

 **Università degli Studi di Bari**  *Laboratorio di Acquisizione
della Conoscenza e
Apprendimento delle Macchine*




Tutorial
Dipartimento di Informatica
Università di Bari

 **RDF Tutorial** 

Credits

- Pasquale Lops e Giovanni Semeraro
- Daniele Capursi e Luigi Iannone
- Eufemia Tinelli

2


 **RDF Tutorial** 

Overview

- What is RDF ?
- Resources
- RDF Data Model
- RDF Containers
- RDF Schema
- RDF-based framework

3

RDF Tutorial




The Web and the Semantic Web...

- The World Wide Web was invented by Tim Berners-Lee in the 1989 allowing CERN scientists to share scientific documents across the World.
- Nowadays the Web is the largest collection of document ever produced by the Humanity.
- Web should allow (says Berners – Lee)
 - to share information between men,
 - to accomplish tasks using the Web

4

RDF Tutorial




What's wrong with the Web...

- The Web is big
- The Web contains information human tailored
 - Machine cannot understand the meaning of a document
 - Only syntactic techniques can be exploited (keywords as such)

5

RDF Tutorial



...What's wrong with the Web...


- Polisemy
 - A term can have multiple meanings

↓

Low precision
- Synonymy
 - Several terms having the same meaning



↓

Low recall



6

RDF Tutorial





...What's wrong with the Web

- How to integrate the information from several sources?
- How to perform task integrating several services available on the Web? E.g.:
 - "I want to book a ticket for "Matrix Reloaded" and a dinner at an Italian Restaurant within five blocks from the cinema".
- So I need computers understanding information reasoning about it (Computers dealing with Knowledge).

7

RDF Tutorial





The Semantic Web...

- The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web.
- The Semantic Web turns the Web into a huge Knowledge Base.
- The Web is a network of information linked each other, to be consumed by Humans.
- The Semantic Web is a network of information semantically linked, to be consumed by computers.

8

RDF Tutorial



...The Semantic Web

- The Web already offers the infrastructure to publish information that everyone can reach and to integrate her information with the one provided by others (via the URL and links mechanisms).
- We need a framework that permits to express knowledge in a way that can be machine-processable.

9

RDF Tutorial

What we need to share Knowledge on the Web

- We need a language that:
 - Has a syntax that ensures syntactical interoperability
 - Has a semantics that is shared between computers
 - Is able to express ontologies
 - Permits very efficient elaborations, since the knowledge in the Web is very big
- We need an infrastructure that:
 - Permits to add semantics to the existing documents
 - Permits to refer to things unambiguously on the entire Web
 - Permits to integrate several source of knowledge in the distribute environment which is the Web

10

RDF Tutorial

Interoperability levels needed to the Semantic Web

- **Syntax** → requires a syntactic structure for describing and interchanging data (**XML**)
- **Semantics** → requires mapping between terms, which in turn requires content analysis (**RDF&RDFS,OWL**)
- **Logic** → requires formal and explicit specifications of domain models to represent and reason about knowledge (**Description Logics**)



11

RDF Tutorial

Structure of the Semantic Web

12

RDF Tutorial





A possible solution: RDF

- RDF principles:
 1. "Everything must be identified by URIs".
 2. The "Least Power" principle:
"When expressing anything, use the least expressive language".
 3. "Anything can say anything about anything".

13

RDF Tutorial





What is RDF?

- Resource Description Framework (RDF) is a simple framework for the description of Knowledge in the World Wide Web, enabling the encoding, exchange, and reuse of structured information.
- It was specifically designed for the description of "metadata" (data about data) related to resources available through the World Wide Web.
- It is a W3C recommendation.
- It is at the base of what is called "the Semantic Web", as it permits to share knowledge on the Web.
- Latest version drafts released on 22 February 1999

14

RDF Tutorial



RDF consists of...


- Model and Syntax Specification (22 Febbraio 1999) - model for representing RDF metadata as well as a syntax for encoding and transporting this metadata
- RDF/XML Syntax Specification (Revised - 25 Marzo 2005) - defines an XML syntax for RDF as amended and clarified by RDFCore from that originally described in the 'RDF Model & Syntax Document'
- RDF Schema (27 Marzo 2000) - describes how to use RDF to describe RDF vocabularies.


