RDF – Resource Description Framework

- Resource Description Framework (RDF) is a “language” for representing information about *resources* in the WWW
- All *resources* are identified by a URI. A resource is anything that has identity. For example, a resource may be an electronic document, an image, a service (e.g., "today's weather report for Los Angeles"), and a collection of other resources. Not all resources are network "retrievable"; e.g., human beings, corporations, and bound books in a library can also be considered resources
- *Resources* are described in terms of simple statements specifying properties and property values
- A statements is: (each component is indicated by a URI)
  - A subject
  - A predicate (about the subject)
  - An object (the value of the predicate)

Simple RDF statement

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

- the subject is the following URI **`http://www.example.org/index.html`**
- the predicate is the phrase “has a creator"
- the object is the phrase "John Smith“

- To avoid “misunderstandings”, all three should be indicated by URIs
  - Subject http://www.example.org/index.html
  - Predicate http://purl.org/dc/elements/1.1/creator
  - Object http://www.example.org/staffid/85740

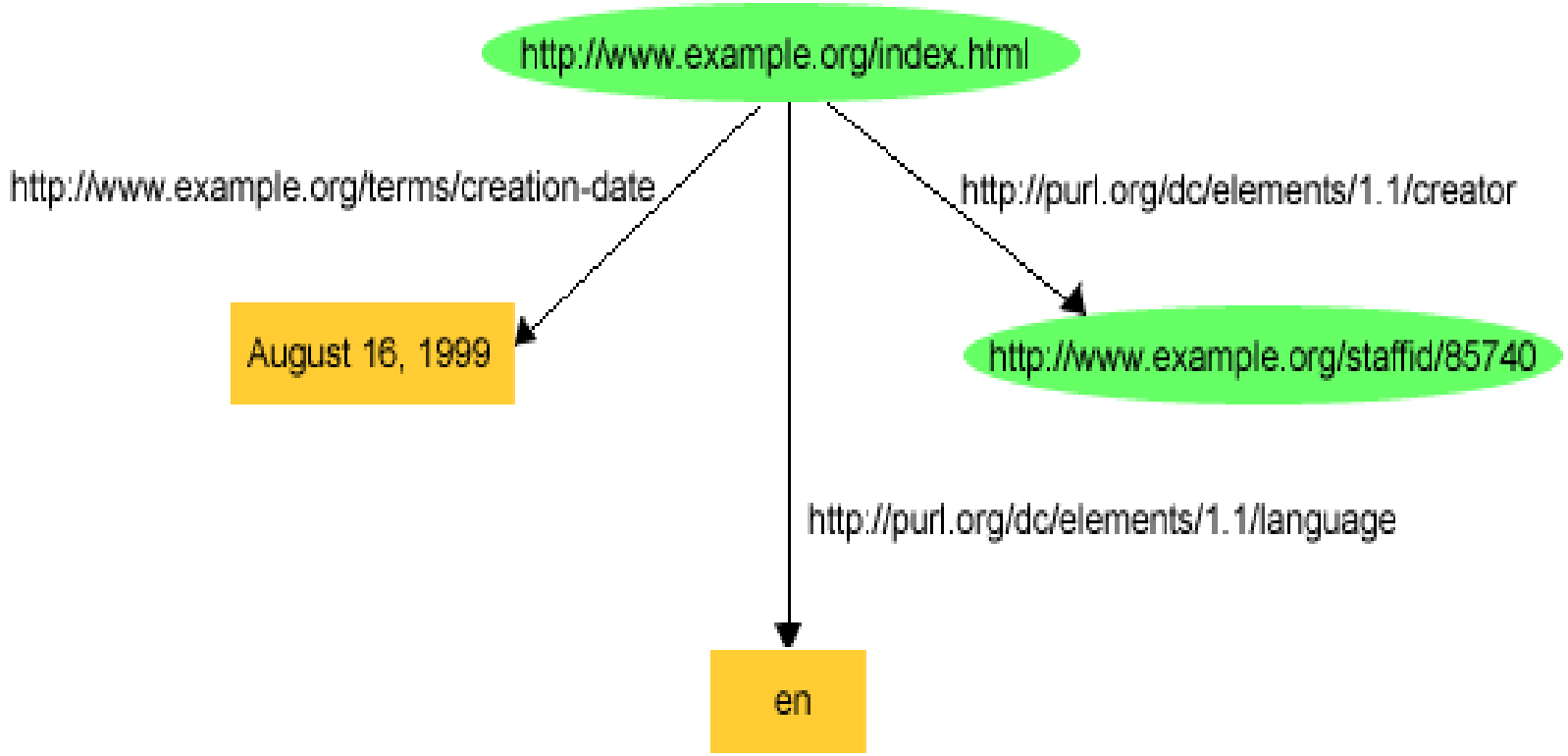
Additional RDF statements

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

- 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

RDF is represented as a graph



RDF statements as triples

Each triple corresponds to an arc in the graph

- <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" .

Abbreviation of URIs

- prefix rdf:, namespace URI:
  http://www.w3.org/1999/02/22-rdf-syntax-ns#
- prefix rdfs:, namespace URI:
  http://www.w3.org/2000/01/rdf-schema#
- prefix dc:, namespace URI:
  http://purl.org/dc/elements/1.1/
- prefix xsd:, namespace URI:
  http://www.w3.org/2001/XMLSchema#

- prefix ex:, namespace URI:
  http://www.example.org/ (or http://www.example.com/)
- prefix exterms:, namespace URI:
  http://www.example.org/terms/
- prefix exstaff:, namespace URI:
  http://www.example.org/staffid/

