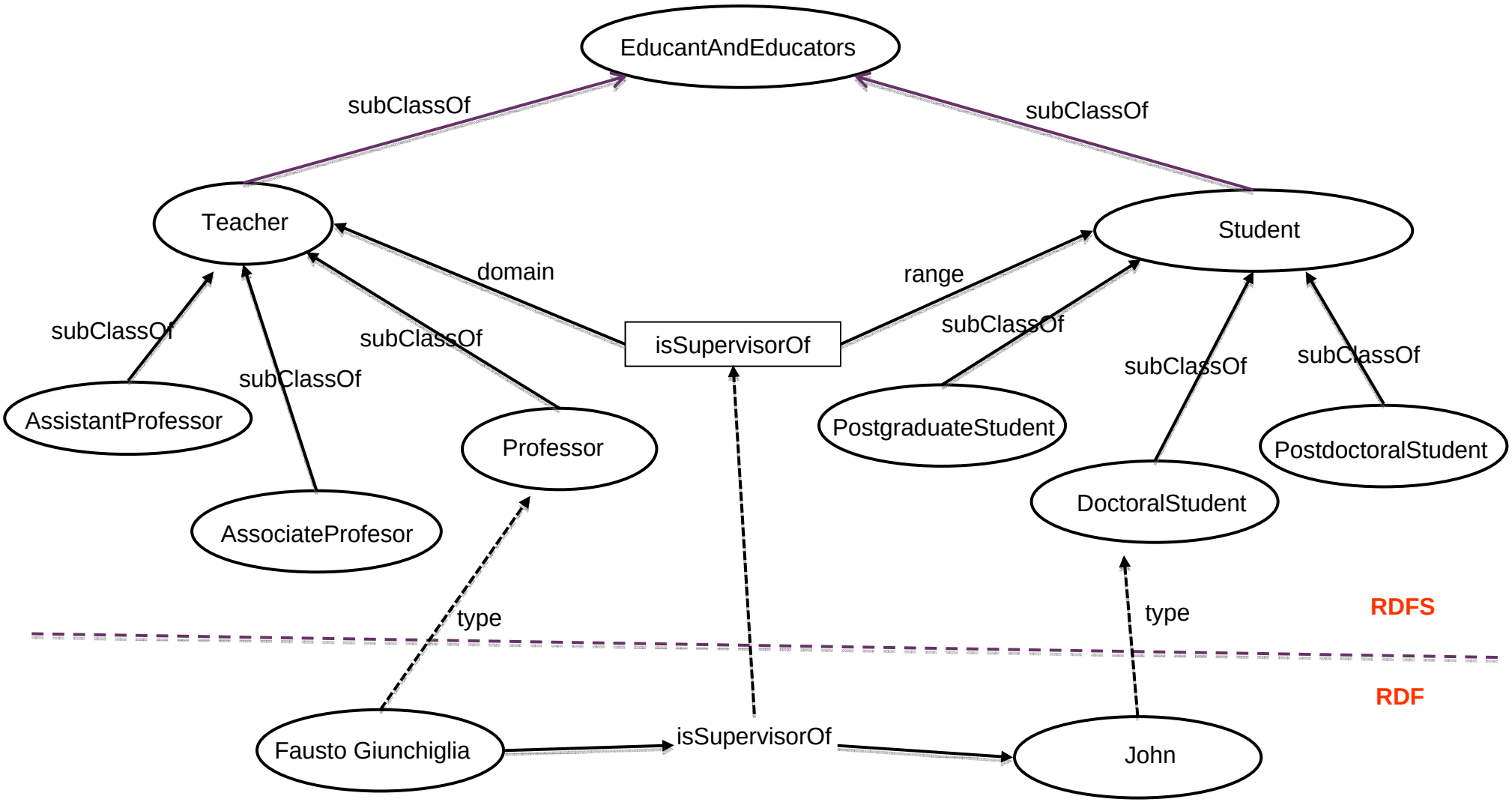
RDF Schema (RDFS)