15

RDF Tutorial

Resources

- All things described by RDF expressions are called *resources*.
- Example:
 - Web pages
 - Collections of pages
 - Web sites
 - Books
 - People
 - Feelings






 not directly accessible via the Web


16

RDF Tutorial

Resources and URIs...

- RDF define a *Resource* anything that can be identified by an URI (Universal Resource Identifier).
- A URI is a name that refers to a unique object in the World.
- A URI can be a URN (Universal Resource Name) or a URL (Universal Resource Locator).
- URLs are URIs that can be used to retrieve the resource they refers to.

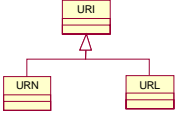




17

RDF Tutorial

...Resources and URIs...





```

classDiagram
    class URI
    class URN
    class URL
    URI <|-- URN
    URI <|-- URL
  
```

urn:IT_CPRDLG55D02A669S urn:...	http://digilander.iol.it/daniele ftp://ftp.microsoft.com mailto:semeraro@di.uniba.it http://www.daniele.com/amici#Luigi
------------------------------------	--

18

RDF Tutorial






...Resources and URIs


- Using URLs as URIs, it's easy to ensure uniqueness of the name
- Only the IETF can assign top level domains (e.g. .it, .com, .org, .gov)
- Only national organizations can assign secondary level domains (e.g. libero.it, tin.it, gazzetta.it)
- Only the owner of a domain can assign sub-domains and URLs of that domain (e.g. http://www.libero.it, http://www.gazzetta.it/fantacalcio, http://digilander.iol.it/daniele/amici#Roberto)

19

RDF Tutorial



RDF data model



- Resources have associated
 - properties that have associated values
- Resources: identified by URIs
- Properties: identified by URIs
- Values: resources or primitive data types (string, integer, date,...)

20

RDF Tutorial





RDF Statements: the triples

- The basic unit of information in RDF is the *Statement*:
 - an assertion that states that a certain resource (the *Subject*) has a certain value (the *Object*) for a certain property (the *Predicate*).
- So a statement is actually a *Triple* (Subject, Predicate, Object).
- A set of statements is called *Model*.

21

RDF Tutorial




Finally some examples!!!

- “The author of the RDF tutorial is Daniele Capursi”
- “The first name of Daniele Capursi is ‘Daniele’”

Subject	Predicate	Object
http://www.di.uniba.it/~gci/RDF_Tutorial.ppt	http://purl.org/dc/elements/1.1/creator	mailto:capursi@sword.it
mailto:capursi@sword.it	http://www.myspace.com/properties#firstName	Daniele

22

RDF Tutorial



RDF data model: directed labelled graph...

- One or more RDF Statements can be displayed as a very simple directed labelled graph:

Resource

Property →

Literal Value


Resource

Property →

Resource Value

23

RDF Tutorial



...RDF data model: directed labelled graph...

- The previous examples:

http://www.di.uniba.it/~gci/RDF_Tutorial.ppt

<http://purl.org/dc/elements/1.1/creator> →

<mailto:capursi@sword.it>

<mailto:capursi@sword.it>

<http://www.myspace.com/properties#firstName> →

Daniele

24

RDF Tutorial

...RDF data model: directed labelled graph

- This permits to display the semantic relationships in a compact way:

25

RDF Tutorial

Interchange syntax

- A serialization syntax is needed:
 - To save RDF Statements into a persistent form (e.g. a file)
 - To transfer them from one application to another
 - To permit their retrieval from the Web
- RDF Model & Syntax specification defines a **standard** XML encoding of the data model.
- Why XML encoding?
 - Parsers are easy to implement (are based on XML parsers)
 - XML deals with interoperability and internationalization issues

26


RDF Tutorial

The example encoded in XML

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.0/"
  xmlns:NS0="http://www.myspace.com/properties#"
  >
  <rdf:Description rdf:about="http://www.di.uniba.it/~gci/RDFTutorial.ppt">
    <dc:creator rdf:resource="mailto:capursi@sword.it"/>
  </rdf:Description>
  <rdf:Description rdf:about="mailto:capursi@sword.it">
    <NS0:firstName>Daniele</NS0:firstName>
    <NS0:age>25</NS0:age>
  </rdf:Description>
</rdf:RDF>
```

27

RDF Tutorial



The example encoded in Abbreviated XML