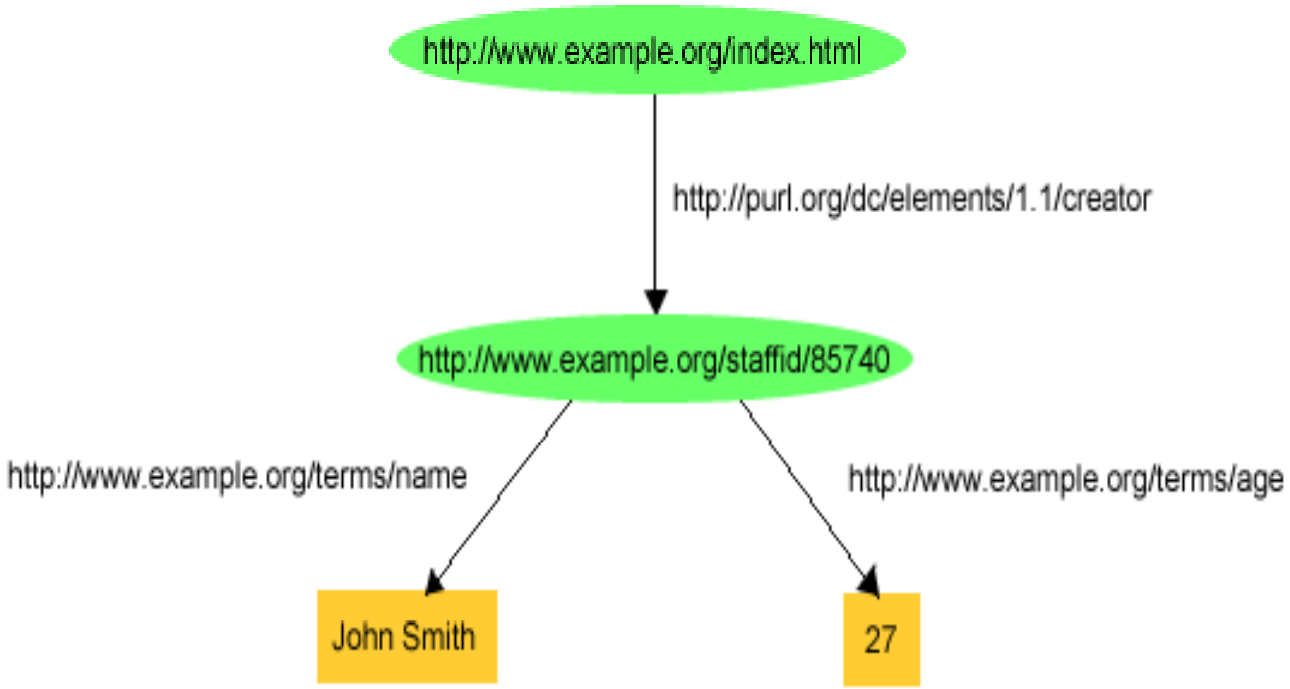
Abbreviated triples

- ex:index.html
  dc:creator
  exstaff:85740 .

