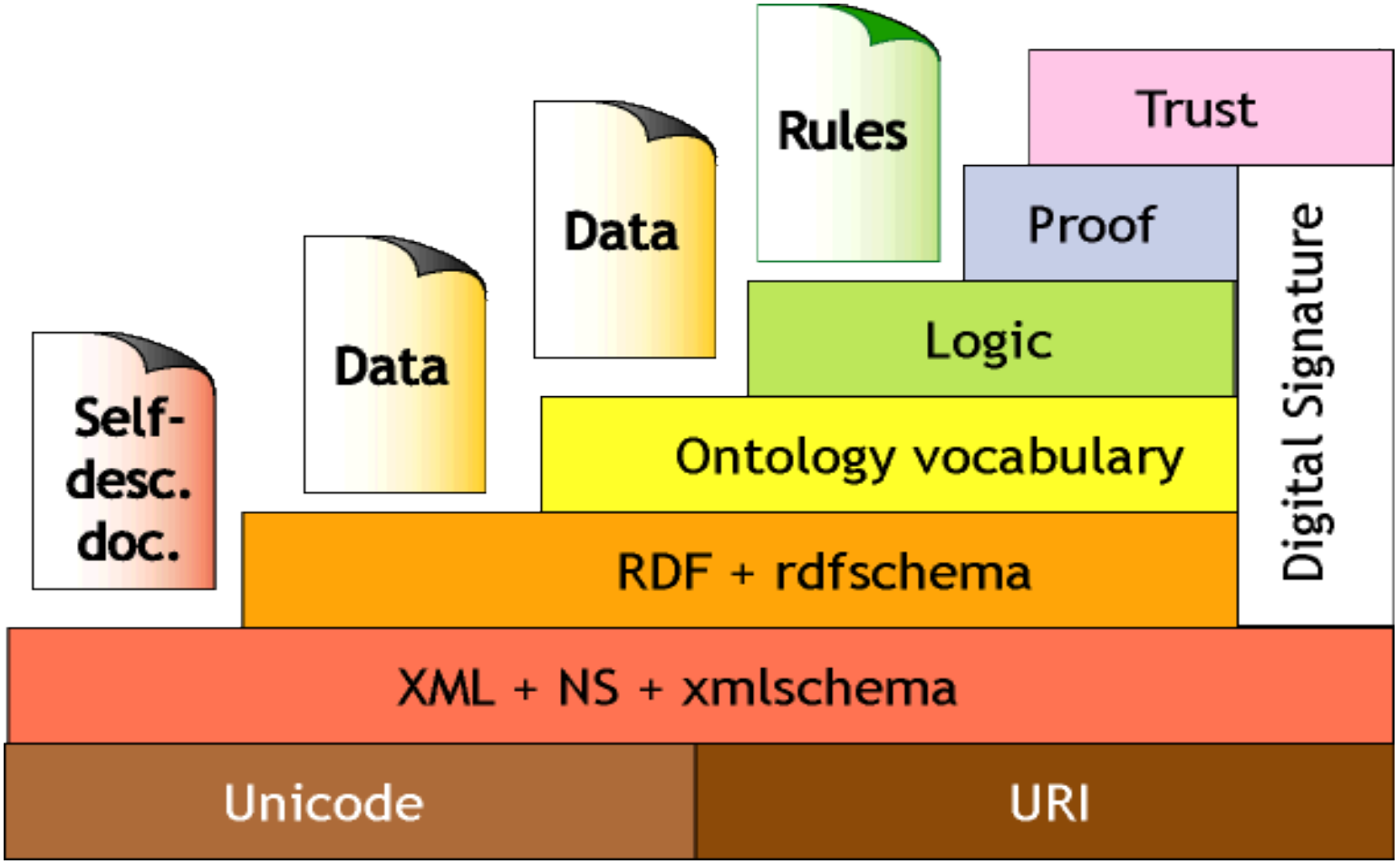
Grafovski podatkovni model RDF

Iztok Savnik

18/19

Stolp semantičnega spleta



XML

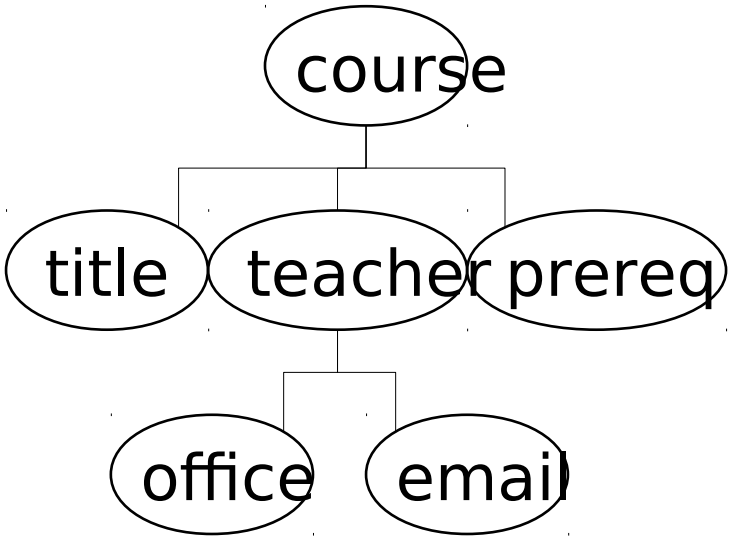
- XML: eXtensible Mark-up Language
- XML dokumenti so napisani z uporabniško definiranimi značkami
  - Oznake so uporabljene za izražanje “pomena” delov podatkov
- Pogosto uporabljene značke so definirane v slovarjih
  - Vsi poznajo pomen značk v slovarjih
  - Podatke zapisane v eni aplikaciji lahko uporablja druga aplikacija

XML

- XML: dokument = označeno drevo
- vozlišče = oznaka + atributi/vrednosti + vsebina

```
<course date="...">
  <title>...</title>
  <teacher>
    <office>...</office>
    <email>...</email>
  </teacher>
  <room>...</room>
  <prereq>...</prereq>
</course>
```

=



XML

- XML Schema = slovnica za opis dreves in podatkovnih tipov
  - Grafov ne moremo opisati
- XML na trivialen način ne moremo uporabiti za opis usmerjenih grafov
  - Z usmerjenimi grafi lahko predstavimo stavke jezika za predstavitev znanja (KR)
  - Lahko predstavimo tudi ontologije, objektno-usmerjen model, opisno logiko (predikatni račun), ...3
- XML je uporabljen kot eden od jezikov za predstavitev RDF

Graph data model

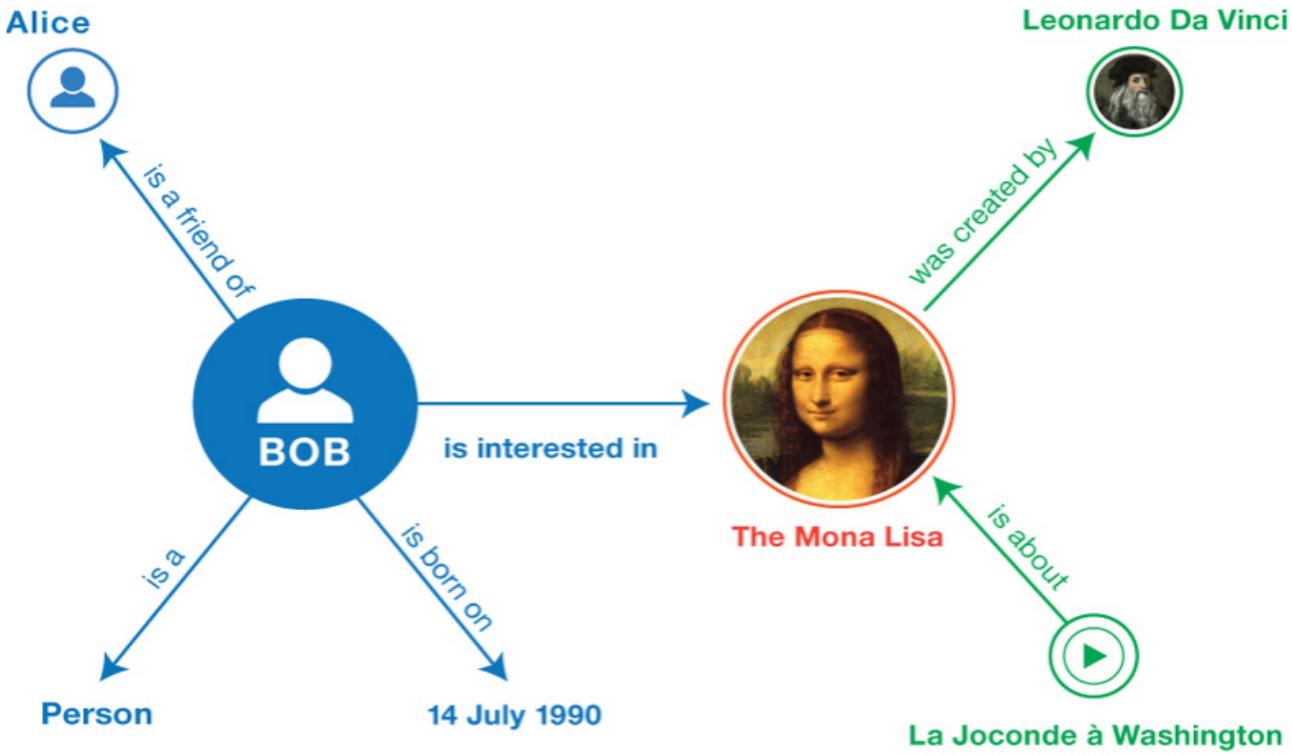
- Graph database
  - Database that uses graphs for the representation of data and queries
- Vertexes
  - Represent things, persons, concepts, classes, ...
- Arcs
  - Represent properties, relationships, associations, ...
  - Arcs have labels !

