RDF and RDF Schema

Raúl García Castro, Óscar Corcho

Boris Villazón-Terrazas
bvillazon@isoco.com
@boricles

Slides available at: http://www.slideshare.net/boricles/
Overview

RDF
- Introduction
- RDF components
- Serializing RDF
- Hands-on

RDF Schema
- Introduction
- Main RDF-S components
- Serializing RDF-S
- Hands-on
- RDF(S) Limitations
- Hands-on

RDF(S) management APIs
RDF: Resource Description Framework

Goal
- To describe the semantics of information in a machine-processable way

W3C recommendations
- Model
- Syntax
- Semantics
RDF(S) in the Semantic Web
- Overview
- **RDF**
  - Introduction
  - RDF components
  - Serializing RDF
  - Hands-on
- **RDF Schema**
  - Introduction
  - Main RDF-S components
  - Serializing RDF-S
  - Hands-on
  - RDF(S) Limitations
  - Hands-on
- **RDF(S) management APIs**
Also known as “triples”
- [Subject, Predicate, Object]

“Raúl is a member of the Ontology Engineering Group”
- [Raúl, is member of, Ontology Engineering Group]

“Raúl’s full name is Raúl García Castro”
- [Raúl, has full name, Raúl García Castro]

“Raúl was born on 26 December 1975”
- [Raúl, was born, 26 December 1975]

“The Ontology Engineering Group web page is http://www.oeg-upm.net/”
- [Ontology Engineering Group, has web page, http://www.oeg-upm.net/]

[RDF statements]
RDF graphs are sets of triples

- RDF graphs are sets of triples

- [Graph diagram]
  - Raúl
  - has full name: Raúl García Castro
  - has birth date: 26 December 1975
  - is member of: Ontology Engineering Group
  - has web page: http://www.oeg-upm.net/
- Triple objects can be literals (character strings)
  - Subject and predicates are always resources
- Literals can be typed
  - Usually using XML Schema datatypes
  - RDF provides the \textit{rdf:XMLLiteral} datatype

- \textbf{Example:}

  
  \begin{itemize}
  \item Raúl
  \item is member of Ontology Engineering Group
  \item has full name “Raúl García Castro”
  \item has birth date “1975-12-26”^^xsd:date
  \item has web page http://www.oeg-upm.net/
  \end{itemize}
**URIs in RDF**

- **URI component parts (RFC3986)**
  - http://www.oeg-upm.net:8080/Info/People?position=current#Raul

- **RDF URIs:**
  - Are URI references: URI + Fragment
  - Can contain Unicode characters
  - Identify resources and values (e.g., mailto:rgarcia@fi.upm.es)
Namespaces defined using XML qualified names

URIs under a namespace are called vocabularies

<table>
<thead>
<tr>
<th>Prefix</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>people</td>
<td><a href="http://www.oeg-upm.net/People#">http://www.oeg-upm.net/People#</a></td>
</tr>
<tr>
<td>organization</td>
<td><a href="http://www.oeg-upm.net/Organization#">http://www.oeg-upm.net/Organization#</a></td>
</tr>
<tr>
<td>properties</td>
<td><a href="http://www.oeg-upm.net/Properties#">http://www.oeg-upm.net/Properties#</a></td>
</tr>
<tr>
<td>rdf</td>
<td><a href="http://www.w3.org/1999/02/22-rdf-syntax-ns#">http://www.w3.org/1999/02/22-rdf-syntax-ns#</a></td>
</tr>
<tr>
<td>rdfs</td>
<td><a href="http://www.w3.org/2000/01/rdf-schema#">http://www.w3.org/2000/01/rdf-schema#</a></td>
</tr>
<tr>
<td>xsd</td>
<td><a href="http://www.w3.org/2001/XMLSchema#">http://www.w3.org/2001/XMLSchema#</a></td>
</tr>
</tbody>
</table>

```
properties:isMemberOf

people:Raul

properties:hasFull Name

“Raúl García Castro”

properties:hasBirthDate

“1975-12-26”^^xsd:date

organisation:OEG

properties:hasWebPage

http://www.oeg-upm.net/
```
- Overview
- RDF
  - Introduction
  - RDF components
  - Serializing RDF
  - Hands-on
- RDF Schema
  - Introduction
  - Main RDF-S components
  - Serializing RDF-S
  - Hands-on
  - RDF(S) Limitations
  - Hands-on
- RDF(S) management APIs
The **rdf:type** property is used to classify resources in categories/classes.