```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.0/"
  xmlns:NS0="http://www.myspace.com/properties#"
  >
  <rdf:Description rdf:about="http://www.di.uniba.it/~gci/RDF Tutorial.ppt">
    <dc:creator rdf:resource="mailto:capursi@sword.it"
      NS0:firstName="Daniele"
      NS0:age="25"/>
  </rdf:Description>
</rdf:RDF>

```

28

RDF Tutorial



Other non-standard syntax encoding

- Other syntax encoding exist
 - N3 (Notation 3)
 - N-TRIPLE

The example encoded in N-TRIPLE


```

<http://www.di.uniba.it/~gci/RDF Tutorial.ppt>
  <http://purl.org/dc/elements/1.0/creator> <mailto:capursi@sword.it> .
<mailto:capursi@sword.it> <http://www.myspace.com/properties#firstName>
  "Daniele" .
<mailto:capursi@sword.it> <http://www.myspace.com/properties#age>
  "25" .

```

29

RDF Tutorial



RDF IS NOT XML!!!

- XML is just one of the possible way to serialize RDF.
- XML is the standard way to encode RDF for putting it into the Semantic Web, not to process it.
- There are more than one way to serialize RDF in XML.
- If a program manipulates RDF as XML, the code must be repeated for every possible XML serialization!
- Manipulating RDF as XML, we break the abstraction of RDF data model.
- RDF/XML is not easy to read for Humans.

30

RDF Tutorial

More RDF features: blank nodes

"The girl with red hair is 23 years old"
What is the URI of the girl?

The diagram shows a central blank node (white oval) with two outgoing edges. The top edge is labeled with the URI `http://myspace.com/properties#hairColor` and points to a light blue oval containing the URI `urn:colors/red`. The bottom edge is labeled with the URI `http://www.myspace.com/properties#age` and points to a light blue rectangle containing the number `23`.

31

RDF Tutorial

More RDF features: RDF Containers

- Bag
 - unordered list of resources or literals
- Sequence
 - ordered list of resources or literals
- Alternatives
 - alternate values
 - need to choose
 - at least one value
 - first value is default or preferred value

32

RDF Tutorial

Collection example in RDF