RDF

- Resource Description Framework
  - Tim Berners Lee, 1998-2009
  - This is movement !
- What is behind?
  - Graphs
  - Triples
  - Semantic networks, AI frames
  - Semantic data models
  - Human associative memory (psychology)

RDF

```
<Bob> <is a> <person>.
<Bob> <is a friend of> <Alice>.
<Bob> <is born on> <the 4th of July 1990>.
<Bob> <is interested in> <the Mona Lisa>.
<the Mona Lisa> <was created by> <Leonardo da Vinci>.
<the video 'La Joconde à Washington'> <is about> <the Mona Lisa>
```



RDF syntax

- N3, TVS
- Turtle
- TriG
- N-Triples
- RDF/XML
- RDF/JSON

Name spaces

- Using short names for URL-s
  - Long names are tedious
- Simple but strong concept
- Defining name space:

prefix rdf:, namespace URI: http://www.w3.org/1999/02/22-rdf-syntax-ns#

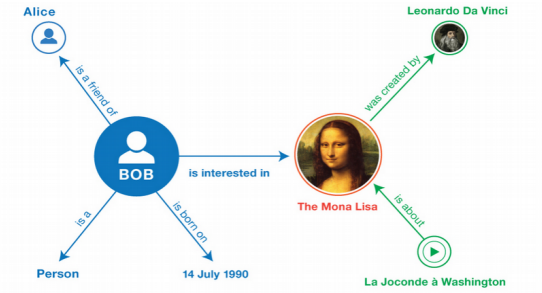
prefix rdfs:, namespace URI: http://www.w3.org/2000/01/rdf-schema#

prefix dc:, namespace URI: http://purl.org/dc/elements/1.1/

prefix owl:, namespace URI: http://www.w3.org/2002/07/owl#

prefix ex:, namespace URI: http://www.example.org/ (or http://www.example.com/)

prefix xsd:, namespace URI: http://www.w3.org/2001/XMLSchema#



N-Triples

```
<http://example.org/bob#me> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
<http://example.org/bob#me> <http://xmlns.com/foaf/0.1/knows> <http://example.org/alice#me> .
<http://example.org/bob#me> <http://schema.org/birthDate> "1990-07-04"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://example.org/bob#me> <http://xmlns.com/foaf/0.1/topic_interest> <http://www.wikidata.org/entity/Q12418> .
<http://www.wikidata.org/entity/Q12418> <http://purl.org/dc/terms/title> "Mona Lisa" .
<http://www.wikidata.org/entity/Q12418> <http://purl.org/dc/terms/creator> <http://dbpedia.org/resource/Leonardo_da_Vinci>
<http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619> <http://purl.org/dc/terms/subject
```

Turtle

```
01  BASE   <http://example.org/>
02  PREFIX foaf: <http://xmlns.com/foaf/0.1/>
03  PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
04  PREFIX schema: <http://schema.org/>
05  PREFIX dcterms: <http://purl.org/dc/terms/>
06  PREFIX wd: <http://www.wikidata.org/entity/>
07
08  <bob#me>
09      a foaf:Person ;
10      foaf:knows <alice#me> ;
11      schema:birthDate "1990-07-04"^^xsd:date ;
12      foaf:topic_interest wd:Q12418 .
13
14  wd:Q12418
15      dcterms:title "Mona Lisa" ;
16      dcterms:creator <http://dbpedia.org/resource/Leonardo_da_Vinci> .
17
```

Additional RDF Constructs

- Complex values
  - Bags, lists, trees, graphs
- Blank nodes
- Types of atomic values
- Types of nodes
- Reification

RDF Schema

- RDFS
- Knowledge representation language
  - Not just graph any more !
  - AI Frames, Object Model
- Small dictionary for RDFS
  - rdfs:class, rdfs:subClassOf, rdfs:type
  - rdfs:property, rdfs:subPropertyOf
  - rdfs:domain, rdfs:range

RDF data model

- RDF is a data model
  - Independent of the domain and application
  - Data model is based on the directed labelled graph
- RDF datamodel is abstract and it does not depend or XML or some other data representation language
  - XML is one of the possible syntaxes for RDF

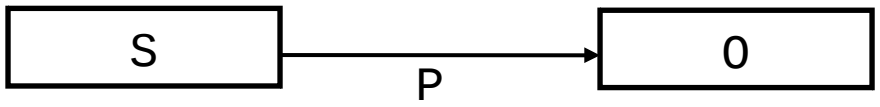
RDF data model

RDF graph = a set of RDF triples

triple = statement

```
(subject, predicate, object)
```

- Subject = resource
- Predicate = property
- Object = value



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

RDF triple

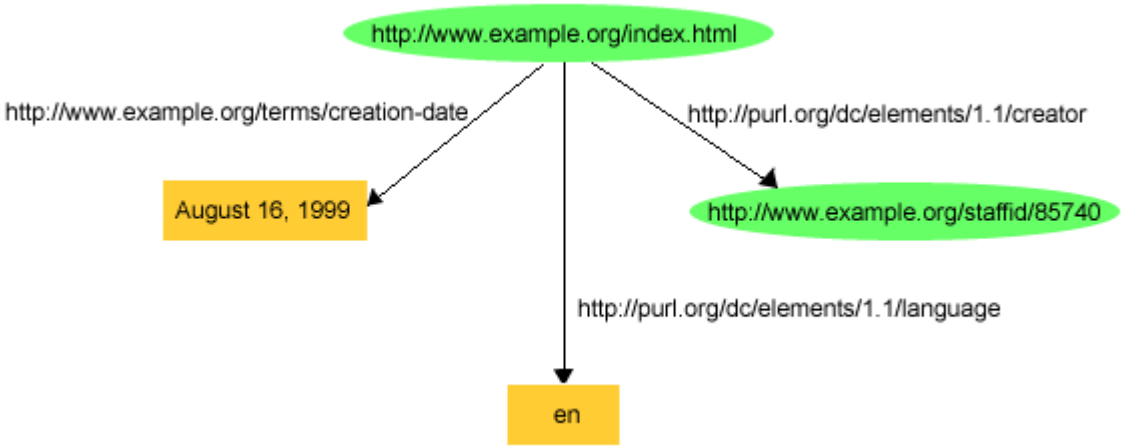
- Subject: http://www.example.org/index.html
- Predicate: http://purl.org/dc/elements/1.1/creator
- Object: http://www.example.org/staffid/85740

Every triple represents one arc in the graph.

RDF description

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

<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" .