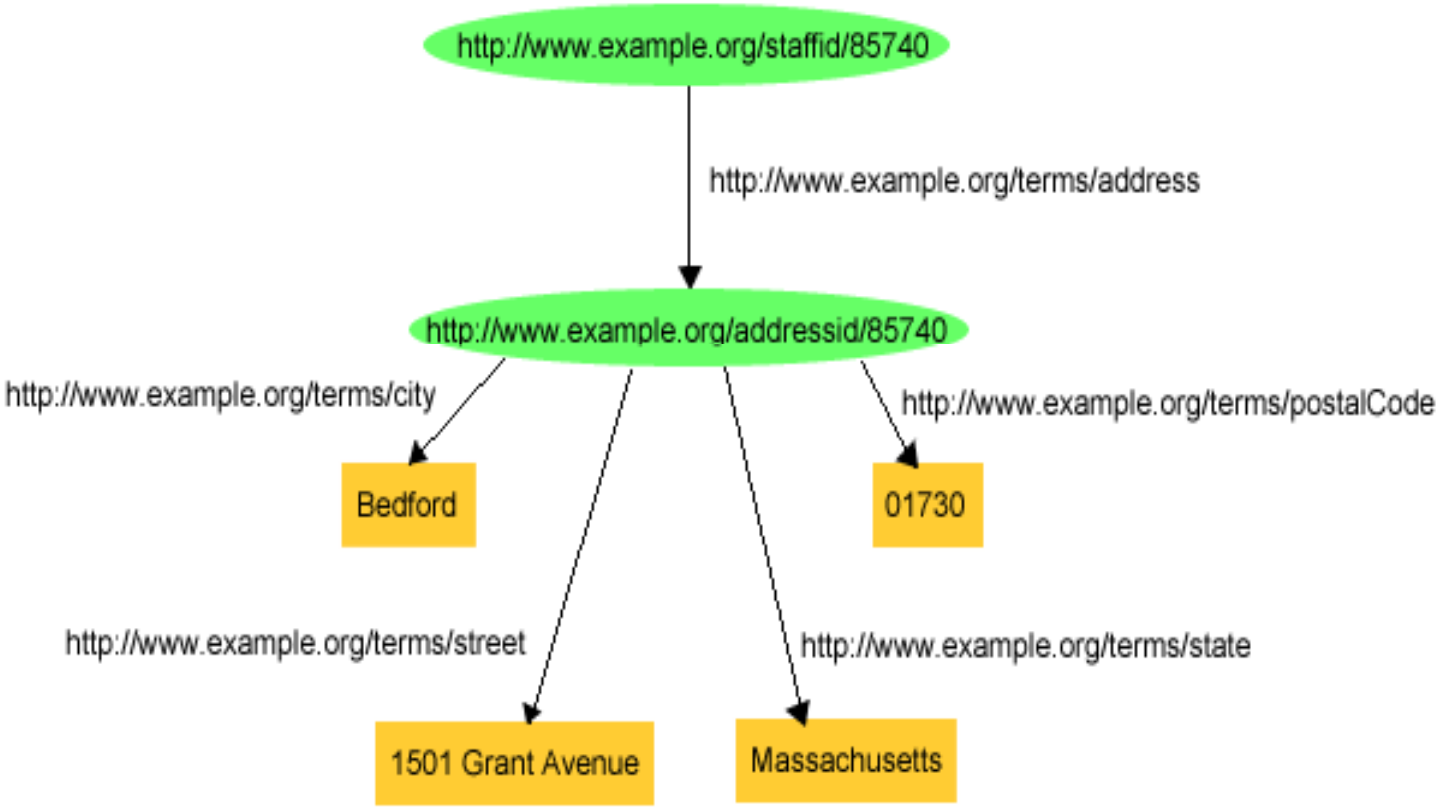
- ex:index.html
  exterms:creation-date
  "August 16, 1999" .