- **Classes**
  - category:Person
    - people:Raul
    - people:Oscar
  - category:Animal
    - people:Missy
    - people:Fantasma

- **Instances**
  - people:Raul
  - people:Oscar
  - people:Missy
  - people:Fantasma

The **rdf:Property** class is the class of all properties.
RDF Containers

- **Describe groups of things**
  - A book was created by several authors
  - A lesson is taught by several persons
  - etc.

- **RDF provides a container vocabulary**
  - *rdf:Bag*. Group of resources or literals, including duplicates, where order is not significant
  - *rdf:Seq*. Group of resources or literals, including duplicates, where order is significant
  - *rdf:Alt*. Group of resources or literals that are alternatives (typically for a single value of a property)
- Groups of things represented as a list structure
  - “A container with limits”
- Constructed using `rdf:List`, `rdf:first`, `rdf:rest`, and `rdf:nil`
RDF statements about other RDF statements
- “Raúl believes that Oscar’s birthdate is on Feb 2nd, 1976 and that his e-mail address is ocorcho@fi.upm.es”

Expressed using rdf:Statement, rdf:subject, rdf:_predicate, and rdf:object

RDF Reification
- Allows expressing beliefs (and other modalities)
- Allows expressing trust models, digital signatures, etc.
- Allows expressing metadata about metadata
Sometimes one of the values of a structured value is the main one
- The weight of an item is 2.4 kilograms
- The main value is 2.4, which is expressed with `rdf:value`

Scarcely used
### RDF vocabulary summary

<table>
<thead>
<tr>
<th>Classes</th>
<th>Properties</th>
<th>Individuals</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Classification</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>rdf:Property</code></td>
<td><code>rdf:type</code></td>
<td></td>
</tr>
<tr>
<td><strong>Containers</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>rdf:Bag</code></td>
<td><code>rdf:_1, rdf:_2, rdf:_3…</code></td>
<td></td>
</tr>
<tr>
<td><code>rdf:Seq</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>rdf:Alt</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Collections</strong></td>
<td></td>
<td><code>rdf:nil</code></td>
</tr>
<tr>
<td><code>rdf:List</code></td>
<td><code>rdf:first</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>rdf:rest</code></td>
<td></td>
</tr>
<tr>
<td><strong>Reification</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>rdf:Statement</code></td>
<td><code>rdf:subject</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>rdf:predicate</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>rdf:object</code></td>
<td></td>
</tr>
<tr>
<td><strong>Values</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>rdf:XMLLiteral</code></td>
<td><code>rdf:value</code></td>
<td></td>
</tr>
</tbody>
</table>
Overview

RDF
- Introduction
- RDF components
- Serializing RDF
- Hands-on

RDF Schema
- Introduction
- Main RDF-S components
- Serializing RDF-S
- Hands-on
- RDF(S) Limitations
- Hands-on

RDF(S) management APIs
RDF Serialisations

- **Normative**
  - **RDF/XML** (www.w3.org/TR/rdf-syntax-grammar/)

- **Recommendation (25 February 2014)**
  - **Turtle** (http://www.w3.org/TR/turtle/)

- **Alternative (for human consumption)**
  - **N3** (http://www.w3.org/DesignIssues/Notation3.html)
  - **TriX** (http://www.w3.org/2004/03/trix/)
  - **JSON-LD** (http://www.w3.org/TR/json-ld/)
  - ...

- **Important:** the RDF serializations allow different syntactic variants.
  - E.g., the order of RDF statements has no meaning
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:person="http://www.ontologies.org/ontologies/people#"
    xmlns="http://www.oeg-upm.net/ontologies/people#"
    xml:base="http://www.oeg-upm.net/ontologies/people">

    <rdf:Property rdf:about="http://www.ontologies.org/ontologies/people#hasHomePage"/>
    <rdf:Property rdf:about="http://www.ontologies.org/ontologies/people#hasColleague"/>
    <rdf:Property rdf:about="http://www.ontologies.org/ontologies/people#hasName"/>

    <rdf:Description rdf:about="#Raul"/>
    <rdf:Description rdf:about="#Asun">
        <person:hasColleague rdf:resource="#Raul"/>
        <person:hasHomePage>http://www.fi.upm.es</person:hasHomePage>
    </rdf:Description>
    <rdf:Description rdf:about="#Oscar">
        <person:hasColleague rdf:resource="#Asun"/>
        <person:hasName>Óscar Corcho García</person:hasName>
    </rdf:Description>
</rdf:RDF>
Turtle. URIs, literals

- **URIs**
  - Enclosed in `<>
    `<URI>`
  - or
    `@prefix prefix <http://....> prefix:name`

- **Literals**
  - "Literal"
  - "Literal"@language
  - """Long literal with newlines""

- **Datatyped Literals**
  - "lexical form"^^datatype URI
    - "10"^^xsd:integer
    - "2006-09-04"^^xsd:date
Turtle. Triples and abbreviations

- Triples separated by .
  