- Full URIs are used
- Concepts are ovals
- Literals are rectangles

Uniform Resource Identifiers

- Web provides a general form of identifier, called the Uniform Resource Identifier (URI)
  - Identifying (naming) resources on the Web
  - Uniform Resource Locator (URL) is a special form of URI
  - Different persons or organizations can independently create URIs
  - Not limited to identifying things that have network locations, or use other computer access mechanisms

Uniform Resource Identifiers

- To be precise, RDF uses URI references [URIS]
  - URI reference (or URIref) is a URI, together with an optional fragment identifier at the end
  - For example, the URI reference

    http://www.example.org/index.html#section2

    consists of the URI http://www.example.org/index.html and (separated by the "#" character) the fragment identifier Section2

URI-based Vocabulary and Node Identification

- A RDF node may be
  - URI with optional fragment identifier (URIref),
  - a literal, or
  - a blank node
- Properties are URI references
- Literals are strings
  - converted into typed values
- Blank node
  - unique node that can be used in one or more RDF statements

Datataypes

- Datatypes are used by RDF in the representation of values such as integers, floating point numbers and dates
  - predefined XML Schema datatypes are expected to be widely used for this purpose.
  - datatype consists of a lexical space, a value space and a lexical-to-value mapping (details later)
  - RDF predefines just one datatype rdf:XMLLiteral

Literals

- Literals are used to identify values such as numbers and dates by means of a lexical representation
  - literal may be the object of an RDF statement, but not the subject or the predicate
- Literals may be plain or typed :
  - plain literal is a string combined (optional language tag)
  - typed literal is a string combined with a datatype URI
- Example: xsd:boolean

Typed Literal	Lexical-to-Value Mapping	Value
<xsd:boolean, "true">	<"true", T>	T
<xsd:boolean, "1">	<"1", T>	T
<xsd:boolean, "false">	<"false", F>	F
<xsd:boolean, "0">	<"0", F>	F

Dictionaries

- Well-defined and documented dictionaries of predicates are very important
  - This is the vocabulary of the language

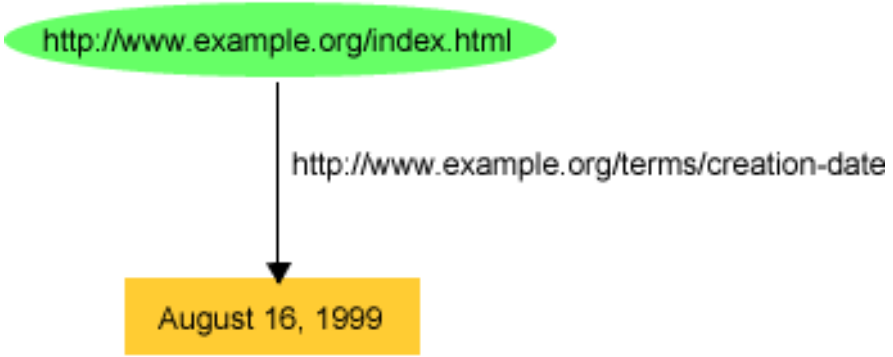
```
prefix rdf:, namespace URI: http://www.w3.org/1999/02/22-rdf-syntax-ns#
prefix rdfs:, namespace URI: http://www.w3.org/2000/01/rdf-schema#
prefix dc:, namespace URI: http://purl.org/dc/elements/1.1/
prefix owl:, namespace URI: http://www.w3.org/2002/07/owl#
prefix ex:, namespace URI: http://www.example.org/ (or http://www.example.com/)
prefix xsd:, namespace URI: http://www.w3.org/2001/XMLSchema#
```

- User-defined predicates may be used but no other application would understand that

One sentence

http://www.example.org/index.html has a creation-date whose value is August 16, 1999

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




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

<rdf:Description rdf:about="http://www.example.org/index.html">
     <exterms:creation-date>August 16, 1999</exterms:creation-date>
</rdf:Description>

</rdf:RDF>
```

Two sentences

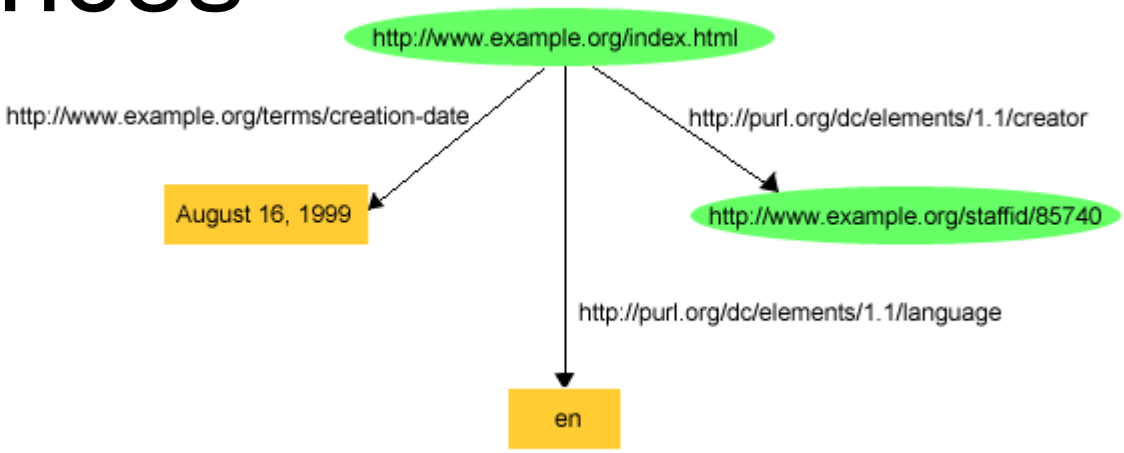
http://www.example.org/index.html has a creation-date whose value is August 16, 1999
http://www.example.org/index.html has a language whose value is English

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

```
<?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/1.1/"
         xmlns:exterms="http://www.example.org/terms/">

<rdf:Description rdf:about="http://www.example.org/index.html">
<exterms:creation-date>August 16, 1999</exterms:creation-date>
</rdf:Description>

<rdf:Description rdf:about="http://www.example.org/index.html">
<dc:language>en</dc:language>
</rdf:Description>

</rdf:RDF>
```

ex:index.html dc:creator exstaff:85740 .
ex:index.html exterms:creation-date "August 16, 1999" .
ex:index.html dc:language "en" .

Three sentences

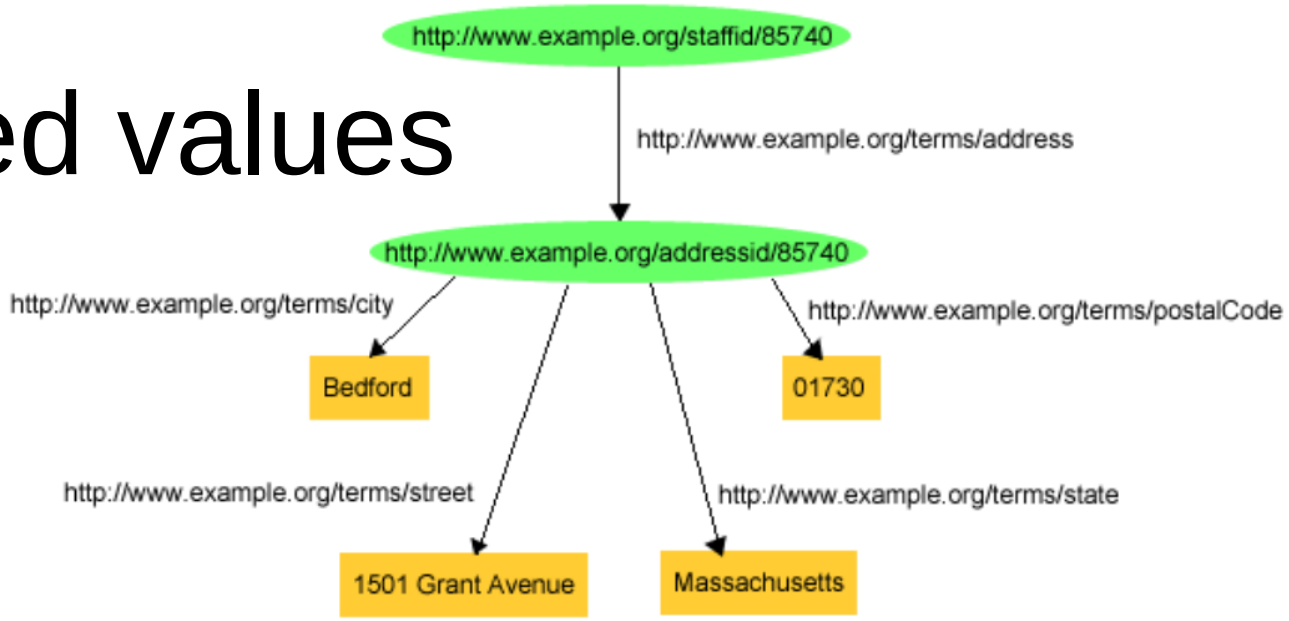