```

<rdf:Description about="http://concerto.pira.co.uk">
  <concerto:FinalUsers>
    <rdf:Bag>
      <rdf:li>PIRA International</rdf:li>
      <rdf:li>Assett Biovista</rdf:li>
    </rdf:Bag>
  </concerto:FinalUsers>
</rdf:Description>

```

The diagram shows a central blank node (white oval) with four outgoing edges. The top edge is labeled `concerto:FinalUsers` and points to a light blue oval containing the URI `http://concerto.pira.co.uk`. The bottom-left edge is labeled `rdf:type` and points to a light blue oval containing `rdf:Bag`. The bottom-middle edge is labeled `rdf:_1` and points to a light blue rectangle containing `Pira International`. The bottom-right edge is labeled `rdf:_2` and points to a light blue rectangle containing `Assett Biovista`.

33

RDF Tutorial

More RDF features: reification...

- A feature that is rarely used and has a cumbersome syntax: statements about statements.
- A reificated statement is NOT asserted!

E.g. "Daniele says that the donkeys can fly"

34

RDF Tutorial

...More RDF features: reification

35

RDF Tutorial


Reification Properties

- **Rdf:subject** - corresponds to the property called the 'subject' in the formal model for RDF
- **Rdf:predicate** - corresponds to the property called the 'predicate' in the formal model for RDF
- **Rdf:object** - corresponds to the property called the 'object' in the formal model for RDF
- **Rdf:statement** - corresponds to the set called the 'Statement' in the formal model for RDF

A new resource with the above four properties represents the original statement


36

RDF Tutorial




Is RDF good?...

- How does RDF give semantics?
- Why is RDF better than XML?
- XML tags and attributes doesn't have a semantics.
- Everybody can define his own tags or interpret the meaning of a tag in the way he likes.
- XML leaves too much freedom on how to represent information.




37

RDF Tutorial



...Is RDF good?


- RDF syntax is very poor: it's very likely that different people uses the same concepts (Resources and Properties) to describe the same things, apart from the URIs.
- If some people (or some programs) agree on how to assign URIs to some resources, they perfectly share the meaning of that resources (they know what objects correspond to that URIs),
- If they agree on the meaning of some properties, they cant perfectly understand what they represent.



Still not enough


38

RDF Tutorial




So, where is semantics?

- RDF has been designed to be used together with shared vocabularies.
- A Vocabulary is a set of Properties that have a precise meaning.
- There are lots of standard vocabularies for many domains.
- If we use only standard vocabularies, we share the semantics of our RDF statements.
- What if we want to define our own vocabularies?




39

RDF Tutorial



“Dublin Core Metadata Initiative” Vocabulary




- “Dublin Core” is a standard for the representation of essential metadata of Web documents
- Dublin Core 1.0 properties:
“<http://purl.org/dc/elements/1.0/>” + ...


Content	Intellectual Property Rights	Instance
Title	Creator	Date
Subject	Publisher	Format
Description	Contributor	Identifier
Type	Rights	Language
Source		
Relation		
Coverage		

40

RDF Tutorial




How to define a vocabulary: RDF Schema...




- RDF allows people to define new vocabularies (called “Schema”).
- RDFS (RDF Schema) is a language, based on RDF itself, to describe RDF vocabularies. An instance of RDFS is called “RDF Schema” too!
- A *Schema* is a set of *Classes* of resources and *Properties* with a precise semantics.
- A schema can define new elements from scratch or by “importing” concepts from other schemas and extending them.

41

RDF Tutorial



...RDF Schema...



RDFS is a lightweight ontology language, since it allows to define Classes and Property of classes:

- RDFS allows to define Classes:
 - From scratch
 - As subclasses of existing ones
 - As equivalent to existing ones (latest RDF version)
- RDFS allows to define Properties:
 - From scratch
 - As sub-properties of existing ones
 - As equivalent to existing ones (latest RDF version)
- RDFS is domain independent

42

RDF Tutorial

RDFS vocabulary

The diagram illustrates the RDFS vocabulary structure. It is organized into three main categories: **Resource**, **Class**, and **Property**.
 - **Resource** includes: `rdfs:Resource`, `rdfs:Class`, `rdfs:Property`, `rdfs:ConstraintProperty`, and `rdfs:Literal`.
 - **Class** includes: `rdfs:Resource`, `rdfs:Class`, `rdfs:Property`, and `rdfs:ConstraintProperty`.
 - **Property** includes: `rd:type`, `rdfs:subClassOf`, `rdfs:subPropertyOf`, `rdfs:comment`, `rdfs:label`, `rdfs:seeAlso`, and `rdfs:isDefinedBy`.
 - **ConstraintProperty** includes: `rdfs:range` and `rdfs:domain`.

43

RDF Tutorial

RDFS – Classes

- **rdfs:Class** - is the class of resources that are RDF classes, `rdfs:Class` is an instance of `rdfs:Class`.
- **rdfs:Resource** - all things described by RDF are called *resources*, and are instances of the class `rdfs:Resource`.
- **rdfs:Literal** - is the class of literal values such as strings and integers.
- **rdfs:Property** - is the class of RDF properties, `rdfs:Property` is an instance of `rdfs:Class`.

44

RDF Tutorial

RDFS – Properties

- **rd:type** - is an instance of `rdfs:Property` that is used to state that a resource is an instance of a class.
- **rdfs:subClassOf** - is an instance of `rdfs:Property` that is used to state that all the instances of one class are instances of another.
- **rdfs:subPropertyOf** - is an instance of `rdfs:Property` that is used to state that all resources related by one property are also related by another.
- **rdfs:seeAlso** - `rdfs:seeAlso` is an instance of `rdfs:Property` that is used to indicate a resource that might provide additional information about the subject resource.
- **rdfs:label** - is used to provide a human-readable version of a resource name.

45

RDF Tutorial

...RDF Schema...

- RDFS allows to specify properties of properties:
 - Domain
I.e. the classes it applies to (the classes of the resources that can be the subject of a statement with that property as predicate)
 - Range
I.e. the classes of the resources that can be value of that property (the classes of the resources that can be the object of a statement with that property as predicate)

46

RDF Tutorial

RDF Schema and RDF

- An RDF Schema is written in RDF itself.
- Obviously using properties and classes whose semantics is standard.

==>

- There is an RDF Schema that describes the elements of the RDF Schema language itself!
- So RDFS can be described in RDFS, that is still RDF.


47

RDF Tutorial

RDFS – Data Model

48

RDF Tutorial



RDFS example class encoded in XML