- Common triple predicate and subject:
  
  :a :b :c, :d .
  which is the same as :a :b :c . :a :b :d .

- Common triple subject:
  
  which is the same as: :a :b :c . :a :d :e .

- RDF collections
  
  which is short for many triples
@base <http://www.oeg-upm.net/ontologies/people#>  
@prefix : <http://www.oeg-upm.net/ontologies/people#>  
@prefix person: <http://www.ontologies.org/ontologies/people#>  
:Asun person:hasColleague :Raul ;  
  person:hasHomePage "http://www.fi.upm.es/".  
:Oscar person:hasColleague :Asun ;  
  person:hasName "Óscar Corcho García"."
- Overview
- **RDF**
  - Introduction
  - RDF components
  - Serializing RDF
  - Hands-on
- **RDF Schema**
  - Introduction
  - Main RDF-S components
  - Serializing RDF-S
  - Hands-on
  - RDF(S) Limitations
  - Hands-on
- **RDF(S) management APIs**
• Objective
  • Get used to the different syntaxes of RDF
• Tasks
  • Take the text of an RDF file and create its corresponding graph
  • Take an RDF graph and create its corresponding Turtle files
Exercise 1.a. Create a graph from a file

- Open the file lab-1.ttl
- Create the corresponding graph from it
- Compare your graph with those of your colleagues
Exercise 1.a. StickyNote.ttl
Exercise 1.b. Create files from a graph

- Transform the following graph into the Turtle syntax

```
Sensor029
  includes
  Class01

Sensor029
  hasMeasurement
  Measurement8401

Measurement8401
  hasTemperature
  29
  atTime
  2010-06-12T12:00:12

Computer101
  includes
  Class01

Computer101
  hasOwner
  User10A

User10A
  hasName
  Pedro
```
- Overview
- RDF
  - Introduction
  - RDF components
  - Serializing RDF
  - Hands-on

- RDF Schema
  - Introduction
  - Main RDF-S components
  - Serializing RDF-S
  - Hands-on
  - RDF(S) Limitations
  - Hands-on

- RDF(S) management APIs
- Extends RDF
- Allows describing classes of resources and their properties
- Adds constraints on models

![RDF Schema Diagram]
Overview
RDF
- Introduction
- RDF components
- Serializing RDF
- Hands-on

RDF Schema
- Introduction
- Main RDF-S components
- Serializing RDF-S
- Hands-on
- RDF(S) Limitations
- Hands-on

RDF(S) management APIs
Describing classes

RDF(S)

Person

Data

“Óscar Corcho García”

person:hasName

http://www.oeg-upm.net/

person:hasHomePage

person:hasColleague

data:Oscar

data:Asun

data:Raul

person:hasColleague

person:Person

person:Professor

person:FullProfessor

person:AssociateProfessor

person:InterimAssociateProfessor

rdfs:subClassOf

rdfs:subClassOf

rdfs:subClassOf

rdfs:Class

rdf:type
Describing individuals

RDF(S)

Person

Data

“Óscar Corcho García”

http://www.oeg-upm.net/

person:hasName

person:hasColleague

person:hasHomePage
Describing properties

RDF(S)

Person

Rdfs:Class

Rdfs:Literal

Person:hasName

Person:Person

Person:hasColleague

Person:Professor

Person:hasHomePage

Person:FullProfessor

Person:AssociateProfessor

Person:InterimAssociateProfessor

Data

data:Oscar

“Óscar Corcho García”

data:Asun

http://www.oeg-upm.net/

data:Raul

Person:hasColleague

Person:hasName

Person:hasHomePage
Annotating resources

data:Raul

- rdfs:label: “Raúl García”
- rdfs:label: “Raúl”
- rdfs:comment: “Resource for Raúl García Castro”
- rdfs:seeAlso: http://delicias.dia.fi.upm.es/~rgarcia/
- rdfs:isDefinedBy: http://delicias.dia.fi.upm.es/~rgarcia/foaf.rdf
## RDF-S vocabulary summary