- ex:index.html
  dc:language
  "en" .

URI values can be subjects ...



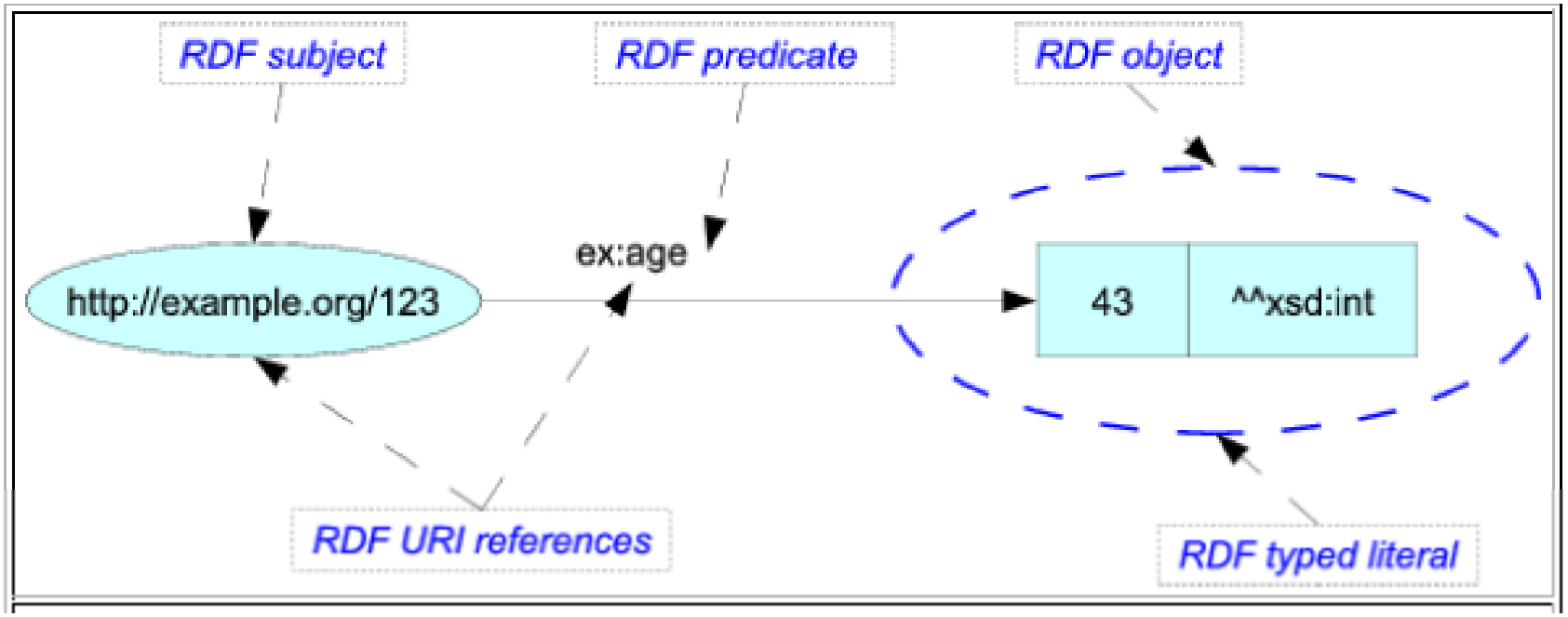
... to any depth ...