```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
                 xmlns:dc="http://purl.org/dc/elements/1.1/"
                 xmlns:exterms="http://www.example.org/terms/">

<rdf:Description rdf:about="http://www.example.org/index.html">
   <exterms:creation-date>August 16, 1999</exterms:creation-date>
   <dc:language>en</dc:language>
   <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
</rdf:Description>

</rdf:RDF>
```

Complex values

- Values of properties are not just simple literals
- … they can be the nodes of a graph
  - Arbitrarily complex trees and graphs can be constructed
  - Values can be nested sintacticaly or referenced

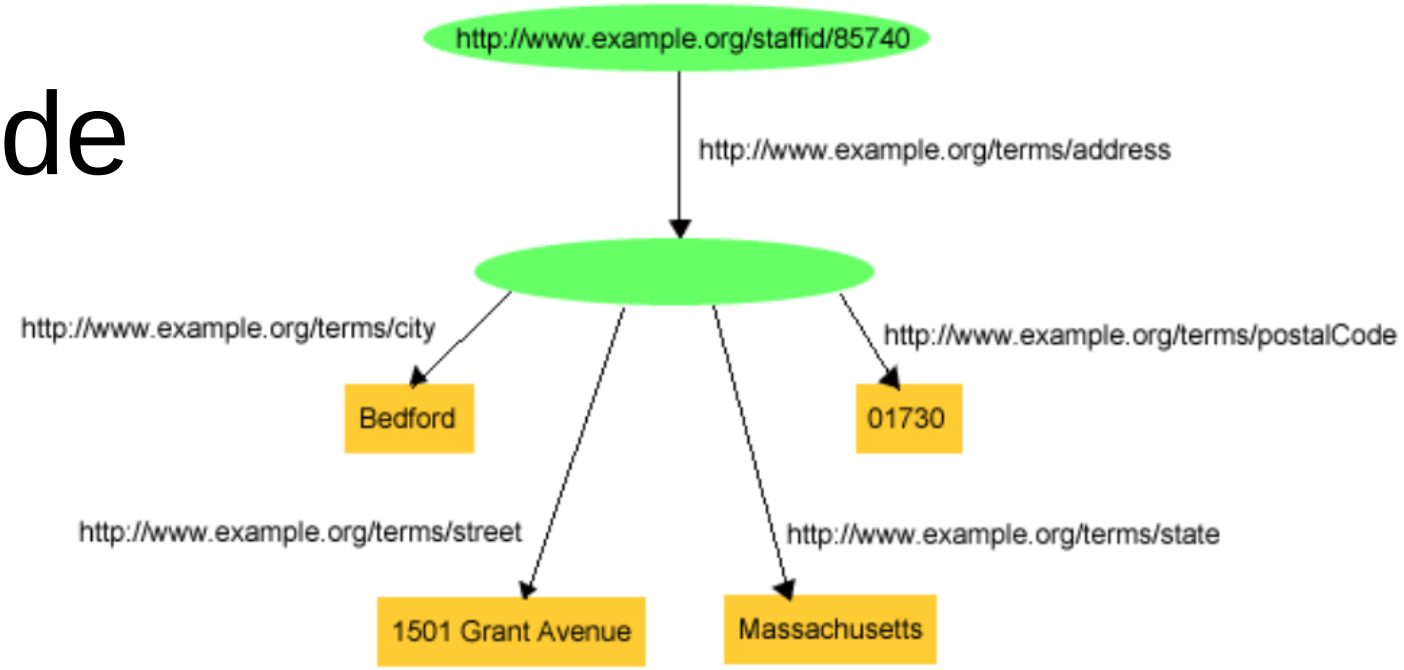
Structured values




```
exstaff:85740       exterms:address        exaddressid:85740 .
exaddressid:85740   exterms:street         "1501 Grant Avenue" .
exaddressid:85740   exterms:city           "Bedford" .
exaddressid:85740   exterms:state          "Massachusetts" .
exaddressid:85740   exterms:postalCode     "01730" .
```

- URI exaddressid:85740 may never be accessed directly but only through the person exstaff:85740
  - Therefore, there is no need for universal identifier
  - But, a graph may contain more than one blank node

Blank node




```
exstaff:85740   exterms:address          ??? .
???             exterms:street           "1501 Grant Avenue" .
???             exterms:city             "Bedford" .
???             exterms:state            "Massachusetts" .
???             exterms:postalCode       "01730" .
```

```
exstaff:85740   exterms:address          _:johnaddress .
_:johnaddress   exterms:street           "1501 Grant Avenue" .
_:johnaddress   exterms:city             "Bedford" .
_:johnaddress   exterms:state            "Massachusetts" .
_:johnaddress   exterms:postalCode       "01730" .
```

- Blank node does not need URI
- It has local ID
- Explicit identifier is needed in order to represent this graph as triples

Blank node

- Have meaning only in the single graph
- Blank node identifiers
  - just a way of representing the blank nodes in a graph when the graph is written in triple form
  - If it is expected that a node in a graph will need to be referenced from outside the graph, a URIref should be assigned

Blank node

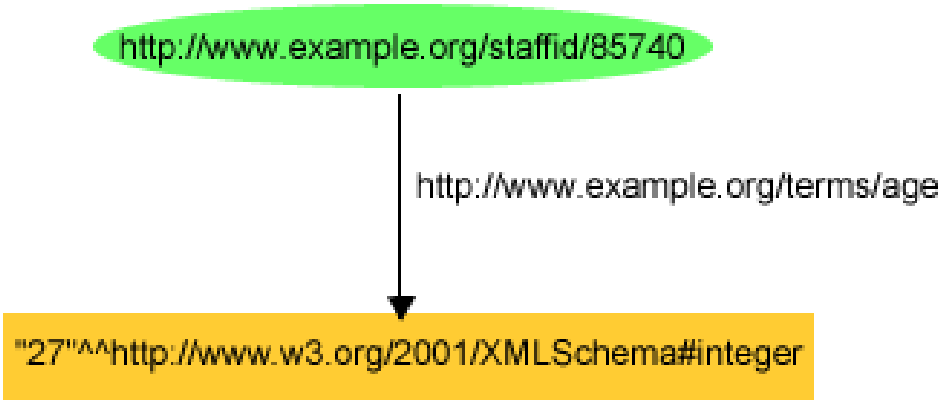
- Important aspect of RDF:
  - RDF directly represents only binary relationships
- N-ary (n-way) relationship
  - Blank node can represent relationship
  - n-way relationship must be broken up into a group of separate binary relationships

Literal type

- Literal has type
  - 27 means integer number and not characters “2” and “7”
  - Decimal, hexadecimal, string ?
  - Lexsical space & value space
  - RDF does not have internal types
- RDF datatype
  - Types are defied outside RDF
  - XML Schema simple types
    - `xsd:integer, xsd:float, xsd:double, xsd:boolean, xsd:boolean, xsd: date, …`

Typed literals

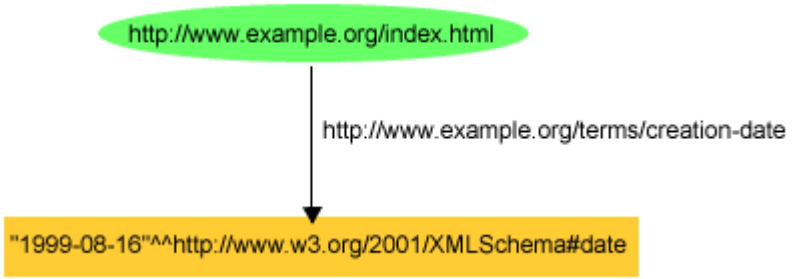
- Datatype URIs
  - XML Schema
  - Mapping between the lexical and value spaces
  - QName simplification for writing long URIs




```
<http://www.example.org/staffid/85740>
<http://www.example.org/terms/age>
"27"^^<http://www.w3.org/2001/XMLSchema#integer> .
```