<table>
<thead>
<tr>
<th>Classes</th>
<th>Properties</th>
<th>Individuals</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Classification</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rdfs:Resource</td>
<td>rdfs:subClassOf</td>
<td></td>
</tr>
<tr>
<td>rdfs:Class</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Properties</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rdfs:domain</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rdfs:range</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rdfs:subPropertyOf</td>
<td></td>
</tr>
<tr>
<td><strong>Datatypes</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rdfs:Literal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rdfs:Datatype</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Containers</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rdfs:Container</td>
<td>rdfs:member</td>
<td></td>
</tr>
<tr>
<td>rdfs:ContainerMembershipProperty</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Annotation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rdfs:label</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rdfs:comment</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rdfs:seeAlso</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rdfs:isDefinedBy</td>
<td></td>
</tr>
</tbody>
</table>
Overview

RDF
- Introduction
- RDF components
- Serializing RDF
- Hands-on

RDF Schema
- Introduction
- Main RDF-S components
- **Serializing RDF-S**
- Hands-on
- RDF(S) Limitations
- Hands-on

RDF(S) management APIs
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY person "http://www.oeg-upm.net/ontologies/person#" >
]>

<rdf:RDF xmlns="http://www.oeg-upm.net/ontologies/person#"
  xml:base="http://www.oeg-upm.net/ontologies/person"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:person="http://www.oeg-upm.net/ontologies/person#">
  <rdfs:Property rdf:about="&person;hasColleague">
    <rdfs:range rdf:resource="#Person"/>
    <rdfs:domain rdf:resource="#Person"/>
  </rdfs:Property>
  <rdfs:Property rdf:about="&person;hasHomePage">
    <rdfs:domain rdf:resource="#FullProfessor"/>
  </rdfs:Property>
  <rdfs:Property rdf:about="&person;hasName">
    <rdfs:domain rdf:resource="#Person"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
  </rdfs:Property>
  ...
</rdf:RDF>
...<rdfs:Class rdf:about="#AssociateProfessor">
    <rdfs:subClassOf rdf:resource="#Professor"/>
</rdfs:Class>

<rdfs:Class rdf:about="#FullProfessor">
    <rdfs:subClassOf rdf:resource="#Professor"/>
</rdfs:Class>

<rdfs:Class rdf:about="#InterimAssociateProfessor">
    <rdfs:subClassOf rdf:resource="#AssociateProfessor"/>
</rdfs:Class>

<rdfs:Class rdf:about="#Person"/>

<rdfs:Class rdf:about="#Professor">
    <rdfs:subClassOf rdf:resource="#Person"/>
</rdfs:Class>

<FullProfessor rdf:about="#Asun">
    <person:hasHomePage>http://www.fi.upm.es</person:hasHomePage>
    <person:hasColleague rdf:resource="#Raul"/>
</FullProfessor>

<AssociateProfessor rdf:about="#Oscar">
    <person:hasName>Oscar Corcho García</person:hasName>
    <person:hasColleague rdf:resource="#Asun"/>
</AssociateProfessor>

<InterimAssociateProfessor rdf:about="#Raul"/>
</rdf:RDF>
@base <http://www.oeg-upm.net/ontologies/person#> .
@prefix : <http://www.oeg-upm.net/ontologies/person#> .
@prefix person: <http://www.oeg-upm.net/ontologies/person#> .

:hasColleague a rdfs:Property ;
    rdfs:domain :Person ;
    rdfs:range :Person .

:hasHomePage a rdfs:Property ;
    rdfs:domain :FullProfessor .

:hasName a rdfs:Property ;
    rdfs:domain :Person ;
    rdfs:range rdfs:Literal .

...
... 
:Person a rdfs:Class .

:Professor a rdfs:Class ;
    rdfs:subClassOf :Person .

:FullProfessor a rdfs:Class ;
    rdfs:subClassOf :Professor .

:AssociateProfessor a rdfs:Class ;
    rdfs:subClassOf :Professor .

:InterimAssociateProfessor a rdfs:Class ;
    rdfs:subClassOf :AssociateProfessor .

:Asun a :FullProfessor ;
    :hasHomePage "http://www.fi.upm.es" ;
    :hasColleague :Raul .

:Oscar a :AssociateProfessor ;
    :hasName "Oscar Corcho García" ;
    :hasColleague :Asun .

