Chapter 3

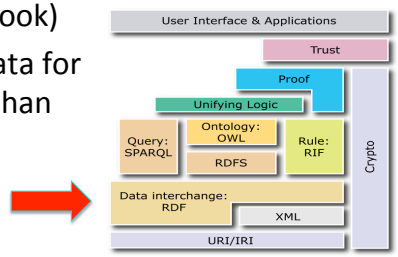
RDF Syntax 1

Topics

- Basic concepts of RDF
  - resources, properties, values, statements, triples
  - URIs and URIrefs
  - RDF graphs
  - Literals and Qnames
- Vocabularies and modeling
  - URIrefs as vocabularies
  - Data modeling IN RDF
- Serialization of RDF graphs: XML, Turtle, …

What is RDF?

- A data model for representing information (esp. **metadata**) about **resources** in the Web
- Can represent information about things that can be **identified** on the Web, even when not **retrievable** (e.g., a book)
- Usecases: provide data for **applications** rather than directly to people



RDF Basics

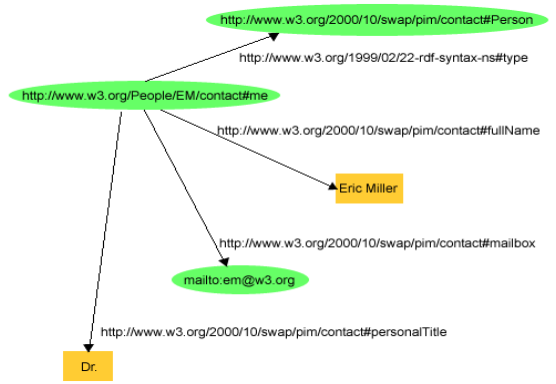
- Core idea: identify resources using **Web identifiers** and describing resources in terms of simple **properties** and property **values**
- To identify resources, RDF uses **Uniform Resource Identifiers (URIs**) and **URI references (URIrefs).**
- **Definition:** A **resource** is anything that is identifiable by a URIref

Example

Consider the following information:

"there is a Person identified by `http://www.w3.org/People/EM/contact#me`, whose name is Eric Miller, whose email address is em@w3.org, and whose title is Dr."

Example (cont'd)



Basics

The resources being described have properties which have values, and that resources can be described by making statements that specify those properties and values

- The part that identifies the thing the statement is about is the **subject**
- The part that identifies the property of the subject the statement specifies is the **predicate**
- The part that identifies the property's value is the **object**

Example

http://www.example.org/index.html has a creator whose value is "John Smith"

- The **subject** is the URL http://www.example.org/index.html
- The **predicate** is the word "creator"
- The **object** is the phrase "John Smith"

RDF Triples

- RDF statements can be written as **triples**
- The simple *ntriples* notation has a set of triples terminated by a period, where URI's are givin inside angle brackets

<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/creator> <http://www.example.org/staffid/85740> .

<http://www.example.org/index.html> <http://www.example.org/terms/creation-date> "August 16, 1999" .

<http://www.example.org/index.html> <http://purl.org/dc/elements/1.1/language> "en" .

URIs and URIREFs

Uniform Resource Identifiers (URIs)

- URIs identify resources on the Web
- Unlike URLs, they aren't limited to identifying things with network locations
- No organization controls who makes URIs or how they can be used
  - Some URI schemes (http: URL's) depend on centralized systems such as DNS
  - Others are **completely decentralized**

URI Reference (URIref)

- A **URIref** is a URI with an optional fragment identifier at the end, e.g:

  http://example.org/index.html#section2
- Fragment usecase:
  - In HTML a # fragment refers to a place in the page
  - In RDF we can use fragments to refer to resources in a RDF graph that the URI denotes, e.g., subjects, predicates or objects
    - http://www.w3.org/2004/02/skos/core : vocabulary for describing topics
    - http://www.w3.org/2004/02/skos/core#broader : the *broader* concept in SKOS Core vocabulary
- Like URLs, URIrefs may be either **absolute** or **relative**
  - Note: the empty URI refers to the resource it's in

URIrefs in RDF (cont'd)

- RDF and Browsers use URIrefs to **identify things**, but interpret URIrefs slightly differently:
  - Browsers also use URIrefs to **retrieve** things
  - RDF uses URIrefs **only** to identify things and these might not even be retrievable
- **Linked Data** best practice is to use HTTP URIs that return RDF data for every URI
  - http://dbpedia.org/page/Alan_Turing
  - curl –I http://dbpedia.org/page/Alan_Turing
  - curl –H "Accept:application/rdf+xml" http://dbpedia.org/page/Alan_Turing