http://www.example.org/staffid/85740
http://www.example.org/terms/address
http://www.example.org/addressid/85740
http://www.example.org/terms/city
http://www.example.org/terms/postalCode
Bedford
01730
http://www.example.org/terms/street
http://www.example.org/terms/state
1501 Grant Avenue
Massachusetts

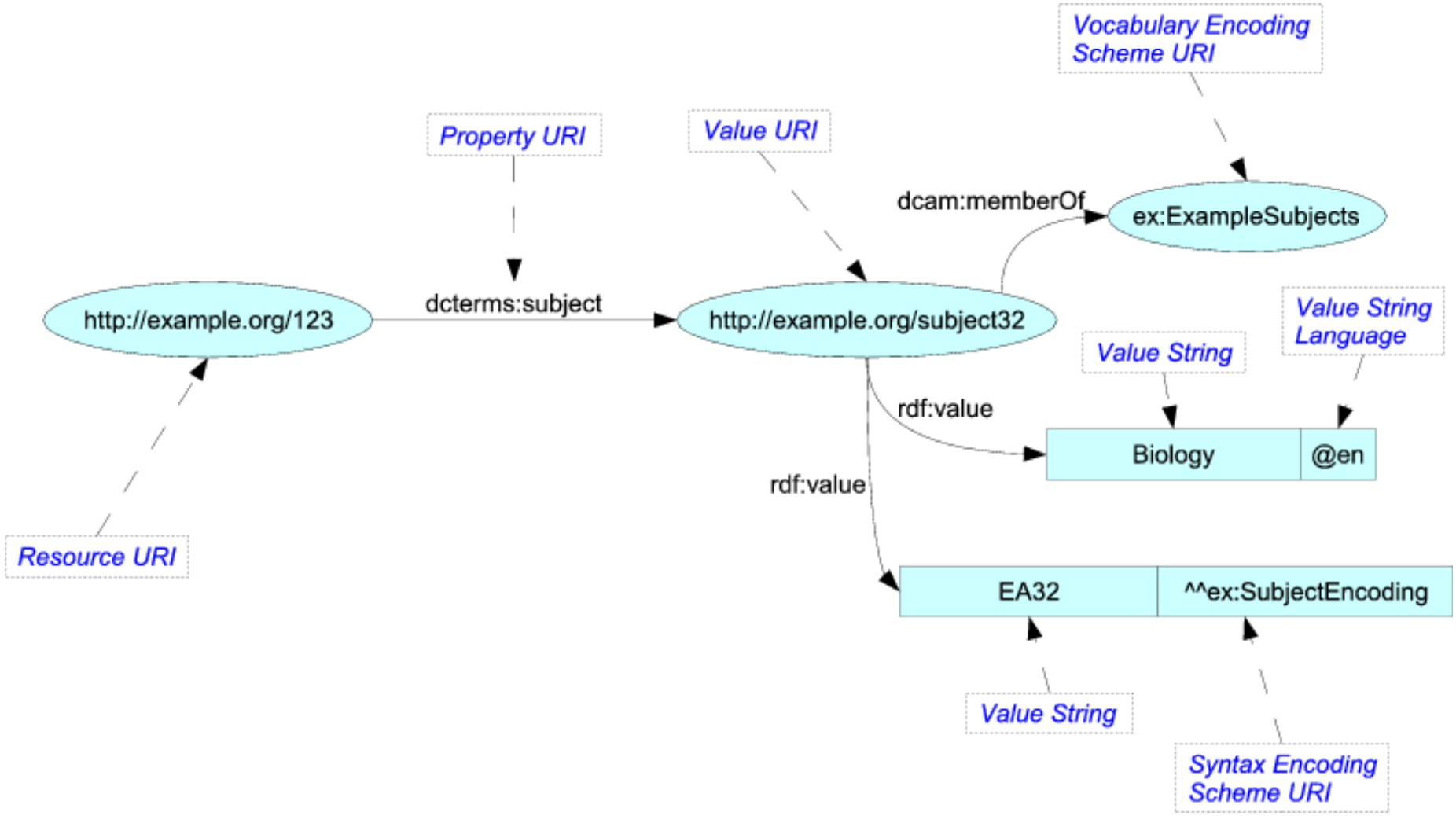
RDF summary

- An RDF graph is a set of RDF triples
- An RDF triple has three components
  - an RDF subject, which is an RDF URI reference or a blank RDF node
  - an RDF predicate, which is an RDF URI reference
  - an RDF object, which is an RDF URI reference, a blank RDF node or an RDF literal
- An RDF literal can be of two kinds
  - an RDF plain literal is a character string with an optional associated language tag describing the language of the character string