:Raul a :InterimAssociateProfessor .

a is equivalent to rdf:type
[{
    "@id": "http://www.oeg-upm.net/ontologies/person#hasColleague",
    "@type": [
        "http://www.w3.org/2000/01/rdf-schema#Property"
    ],
    "http://www.w3.org/2000/01/rdf-schema#domain": [{
        "@id": "http://www.oeg-upm.net/ontologies/person#Person"
    }],
    "http://www.w3.org/2000/01/rdf-schema#range": [{
        "@id": "http://www.oeg-upm.net/ontologies/person#Person"
    }]
},
{
    "@id": "http://www.oeg-upm.net/ontologies/person#hasName",
    "@type": [
        "http://www.w3.org/2000/01/rdf-schema#Property"
    ],
    "http://www.w3.org/2000/01/rdf-schema#domain": [{
        "@id": "http://www.oeg-upm.net/ontologies/person#Person"
    }],
    "http://www.w3.org/2000/01/rdf-schema#range": [{
        "@id": "http://www.w3.org/2000/01/rdf-schema#Literal"
    }]
},
{
    "@id": "http://www.oeg-upm.net/ontologies/person#hasHomePage",
    "@type": [
        "http://www.w3.org/2000/01/rdf-schema#Property"
    ],
    "http://www.w3.org/2000/01/rdf-schema#domain": [{
        "@id": "http://www.oeg-upm.net/ontologies/person#FullProfessor"
    }]
}]

Overview

RDF
- Introduction
- RDF components
- Serializing RDF
- Hands-on

RDF Schema
- Introduction
- Main RDF-S components
- Serializing RDF-S
- Hands-on
- RDF(S) Limitations
- Hands-on

RDF(S) management APIs
Exercise

• Objective
  • Get used to the different syntaxes of RDF(S)

• Tasks
  • Take the text of an RDF(S) file and create its corresponding graph
  • Take an RDF(S) graph and create its corresponding Turtle files
Exercise 2.a. Create a graph from a file

- Open the file lab-1-rdfs.ttl
- Create the corresponding graph from them
- Compare your graph with those of your colleagues
Exercise 2.a. StickyNote-rdfs.ttl
Exercise 2.b. Create files from a graph

- Transform the following graph into the turtle syntax.
- Overview
- RDF
  - Introduction
  - RDF components
  - Serializing RDF
  - Hands-on
- RDF Schema
  - Introduction
  - Main RDF-S components
  - Serializing RDF-S
  - Hands-on
  - RDF(S) Limitations
  - Hands-on
- RDF(S) management APIs
### RDF(S) limitations

- **RDFS too weak to describe resources in sufficient detail**
  - No localised range and domain constraints
    - Can’t say that the range of hasChild is person when applied to persons and elephant when applied to elephants
  - No existence/cardinality constraints
    - Can’t say that all instances of person have a mother that is also a person, or that persons have exactly 2 parents
  - No boolean operators
    - Can’t say or, not, etc.
  - No transitive, inverse or symmetrical properties
    - Can’t say that isPartOf is a transitive property, that hasPart is the inverse of isPartOf or that touches is symmetrical

- Difficult to provide reasoning support
  - No “native” reasoners for non-standard semantics
  - May be possible to reason via FOL axiomatisation
- Overview
- RDF
  - Introduction
  - RDF components
  - Serializing RDF
  - Hands-on
- RDF Schema
  - Introduction
  - Main RDF-S components
  - Serializing RDF-S
  - Hands-on
  - RDF(S) Limitations
  - Hands-on
- RDF(S) management APIs
• **Objective**
  - Understand the features of RDF(S) for implementing ontologies, including its limitations

• **Tasks**
  - From a domain description, create the RDF(S) graph
    - First only include the vocabulary from the domain
    - Then include references to the RDF and RDFS vocabularies
Exercise 3. Domain description

- A certain location can be a location of interest.
- Locations of interest can be touristic locations or establishments, but not both at the same time.
- Touristic locations can be palaces, churches, chapels, and cathedrals.
- Establishments can be hotels, hostels, or guest houses.
- A location is placed in a locality, which can be in turn a town, a village, or a city.
- A location of interest has some postal address that includes the street name and number.
- Localities have a certain number of inhabitants.
- Localities belong to a certain province.