RDF Graphs

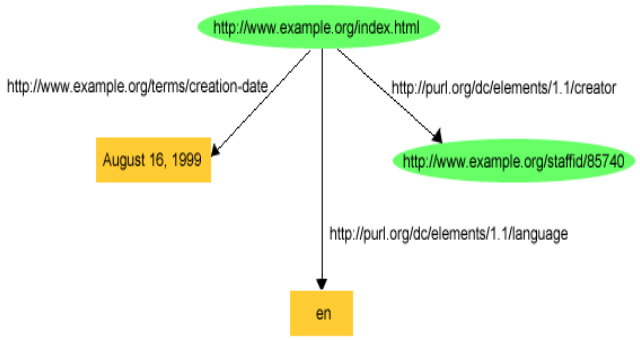
RDF Graphs

- RDF models statements by **nodes** and **arcs** in a **graph**
- A **statement** is represented by a node for the subject, a node for the object and an arc for the predicate (subject => object)
- A **node** may be identified by a **URIref** or it can be a **literal** or a **blank node**
- An **arc** is identified by a **URIref**
- **Note:** We will draw RDF graphs as **directed graphs**
  - But an arc can be the subject of an RDF statement
  - :has_parent owl:inverseOf :has_child

Example

- Consider the following statements:
  - `http://www.example.org/index.html` has a creation-date whose value is August 16, 1999.
  - `http://www.example.org/index.html` has a language whose value is English.

The RDF Graph of the Example

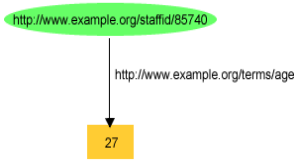


RDF and Related Data Models

- In terms of the **relational model**, an RDF statement is like a **tuple in a relation** *Graph* with columns *Subject, Predicate, Object*
- For **first-order logic**, an RDF statement is like an **atomic formula** *triple(subj, pred, obj)* where *triple* is a FOL predicate and *subj, pred* and *obj* are constants
  - Alternatively: *pred(subj, obj)*

Literals and QNames

Literals



What is 27? Number or string?

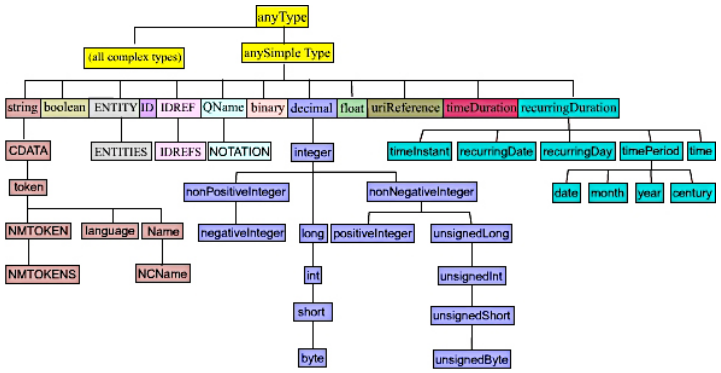
Plain and Typed Literals

- There are two kinds of literals: **plain** and **typed**
- Plain literals have a **lexical form** (their lexical value) and optionally a **language tag**, e.g:
  - "27", "Hello world"@en
- **RDF typed literals** are formed by pairing a string with a URIref for a particular **datatype**, e.g.:
  - "27"^^http://www.w3.org/2001/XMLSchema#integer
  - "27"^^xsd:int

Data Types for Literals

- In practice, the most widely used data typing scheme will be the one by XML Schema
  - But the use of **any** externally defined data typing scheme is allowed in RDF documents
- XML Schema predefines a large range of data types
  - E.g. Booleans, integers, floating-point numbers, times, dates, etc.

XMLSchema Datatypes



http://www.w3.org/TR/xmlschema-2/

Qnames for URIrefs

- The ntriples notation results in very long lines
- We can use an **XML qualified name** (**QName**) w/o brackets for a full URI reference
  - http://dbpedia.org/page/Alan_Turing
  - dbp:Alan_Turing
- A **Qname** has a **prefix** that has been assigned to a **namespace URI**, followed by a **colon**, and then a **local name**.
- The concepts of **names** and **namespaces** used in RDF originate in XML

Topics

- Basic concepts of RDF
  - resources, properties, values, statements, triples
  - URIs and URIrefs
  - RDF graphs
  - Literals and QNames
- Vocabularies and modeling
  - URIrefs as vocabularies
  - Data modeling IN RDF
- Serialization of RDF graphs: XML, Turtle, …