```
exstaff:85740 exterms:age "27"^^xsd:integer .
```

Typed literals




```
ex:index.html exterms:creation-date "1999-08-16"^^xsd:date .
```

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

<rdf:Description rdf:about="http://www.example.org/index.html">
   <exterms:creation-date rdf:datatype=
   "http://www.w3.org/2001/XMLSchema#date">1999-08-16 </exterms:creation-date>
</rdf:Description>

</rdf:RDF>
```

Typed literals and entity

```
<?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:exterms="http://www.example.org/terms/">

<rdf:Description rdf:about="http://www.example.org/index.html">
    <exterms:creation-date rdf:datatype="&xsd;date">1999-08-16
    </exterms:creation-date>
</rdf:Description>

</rdf:RDF
```

Rdf:ID

- Up to here we used `rdf:about` for the definition of the described object
  - Describing resources
- Explicit identifier of an object
  - rdf:ID attribute is somewhat similar to the ID attribute in XML and HTML
  - name which must be unique relative to the current base URI
  - Similar to XML ID: unique inside the base URI

Example:Tents

Collection of objects:
http://www.example.com/2007/04/products

```
1.   <?xml version="1.0"?>
2.   <!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
3.   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.                xmlns:exterms="http://www.example.com/terms/">
5.     <rdf:Description rdf:ID="item10245">
6.          <exterms:model rdf:datatype="&xsd;string">Overnighter</exterms:model>
7.          <exterms:sleeps rdf:datatype="&xsd;integer">2</exterms:sleeps>
8.          <exterms:weight rdf:datatype="&xsd;decimal">2.4</exterms:weight>
9.          <exterms:packedSize rdf:datatype="&xsd;integer">784</exterms:packedSize>
10.    </rdf:Description>
  ...other product descriptions...
11.  </rdf:RDF>
```

- rdf:ID specifies a fragment identifier
- fragment identifier item10245 will be interpreted relative to a base URI
- full URIref for the tent = BaseURI + # + ID

`http://www.example.com/2002/04/products#item10245`

Identification of RDF object

- Every product has its own ID

```
http://www.example.com/2007/04/products#item10245
```

- Instead of rdf:ID="item10245"
  we can use rdf:about="#item10245"
- To describe an object outside the base URI we use

```
<rdf:Description rdf:about="http://www.example.com/2002/04/products#item10245">
```

- Basic principle of WEB
  - Data about an object can be added anywhere !
- XML Base:
  - Basic resource used

Example: XML Base

```
1.   <?xml version="1.0"?>
2.   <!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
3.   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.               xmlns:exterms="http://www.example.com/terms/"
5.               xml:base="http://www.example.com/2002/04/products">

6.     <rdf:Description rdf:ID="item10245">
7.          <exterms:model rdf:datatype="&xsd;string">Overnighter</exterms:model>
8.          <exterms:sleeps rdf:datatype="&xsd;integer">2</exterms:sleeps>
9.          <exterms:weight rdf:datatype="&xsd;decimal">2.4</exterms:weight>
10.         <exterms:packedSize rdf:datatype="&xsd;integer">784</exterms:packedSize>
11.    </rdf:Description>

  ...other product descriptions...

12.  </rdf:RDF>
```

#item10245 will be interpreted relative to a base URI

Example: Rating of the Tent

```
1.  <?xml version="1.0"?>
2.  <!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
3.  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.              xmlns:sportex="http://www.exampleRatings.com/terms/">

5.    <rdf:Description rdf:about="http://www.example.com/2002/04/products#item10245">
6.         <sportex:ratingBy rdf:datatype="&xsd;string">Richard Roe</sportex:ratingBy>
7.         <sportex:numberStars rdf:datatype="&xsd;integer">5</sportex:numberStars>
8.    </rdf:Description>
9.  </rdf:RDF>
```

- RDF located outside the catalog
  - refers to this tent by using the full URIref
- rdf:Description element with an rdf:about attribute
  - full URIref of the tent precisely identifies the tent description
  - catalog is the source for descriptions of those resources
  - anyone should be able to freely add information about an existing resource, using any vocabulary they please

Resource type

- Resources are classified into classes
  - Similar to prog.languages
- The predicate `rdf:type`
  - The value of this predicate is the category or the class of the thing
  - Concept defined with a type is the instance of a class

Example: use rdf:type

```
<?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:exterms="http://www.example.com/terms/"
     xml:base="http://www.example.com/2002/04/products">

<rdf:Description rdf:ID="item10245">
   <rdf:type rdf:resource="http://www.example.com/terms/Tent"/>
   <exterms:model rdf:datatype="&xsd;string">Overnighter</exterms:model>
   <exterms:sleeps rdf:datatype="&xsd;integer">2</exterms:sleeps>
   <exterms:weight rdf:datatype="&xsd;decimal">2.4</exterms:weight>
   <exterms:packedSize rdf:datatype="&xsd;integer">784</exterms:packedSize>
</rdf:Description>

...other product descriptions...

</rdf:RDF>
```

Definicija tipov

- Sam RDF nima gradnikov za definicijo novih tipov
- Definicijo tipov omogoča RDF Schema
  - razširitev slovarja za delo s tipi
- Instance razreda lahko definiramo na enostavnejši način
  - `rdf:type` lastnost je odstranjena
  - Uporabimo QName ime za definicijo značk

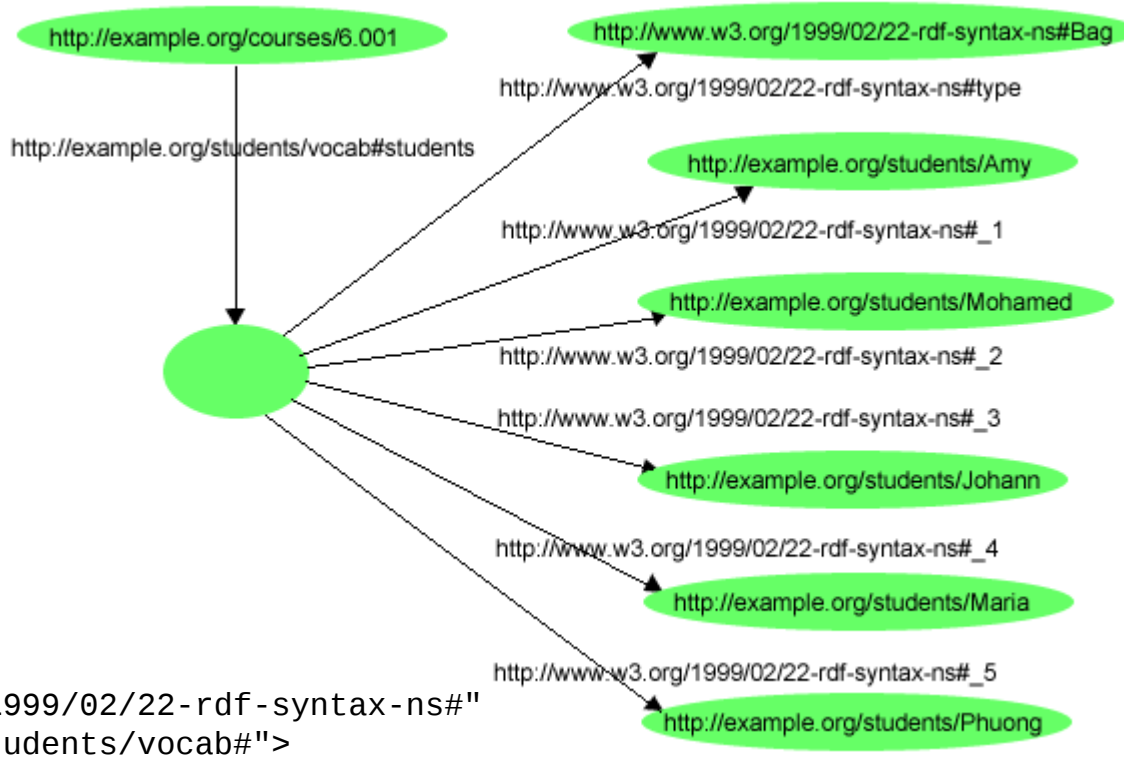
Krajša predstavitev primerkov

```
<?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:exterms="http://www.example.com/terms/"
xml:base="http://www.example.com/2002/04/products">

<exterms:Tent rdf:ID="item10245">
<exterms:model rdf:datatype="&xsd;string">Overnighter</exterms:model>
<exterms:sleeps rdf:datatype="&xsd;integer">2</exterms:sleeps>
<exterms:weight rdf:datatype="&xsd;decimal">2.4</exterms:weight>
<exterms:packedSize rdf:datatype="&xsd;integer">784</exterms:packedSize>
</exterms:Tent>

...other product descriptions...

</rdf:RDF>
```