  - an RDF typed literal is a character string with an associated RDF datatype URI. An RDF datatype defines the syntax and semantics of a set of character strings that represent data such as booleans, integers, dates, etc.

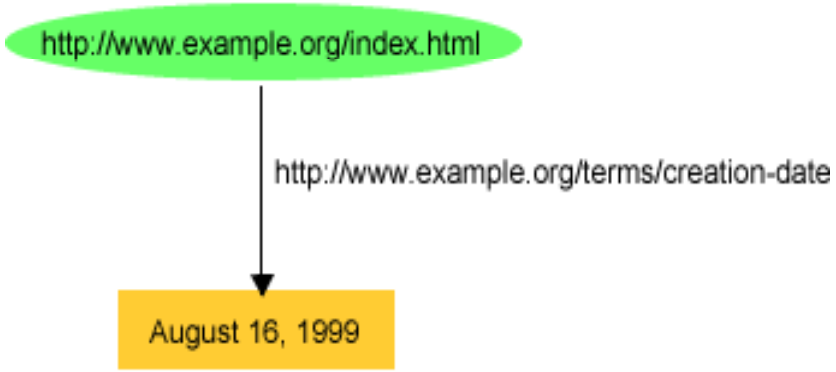
RDF example 1



RDF example 2

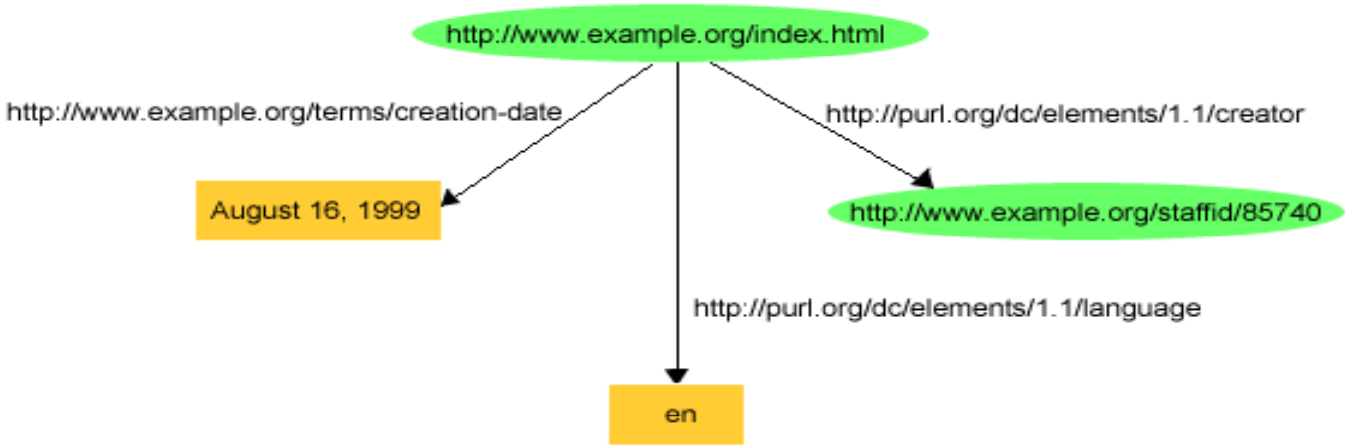


RDF graphs represented in XML




```
1. <?xml version="1.0"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.          xmlns:exterms="http://www.example.org/terms/">
4.   <rdf:Description rdf:about="http://www.example.org/index.html">
5.     <exterms:creation-date>August 16, 1999</exterms:creation-date>
6.   </rdf:Description>
7. </rdf:RDF>
```