- RDF provides a way to express simple statements about resources, using named properties and values, but
- We also need the ability to define the vocabularies (terms) they intend to use in those statements, specifically, to indicate that they are describing specific types or classes of resources
- Users can specify in RDF Schema
  - Classes and properties
  - Class hierarchies
    - Creating subclasses of classes
  - a new class can be created by extending an existing class
  - Class instances
  - Property hierarchies
- A class can have multiple super-classes

RDF schema: type facilities

- RDF Schema definitions consist of classes (= “types”) and properties
- Individual object (s) belong to a class is referred as instances of that class
- The relationship between instances and classes is expressed by rdf:type
- Schema definitions allow constraints on properties (which express validation conditions)
  - domain constraints link properties with classes
  - range constraints limit property values
- Schema definitions are expressed in RDF itself
  - Important: Vocabulary descriptions (i.e., schemas) written in the RDF Schema language are legal RDF graphs

RDF Layer vs. RDFS Layer



Core Classes

- Important: RDF Schema itself does not provide a vocabulary of application-specific classes
  - Provides a framework to do so

1. rdfs:Resource the class of everything (i.e., class of all resources)
2. rdfs:Class the class of all classes
3. rdfs:Literal the class of all literals (strings)
4. rdfs:Datatype is both an instance of and a subclass of rdfs:Class
5. rdf:Property the class of all RDF properties; and is an instance of rdfs:Class.

Core Properties

- rdf:type which 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 super-properties
- rdfs:domain specifies the domain of a property P
  - The class of those resources that may appear as subjects in a triple with predicate P
  - If the domain is not specified, then any resource can be the subject
- rdfs:range which specifies the range of a property P
  - The class of those resources that may appear as values in a triple with predicate P

Reification

- rdf:Statement the class of all reified statements
- 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

Containers Classes and Properties

- rdf:Bag the class of bags
- rdf:Seq the class of sequences
- rdf:Alt the class of alternatives
- rdfs:container is a super-class of all container classes, including the above three classes
- rdfs:member is an instance of rdf:Property that is a super-property of all the container membership properties i.e. each container membership property has an rdfs:subPropertyOf relationship to the property rdfs:member.

RDF Collections

- rdf:List is an instance of rdfs:Class that can be used to build descriptions of lists and other list-like structures.
- rdf:first is an instance of rdf:Property that can be used to build descriptions of lists and other list-like structures.
- rdf:rest is an instance of rdf:Property that can be used to build descriptions of lists and other list-like structures.
- rdf:nil the resource rdf:nil is an instance of rdf:List that can be used to represent an empty list or other list-like structure.

Utility Properties

- rdfs:seeAlso relates a resource to another resource that explains it
- rdfs:isDefinedBy is a subproperty of rdfs:seeAlso and relates a resource to the place where its definition, typically an RDF schema, is found
- rdfs:comment typically longer text, can be associated with a resource
- rdfs:label a human-friendly label (name) is associated with a resource

RDF Schema: Summary

- RDF Schema is a primitive ontology language
- The key concepts in RDF Schema are:
  - Class, and class relations, property, and property relations,
  - domain and range restrictions
- Is quite primitive as a modelling language for the Web
  - Offers limited modelling primitives with fixed meaning
- Many desirable modelling primitives are missing
- So, we need an ontology layer on top of RDF and RDF Schema

For further details…

- RDF Primer, http://www.w3.org/TR/rdf-primer/
- RDF Concepts and Abstract Syntax [RDF-CONCEPTS], http://www.w3.org/TR/rdf-concepts/
- RDF/XML Syntax Specification [RDF-SYNTAX], http://www.w3.org/TR/rdf-syntax-grammar/
- RDF Vocabulary Description Language 1.0: RDF Schema [RDF-VOCABULARY], http://www.w3.org/TR/rdf-schema/
- RDF Semantics [RDF-SEMANTICS], http://www.w3.org/TR/2004/REC-rdf-mt-20040210/
- RDF Test Cases [RDF-TESTS] , http://www.w3.org/TR/rdf-testcases/