Kontejnerji

- Kontejnerji omogočajo grupiranje virov (ali besed)
- Napišemo lahko izjave o kontejnerju (kot celota) ali individualno o njegovih članih
- Poznamo različne tipe kontejnerjev
  - Vreča (rdf:Bag) – neurejena kolekcija
  - Zaporedje (rdf:Seq) – urejena kolekcija (= “sekvenca”)
  - Alternative (rdf:Alt) – predstavlja alternative
- Možno je tudi kreirati kolekcije na osnovi vzorcev URI
- Dvojniki so dovoljeni (ni mehanizma za zagotavljanje unikatnosti vrednosti)

Vreča




```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:s="http://example.org/students/vocab#">

<rdf:Description rdf:about="http://example.org/courses/6.001">
    <s:students>
     <rdf:Bag>
            <rdf:li rdf:resource="http://example.org/students/Amy"/>
            <rdf:li rdf:resource="http://example.org/students/Mohamed"/>
            <rdf:li rdf:resource="http://example.org/students/Johann"/>
            <rdf:li rdf:resource="http://example.org/students/Maria"/>
            <rdf:li rdf:resource="http://example.org/students/Phuong"/>
     </rdf:Bag>
    </s:students>
</rdf:Description>

</rdf:RDF>
```

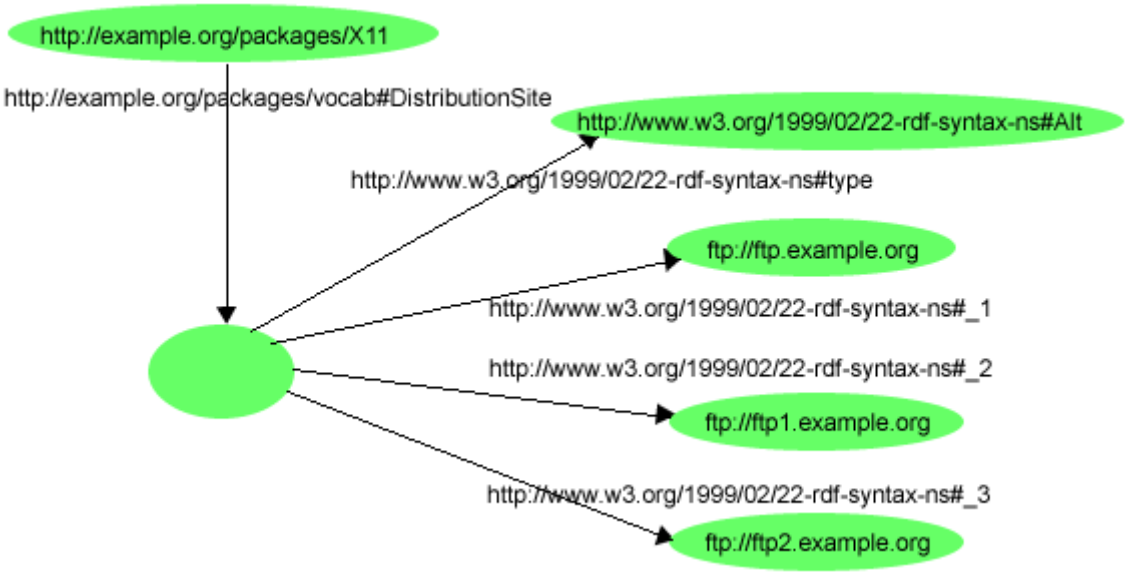
Dostop do elementov

- Lastnosti, ki omogočajo dostop do članov kontejnerja
  - `rdf:_1, rdf:_2, …rdf:_n`
  - Člane definiramo z `rdf:li`
- Lahko definiramo tudi dodatne lastnosti kontejnerjev

Alternative

- Predstavitev možnosti, ki so na razpolago
- Kontejner mora imeti vsaj en element
- Ni možnosti za definicijo privzete alternative
  - Lahko definiramo sami
  - Pomen je na strani aplikacije!
- Uporabniki lahko definirajo kontejnerje na svoj način
  - RDF kontejnerji samo nudijo skupno definicijo!

Alternative

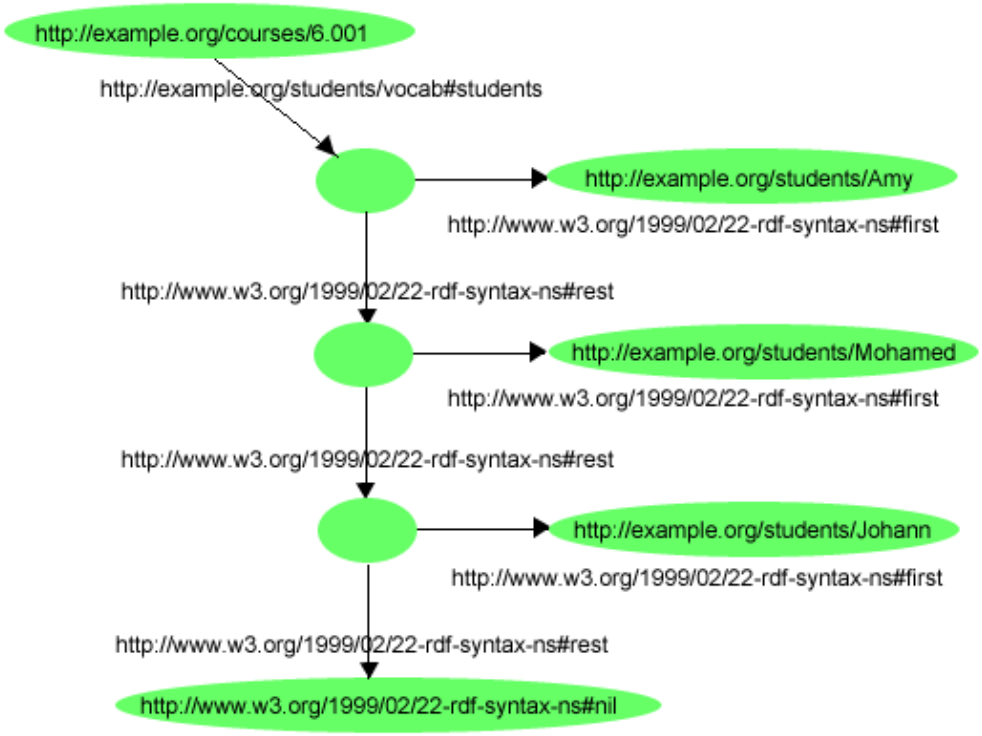



```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:s="http://example.org/packages/vocab#">

<rdf:Description rdf:about="http://example.org/packages/X11">
   <s:DistributionSite>
    <rdf:Alt>
        <rdf:li rdf:resource="ftp://ftp.example.org"/>
        <rdf:li rdf:resource="ftp://ftp1.example.org"/>
        <rdf:li rdf:resource="ftp://ftp2.example.org"/>
    </rdf:Alt>
   </s:DistributionSite>
</rdf:Description>

</rdf:RDF>
```

Kolekcije

- Kontejnerje ne moremo “zapreti”
  - Ne moremo reči: “to so vsi člani kontejnerja”?
  - Kontejner pravi: “nekateri identificirani viri so člani”
  - Ni mogoče povedati, da je nekje še dodaten graf, ki opisuje še druge člane
- Skupina objektov predstavljena z RDF seznamom
- Predefiniran tip **`rdf:List`**
  - **`rdf:first, rdf:rest`**
- Posebna notacija za opis kolekcij **`rdf:parseType="Collection"`**
  - Seznam članov
  - Vsebina elementa naj se obravnava na poseben način !