RDF graph with abbreviated XML




```
1.  <?xml version="1.0"?>
2.  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.           xmlns:dc="http://purl.org/dc/elements/1.1/"
4.           xmlns:exterms="http://www.example.org/terms/">
5.    <rdf:Description rdf:about="http://www.example.org/index.html">
6.       <exterms:creation-date>August 16, 1999</exterms:creation-date>
7.       <dc:language>en</dc:language>
8.       <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
9.    </rdf:Description>
10. </rdf:RDF>
```

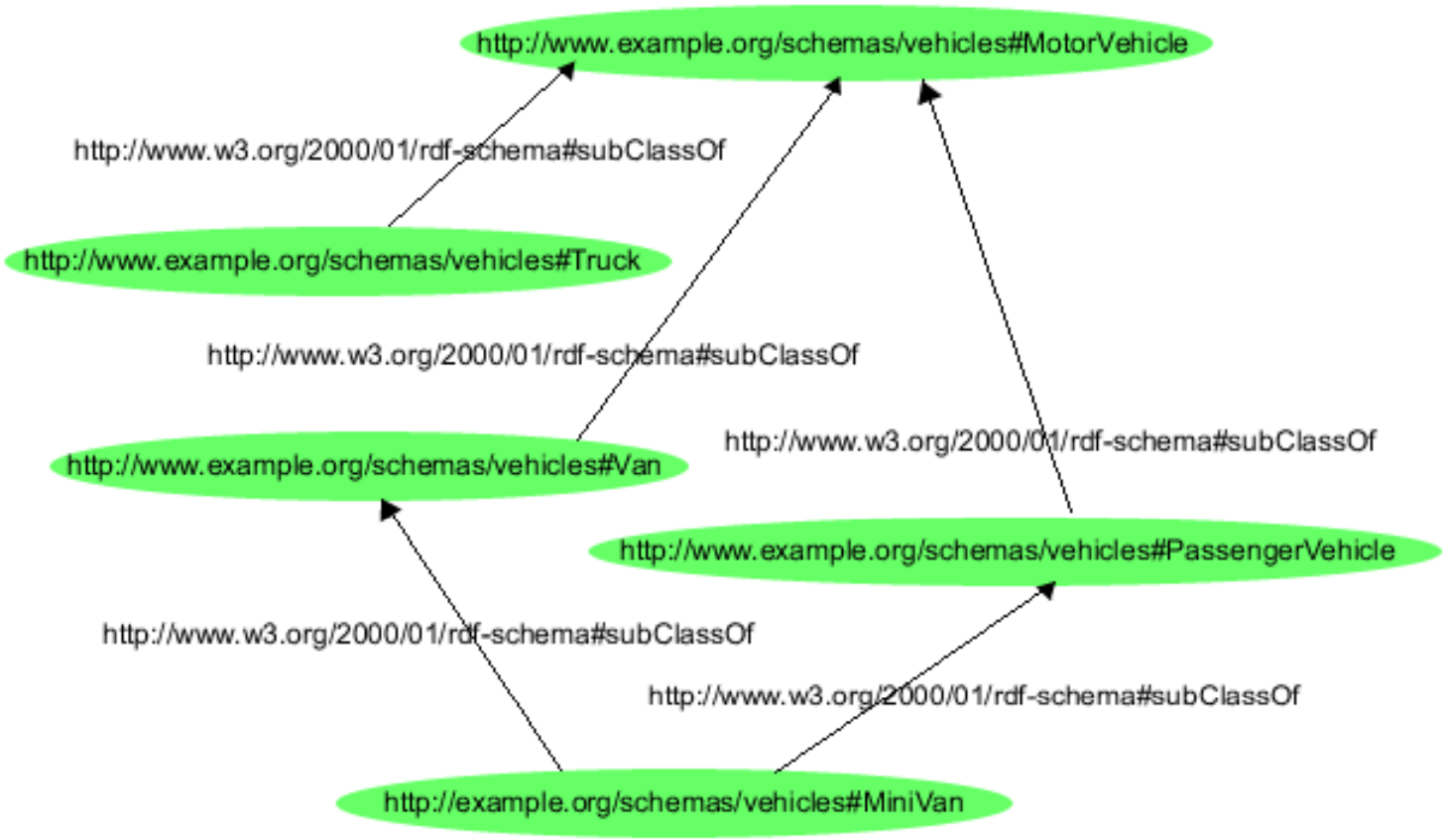
RDF Schema

- RDF provides a way to express simple statements about resources, using “named” properties and values
- It is convenient to define the *vocabularies* (terms) that are going to be used in those statements, to indicate that they are describing specific kinds or classes of resources, and will use specific properties in describing those resources
- For example, to describe bibliographic resources we could define classes such as “Book” or “Journal Article”, and use properties such as “author”, “title” and “subject” to describe them
- RDF Schema defines the terms used in RDF descriptions by providing a *type system* to be used in the RDF descriptions
- In other words, it provides a way to represent a “conceptual model” of a (small) part of the world, by defining its main “concepts” (classes), their properties and their relationships

Main notions of RDF Schema

- The main notions of the RDF schema are:
  - Classes, which can be organized in sub-classes, to any level (defining a taxonomy)
  - Properties, which also can be organized in sub-classes, to any level (defining another taxonomy)
- Vocabulary descriptions (schemas) written in the RDF Schema language are legal RDF graphs
- There is a close analogy with XML douments and XML schemas

Classes and subclasses



Representation of schema with triples

```
ex:MotorVehicle          rdf:type            rdfs:Class .