- Covarrubias is a village with 634 inhabitants in the province of Burgos.
- The “El Galo” restaurant is placed in Covarrubias, in Mayor street, number 5.
- One of Covarrubias’ churches is at Santo Tomas street.
Exercise 3. Sample resulting ontology
- Overview
- RDF
  - Introduction
  - RDF components
  - Serializing RDF
  - Hands-on
- RDF Schema
  - Introduction
  - Main RDF-S components
  - Serializing RDF-S
  - Hands-on
  - RDF(S) Limitations
  - Hands-on
- RDF(S) management APIs
- Overview
- RDF
  - Introduction
  - RDF components
  - Serializing RDF
  - Hands-on
- RDF Schema
  - Introduction
  - Main RDF-S components
  - Serializing RDF-S
  - Hands-on
  - RDF(S) Limitations
  - Hands-on
- RDF(S) management APIs
Sample RDF APIs

- RDF libraries for different languages:
  - Java, Python, C, C++, C#, .Net, Javascript, Tcl/Tk, PHP, Lisp, Obj-C, Prolog, Perl, Ruby, Haskell
  - List in [http://esw.w3.org/topic/SemanticWebTools](http://esw.w3.org/topic/SemanticWebTools)

- Usually related to a RDF repository

- Multilanguage:
  - Redland RDF Application Framework (C, Perl, PHP, Python and Ruby):
    [http://www.redland.opensource.ac.uk/](http://www.redland.opensource.ac.uk/)

- Java:

- PHP:

- Python:
  - RDFLib: [http://rdflib.net/](http://rdflib.net/)
  - Pyrple: [http://infomesh.net/pyrple/](http://infomesh.net/pyrple/)
- Java framework for building Semantic Web applications
- Open source software from HP Labs
- The Jena framework includes:
  - A RDF API
  - An OWL API
  - Reading and writing RDF in RDF/XML, N3 and N-Triples
  - In-memory and persistent storage
  - A rule based inference engine
  - SPARQL query engine
A framework for storage, querying and inferencing of RDF and RDF Schema

A Java Library for handling RDF

A Database Server for (remote) access to repositories of RDF data

Highly expressive query and transformation languages
  - SeRQL, SPARQL

Various backends
  - Native Store
  - RDBMS (MySQL, Oracle 10, DB2, PostgreSQL)
  - Main memory

Reasoning support
  - RDF Schema reasoner
  - OWL DLP (OWLIM)
  - Domain reasoning (custom rule engine)
// some definitions
String personURI = "http://somewhere/JohnSmith";
String givenName = "John";
String familyName = "Smith";
String fullName = givenName + " " + familyName;
// create an empty
Model model = ModelFactory.createDefaultModel();
// create the resource
// and add the properties cascading style
Resource johnSmith = model.createResource(personURI)
    .addProperty(VCARD.FN, fullName)
    .addProperty(VCARD.N, model.createResource())
    .addProperty(VCARD.Given, givenName)
    .addProperty(VCARD.Family, familyName);
// create an empty model
Model model = ModelFactory.createDefaultModel();

// use the FileManager to find the input file
InputStream in = FileManager.get().open(inputFileName);
if (in == null) {
    throw new IllegalArgumentException("File not found");
}

// read the RDF/XML file
model.read(in, ":");

// write it to standard out
model.write(System.out);

<!-- RDF/XML content -->

```xml
<rdf:RDF
    xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
    xmlns:vcard='http://www.w3.org/2001/vcard-rdf/3.0#'>
    <rdf:Description rdf:nodeID="A0">
        <vcard:Family>Smith</vcard:Family>
        <vcard:Given>John</vcard:Given>
    </rdf:Description>
    <rdf:Description rdf:about='http://somewhere/JohnSmith/'>
        <vcard:FN>John Smith</vcard:FN>
    </rdf:Description>
    ...
</rdf:RDF>
```
Some RDF editors

- **IsaViz**
- **Morla**
  - [http://www.morlardf.net/](http://www.morlardf.net/)
- **RDFAuthor**
  - [http://rdfweb.org/people/damian/RDFAuthor/](http://rdfweb.org/people/damian/RDFAuthor/)
- **RdfGravity**
  - [http://semweb.salzburgresearch.at/apps/rdf-gravity/index.html](http://semweb.salzburgresearch.at/apps/rdf-gravity/index.html)
- **Rhodonite**
  - [http://rhodonite.angelite.nl/](http://rhodonite.angelite.nl/)
Main References

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

  
  http://www.w3.org/TR/REC-rdf-syntax/

- RDF validator:
  
  http://www.w3.org/RDF/Validator/

- RDF resources:
  
  http://planetrdf.com/guide/
RDF and RDF Schema

Raúl García Castro, Óscar Corcho

Boris Villazón-Terrazas
bvillazon@isoco.com
@boricles

Slides available at: http://www.slideshare.net/boricles/