Kolekcija




```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:s="http://example.org/students/vocab#">

<rdf:Description rdf:about="http://example.org/courses/6.001">
   <s:students rdf:parseType="Collection">
    <rdf:Description rdf:about="http://example.org/students/Amy"/>
    <rdf:Description rdf:about="http://example.org/students/Mohamed"/>
    <rdf:Description rdf:about="http://example.org/students/Johann"/>
   </s:students>
</rdf:Description>

</rdf:RDF>
```

Daljša verzija

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:s="http://example.org/students/vocab#">

<rdf:Description rdf:about="http://example.org/courses/6.001">
   <s:students rdf:nodeID="sch1"/>
</rdf:Description>

<rdf:Description rdf:nodeID="sch1">
   <rdf:first rdf:resource="http://example.org/students/Amy"/>
   <rdf:rest rdf:nodeID="sch2"/>
</rdf:Description>

<rdf:Description rdf:nodeID="sch2">
   <rdf:first rdf:resource="http://example.org/students/Mohamed"/>
   <rdf:rest rdf:nodeID="sch3"/>
</rdf:Description>

<rdf:Description rdf:nodeID="sch3">
   <rdf:first rdf:resource="http://example.org/students/Johann"/>
   <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
</rdf:Description>

</rdf:RDF>
```

Stavki višjega reda

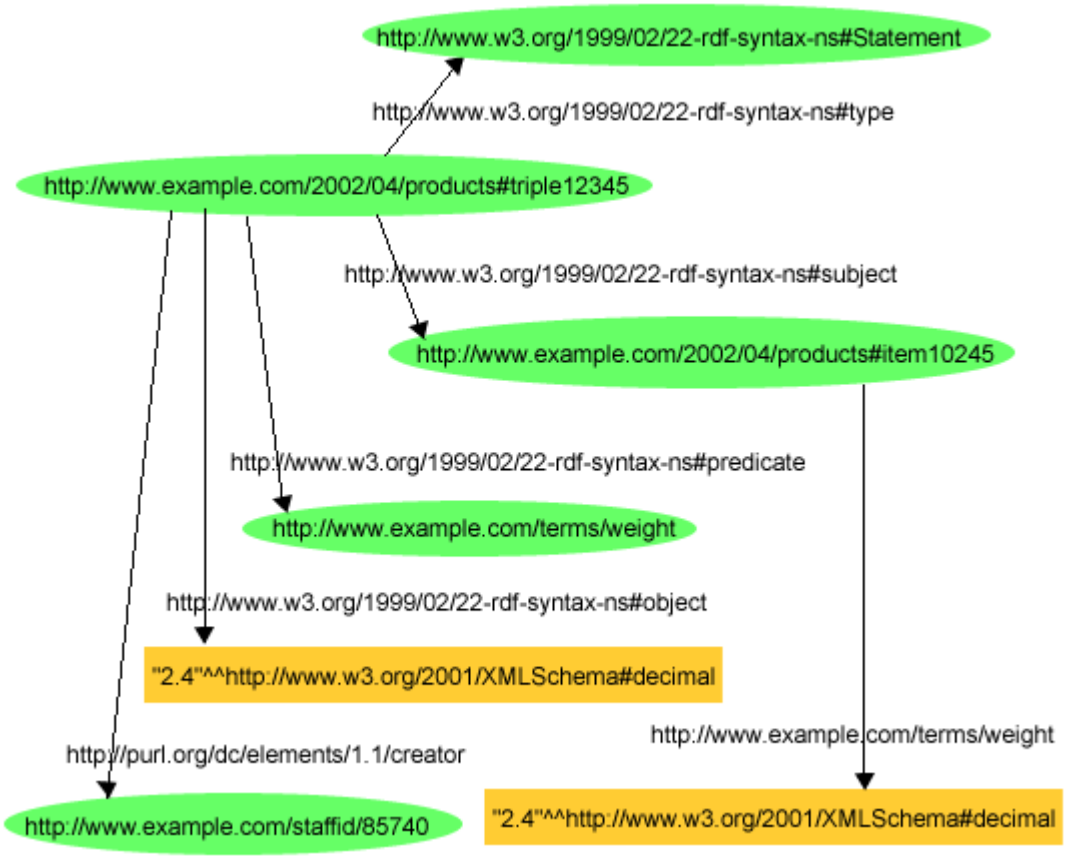
- RDF stavki o RDF stavkih
  - primer: “Tone misli, da splet vsebuje bilijon dokumentov”
- Stavki višjega reda
  - Opišemo lahko lastnosti o lastnostih
  - Pomembni za modele zaupanja, digitalne signature, itd.
  - Meta-podatki
  - Tudi meta-podatki o meta-podatkih
  - RDF lahko opišemo v RDF-ju

⇒ “reification”

Reifikacija

- Shranjevanje podatkov o drugih stavkih
- Kdo je kreiral stavek, kdaj, …
- RDF ima vgrajen slovar za opisovanje stavkov
- Tip: rdf:Statement
  - Definiramo stavek, ki bo služil kot višjenivojski koncept
  1. Definiramo stavek
  2. Dodamo dodatne lastnosti
- Lastnosti rdf:Statement za definicijo stavka: rdf:subject,rdf:predicate,rdf:object

Reifikacija




```
exproducts:triple12345 rdf:type rdf:Statement .
exproducts:triple12345 rdf:subject exproducts:item10245 .
exproducts:triple12345 rdf:predicate exterms:weight .
exproducts:triple12345 rdf:object "2.4"^^xsd:decimal .
exproducts:triple12345 dc:creator exstaff:85740 .
```

Reifikacija

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [<!ENTITY xsd ttp://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:dc="http://purl.org/dc/elements/1.1/"
     xmlns:exterms="http://www.example.com/terms/"
     xml:base="http://www.example.com/2002/04/products">

<rdf:Description rdf:ID="item10245">
   <exterms:weight rdf:datatype="&xsd;decimal">2.4</exterms:weight>
</rdf:Description>

<rdf:Statement rdf:about="#triple12345">
   <rdf:subject rdf:resource="http://www.example.com/2002/04/products#item10245"/>
   <rdf:predicate rdf:resource="http://www.example.com/terms/weight"/>
   <rdf:object rdf:datatype="&xsd;decimal">2.4</rdf:object>
   <dc:creator rdf:resource="http://www.example.com/staffid/85740"/>
</rdf:Statement>

</rdf:RDF>
```

RDF shema

- RDF provides a way to express simple statements about resources, using named properties and values
- Ability to define the vocabularies (terms) describing
  - specific kinds or classes of resources,
  - specific properties in describing those resources
  - Example:
    - company example.comn would describe classes as exterms:Tent that use properties exterms:model, exterms:weightInKg, and exterms:packedSize
    - QNames with “example” namespace prefixes …
- RDF schema provides facilities needed to describe such classes and properties
  - Type system for RDF

RDF shema

- Part of knowledge representation language (KR)
  - “Type system for RDF”
  - Object-oriented model
  - Describing concepts and concept hierarchies
    - Classes (concepts) are instances of rdf:Class
    - Defining taxonomies, ontologies
  - Describing properties and hierarchies of predicates
    - Associations between predicates and classes

RDF shema

- RDF Schema facilities
  - Provided in the form of an RDF vocabulary
  - URIrefs with prefix
    - http://www.w3.org/2000/01/rdf-schema#
    - conventionally associated with the QName prefix rdfs:
    - rdfs:Class, rdfs:subClassOf, rdf:Property, rdfs:subPropertyOf
  - RDF schemas are legal RDF graphs !
- RDF Schema semantics
  - Additional level of software
    - Implementing semantics of RDF Schema
    - Systems: Ontology browsers, RDF DBMSs, ...