ex:PassengerVehicle      rdf:type            rdfs:Class .
ex:Van                   rdf:type            rdfs:Class .
ex:Truck                 rdf:type            rdfs:Class .
ex:MiniVan               rdf:type            rdfs:Class .

ex:PassengerVehicle      rdfs:subClassOf     ex:MotorVehicle .
ex:Van                   rdfs:subClassOf     ex:MotorVehicle .
ex:Truck                 rdfs:subClassOf     ex:MotorVehicle .

ex:MiniVan               rdfs:subClassOf     ex:Van .
ex:MiniVan               rdfs:subClassOf     ex:PassengerVehicle .
```

Properties, domains and ranges

- In the RDF schema properties are defined to better characterize classes
- Properties can be arranged in a taxonomy by defining subProperties
- For a property it is possible also to define its domain and its range
  - Domain is the class of entities that are the subject of an RDF statement where the property is used as the predicate
  - Range is the class of entities that are the object of an RDF statement where the property is used as the predicate

Examples of Properties

```
ex:Person      rdf:type      rdfs:Class
ex:Book        rdf:type      rdfs:Class
ex:hasauthor   rdf:type      rdfs:Property
ex:hasauthor   rdfs:domain   rdfs:Book
ex:hasauthor   rdfs:range    rdfs:Person

ex:hasMother   rdf:type      rdf:Property
ex:hasMother   rdfs:range    ex:Female .
ex:hasMother   rdfs:range    ex:Person .

exstaff:frank  ex:hasMother  exstaff:frances .
```

Frances (the mother of Frank) must be at the same time a person and a female

Summarizing the RDF Schema

- An RDF Schema is a simple “meta” vocabulary used to describe ontologies
  - Class, subClassOf, type
    - e.g., Person, Team
  - Property, subPropertyOf
    - e.g., playsFor
  - Domain (the **class for the subjects** of a particular *property*)
    - **Person** *playsFor* Team
  - Range (the **class for values** of a particular *property*)
    - Person *playsFor* **Team**