```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303">

  <rdf:Description ID="Person">
    <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource"/>
  </rdf:Description>
  .....
```

49

RDF Tutorial




RDFS example property encoded in XML

```

.....
<rdf:Description ID="firstName">
  <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Property"/>
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="http://www.w3.org/TR/xmlschema-2/string"/>
</rdf:Description>
.....
</rdf:RDF>
```

50

RDF Tutorial



RDFS: what for?

- An RDF Schema can be used to (semantically!) validate an RDF Model, in the same way we can validate a knowledge base against an ontology
- RDFS semantics defines some simple inferences, typical of an ontology language, e.g.:
 - If *r* is of class *A* and class *A* is subclass of class *B*, then *r* is of class *B*
 - If *A* is subclass of *B* and *B* is subclass of *C* then *A* is subclass of *C*

↓

Transitivity of subClassOf and subPropertyOf

51

RDF Tutorial




RDFS makes RDF smarter...


- The possibility to reuse and extend existing schemas + such inferences makes RDF agents much smarter:
 - If an agent encounters a property it doesn't know, but it knows one super-property of such property, he can perform meaningful processing as well

52

RDF Tutorial




Some comments about RDFS



- RDFS is "property-centric" VS OO is "class-centric":
 - RDFS defines a property in terms of what classes can have that property and what classes can be objects of that property
 - Object Oriented defines classes in terms of property they own and of that type are the values of these properties
- Everybody can reference and reuse a Class or a Property defined by others, just using its URI. Everybody can reference a Resource defined by others.

53

RDF Tutorial





Some comments about RDFS

- RDFS is "property-centric" VS OO is "class-centric":
 - RDFS defines a property in terms of what classes can have that property and what classes can be objects of that property
 - Object Oriented defines classes in terms of property they own and of that type are the values of these properties
- Everybody can reference and reuse a Class or a Property defined by others, just using its URI. Everybody can reference a Resource defined by others.

54



RDF Tutorial



RDF Core: A component for effective management of RDF Models

55

RDF Tutorial





Overview

- RDFCore functionalities
- RDFCore architecture
- RDFCore empirical evaluation
- Conclusions and future work

56

RDF Tutorial



RDFCore functionalities

- Support for all typical operations on RDF at two level of granularity:
 - Model (Description)
 - Statement (Triple)
- Multi-user environment
- Support for multiple query languages
 - RDQL
 - SquishQL
 - RQL
 - SPARQL

57

RDF Tutorial

RDFCore architecture...

```

    graph TD
      DM[Description Manager] <--> Q[Query Engine]
      TM[Triple Manager] <--> Q
      Q <--> RE[RDF Engine]
      RE <--> RP[RDF Persistence]
      RP <--> P[PDOM File System]
      DM <--> RE
      TM <--> RE
  
```

58

RDF Tutorial

... RDFCore architecture

Strategy Pattern

- Compliance with W3C Object Model
- Pluggable modules for:
 - Persistence
 - Query engines

59


RDF Tutorial

RDF Persistence implementations

- Currently available:
 - File system XML based (relying on PDOM) used as baseline
 - Jena 2 based (more efficient especially in queries)
 - Plain File system (not relying on PDOM)

60

RDF Tutorial




Sneak Comparison

	RDF Suite	Jena Semantic Web Toolkit	RDFCore
Statements Granularity API	n	y	y
Model Granularity API	n	y	y
Multi-user complex support	n	n	y
RDQL	n	y	y
RQL	y	n	n
SquishQL	n	n	y
Scales up to (no. of Triples)	6000000*	800000*	100000

* = Taken from W3C surveys

61

RDF Tutorial




Conclusions

We achieved:

- **A standard framework that allows all typical operations on RDF (Models and Statements)**
- **Flexible architecture with the possibility of plugging in new modules seamlessly**
- **Linear scalability for the basic operations and queries even in this early prototype**

62

RDF Tutorial



Future work

- **New persistence layers (e.g.: RSSDB based)**
- **More extensive tests and comparisons with existing systems**
- **Integration of RDFCore within wider architectures**

63



Further Readings



<http://www.w3.org/RDF>

Latest version drafts:

<http://www.w3.org/TR/2002/WD-rdf-syntax-grammar-20021108/>