Describing Classes

- Identifying concepts to be described
  - Concept, class, category, …
  - People, animals, Web pages, companies, …
- Class description
  - Resources having rdf:type predicate with range rdfs:Class
  - Concepts rdfs:Class and rdfs:Resource,
  - Properties rdf:type and rdfs:subClassOf
  - Classes have instances

Describing Classes

- Example:
  - Information about different kinds of motor vehicles
  - Concept representing motor vehicles
    - URIref `ex:MotorVehicle`

```
ex:MotorVehicle   rdf:type   rdfs:Class .
```

    - `ex:` = URIref http://www.example.org/schemas/vehicles
    - `rdf:type` indicates a resource is an instance of a class
    - Now `exthings:companyCar` is an instance of `ex:MotorVehicle`

```
exthings:companyCar   rdf:type   ex:MotorVehicle .
```

    - `rdfs:Class` itself has an `rdf:type` of `rdfs:Class`

Describing Classes

- Example:
  - classes representing various specialized kinds of motor vehicle
    - passenger vehicles, vans, minivans, …

```
ex:Van     rdf:type   rdfs:Class .
ex:Truck   rdf:type   rdfs:Class .
```

    - indicate their special relationship to class `ex:MotorVehicle`
    - `rdfs:subClassOf` property to relate the two classes

```
ex:Van   rdfs:subClassOf   ex:MotorVehicle .
```

    - any instance of class `ex:Van` is also an instance of class `ex:MotorVehicle`

Describing Classes

- Example:

```
exthings:companyVan  rdf:type ex:Van .
```

  - Inference:
    - `exthings:companyVan` is an instance of `ex:Van`
    - `ex:Van` is sub-class of `ex:MotorVehicle`
    - Therefore, `exthings:companyVan` is an instance of `ex:MotorVehicle`
  - Software must be written to “understand” RDFS
    - Inferences may be useful for explanations, reasoning, …
  - `rdfs:subClassOf` property is transitive

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

Describing Classes

- Example:
  - Inference:
    - `ex:MiniVan` is also a subclass of `ex:MotorVehicle`
    - Instances of `ex:MiniVan` iare also the instances of `ex:MotorVehicle`
    - … as well as being instances of class `ex:Van`
    - `ex:MiniVan` may be a subclass of both `ex:Van` and `ex:PassengerVehicle`
  - Statements (from RDF Primer):
    - RDF Schema defines all classes as subclasses of class `rdfs:Resource`
    - … since the instances belonging to all classes are resources ??

Vehicle Class Hierarchy

```
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 .
```

Vehicle Class Hierarchy

Vehicle Class Hierarchy

```
<?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:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://example.org/schemas/vehicles">
<rdfs:Class rdf:ID="MotorVehicle"/>
<rdfs:Class rdf:ID="PassengerVehicle">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Truck">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Van">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class>
<rdfs:Class rdf:ID="MiniVan">
  <rdfs:subClassOf rdf:resource="#Van"/>
  <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
</rdfs:Class>
</rdf:RDF>
```

An Instance of `ex:MotorVehicle`

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
  xmlns:ex="http://example.org/schemas/vehicles#"
  xml:base="http://example.org/things">

  <ex:MotorVehicle rdf:ID="companyCar"/>
</rdf:RDF>
```

Describing Properties

- Describe specific properties that characterize classes of things
  - Properties are described using
    - RDF class rdf:Property,
    - RDFS properties `rdfs:domain`, `rdfs:range`, and `rdfs:subPropertyOf`
  - All properties are instances of class `rdf:Property`

```
exterms:weightInKg   rdf:type   rdf:Property .
```

  - Describing how properties and classes are intended to be used together
    - `rdfs:range` and `rdfs:domain`

Describing Properties

```
ex:Person   rdf:type      rdfs:Class .
ex:author   rdf:type      rdf:Property .
ex:author   rdfs:range    ex:Person .
```

- `ex:author` is of type rdf:Property
- `rdfs:range`:
  - values of a particular property are instances of a designated class (e.g., `ex:Person`)
- Property can have zero, one, or more than one range property
  - If the property has no range than nothing has been said about the value of property
  - … one range class than values of a propetry are instances of the range class

Describing Properties

- … more than one range class than values of a property are instances of **all** range class

```
ex:hasMother   rdfs:range   ex:Female .
ex:hasMother   rdfs:range   ex:Person .
```

- property `ex:hasMother` has the two ranges `ex:Female` and `ex:Person`
  - Value of `ex:hasMother` has to be an instance of `ex:Female` and `ex:Person`

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

  - Value of a property can be given by a typed literal

```
ex:age   rdf:type      rdf:Property .
ex:age   rdfs:range    xsd:integer .
```

  - datatype `xsd:integer` is identified by its URIref

Describing Properties

- Specifying that URIref is a datatype

```
xsd:integer   rdf:type   rdfs:Datatype .
```

- Class `rdfs:Datatype` is part of RDFS
- Above statement does not constitute a definition

- Property `rdfs:domain` indicates that a particular property applies to a designated class

```
ex:Book      rdf:type       rdfs:Class .
ex:author    rdf:type       rdf:Property .
ex:author    rdfs:domain    ex:Book .
```

- A property may have zero, one, or more than one domain property

Describing Properties

- If a property has no domain property, then nothing is said about the resources that propery describes
- If a property has one domain property, then instances of range class have a given property
- If a property has more than one domain property, then values are instances of all range classes

```
exterms:weight   rdfs:domain   ex:Book .
exterms:weight   rdfs:domain   ex:MotorVehicle .
```

- `exthings:companyCar` is both an instance of `ex:Book` and of `ex:MotorVehicle`

```
exthings:companyCar   exterms:weight   "2500"^^xsd:integer.
```

Describing Properties

- Extension of the vehicle schema
  - adding two properties `ex:registeredTo` and `ex:rearSeatLegRoom`
  - new class `ex:Person`, and
  - describing the datatype `xsd:integer` as a datatype

```
<rdf:Property rdf:ID="registeredTo">
  <rdfs:domain rdf:resource="#MotorVehicle"/>
  <rdfs:range rdf:resource="#Person"/>
</rdf:Property>

<rdf:Property rdf:ID="rearSeatLegRoom">
  <rdfs:domain rdf:resource="#PassengerVehicle"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>

<rdfs:Class rdf:ID="Person"/>

<rdfs:Datatype rdf:about="&xsd;integer"/>
```

Describing Properties

- RDF Schema provides a way to specialize properties
  - predefined `rdfs:subPropertyOf` property
- Example:
  - `ex:primaryDriver` is a specialization of `ex:driver`

```
ex:driver           rdf:type              rdf:Property .
ex:primaryDriver    rdf:type              rdf:Property .
ex:primaryDriver    rdfs:subPropertyOf    ex:driver .
```

- Meaning of `rdfs:subPropertyOf` relationship
  - If an instance `exstaff:fred` is an `ex:primaryDriver` of the instance `ex:companyVan`,
  - then RDF Schema defines `exstaff:fred` as also being an `ex:driver` of `ex:companyVan`

Describing Properties

- More properties for the vehicle schema

```
<rdf:Property rdf:ID="driver">
  <rdfs:domain rdf:resource="#MotorVehicle"/>
</rdf:Property>

<rdf:Property rdf:ID="primaryDriver">
  <rdfs:subPropertyOf rdf:resource="#driver"/>
</rdf:Property>
```

- A property may be a subproperty of zero, one or more properties
- `rdfs:range` and `rdfs:domain` properties that apply to an RDF property also apply to each of its subproperties
  - `ex:primaryDriver` has an rdfs:domain of `ex:MotorVehicle`

Full Vehicle Schema

```
<?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:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://example.org/schemas/vehicles">
<rdfs:Class rdf:ID="MotorVehicle"/>
<rdfs:Class rdf:ID="PassengerVehicle">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Truck">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Van">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdfs:Class>
<rdfs:Class rdf:ID="MiniVan">
  <rdfs:subClassOf rdf:resource="#Van"/>
  <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Person"/>
<rdfs:Datatype rdf:about="&xsd;integer"/>
<rdf:Property rdf:ID="registeredTo">
  <rdfs:domain rdf:resource="#MotorVehicle"/>
  <rdfs:range rdf:resource="#Person"/>
</rdf:Property>
<rdf:Property rdf:ID="rearSeatLegRoom">
  <rdfs:domain rdf:resource="#PassengerVehicle"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
<rdf:Property rdf:ID="driver">
  <rdfs:domain rdf:resource="#MotorVehicle"/>
</rdf:Property>
<rdf:Property rdf:ID="primaryDriver">
  <rdfs:subPropertyOf rdf:resource="#driver"/>
</rdf:Property>
</rdf:RDF>
```

Instance of `ex:PassengerVehicle`

```
<?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:ex="http://example.org/schemas/vehicles#"
            xml:base="http://example.org/things">

  <ex:PassengerVehicle rdf:ID="johnSmithsCar">
       <ex:registeredTo rdf:resource="http://www.example.org/staffid/85740"/>
       <ex:rearSeatLegRoom
           rdf:datatype="&xsd;integer">127</ex:rearSeatLegRoom>
       <ex:primaryDriver rdf:resource="http://www.example.org/staffid/85740"/>
  </ex:PassengerVehicle>
</rdf:RDF>
```

Viri

- http://www.w3.org/TR/rdf-primer/
- http://www.w3.org/TR/rdf-mt/