RDF
Resource Description Framework

G. Falquet
2014

Contents

A Graph Model for KR

RDF graphs express knowledge about resources
- a resource is anything that can be identified (a web page, a person, a country, an abstraction, ...)

The basic unit of knowledge is the triple

(subject, predicate, object)

It represents the fact that a relation (predicate) holds between the subject and the object.

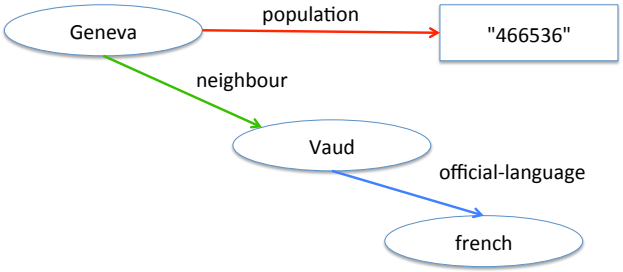
(Geneva population "312445")
The population of Geneva is 312445

(Geneva neighbour Vaud)
Geneva has a neighbour Vaud

(Vaud official-language french)
The official language of Vaud is French

Triples are Graph Edges

➡ (Geneva population "312445")
➡ (Geneva neighbour Vaud)
➡ (Vaud official-language french)



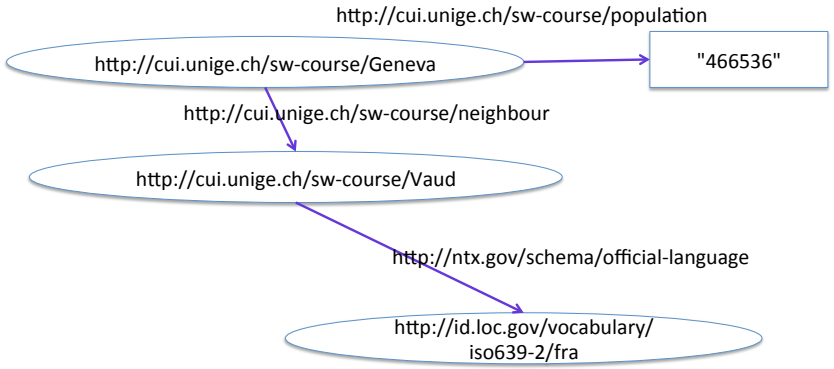
Concrete resources

- Resources are represented by their Uniform Resource Identifier (URI)
  - http://cui.unige.ch
  - http://www.unige.ch/icle/Projects/NZR.html
  - mailto://hk@jmail.me

  not necessarily web documents

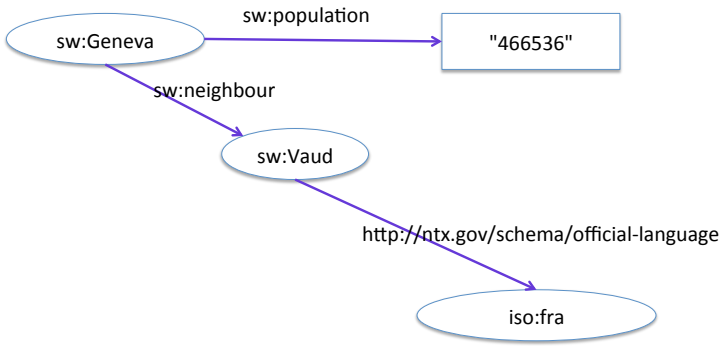
- A RDF graph can refer to any resource

Example concretized



Using prefixes to create compact URIs

prefix sw: http://cui.unige.ch/sw-course/
prefix iso: http://id.loc.gov/vocabulary/iso639-2/



Literals

A lexical form that identifies a value in a value space

strings

"value"

string in a specific language

"value"@language

typed value

"value"^^type

Examples

prefix xsd: http://www.w3c.org/2001/XMLSchema#

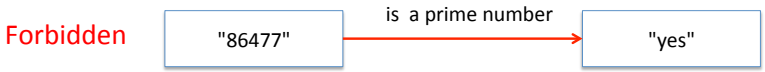
"Scrabble"
"vi povas legi ĉi tiun tekston"@eo
"567"^^xsd:number
"true"^^xsd:boolean
"2002-10-10T12:00:00+02:00"^^xsd:dateTime

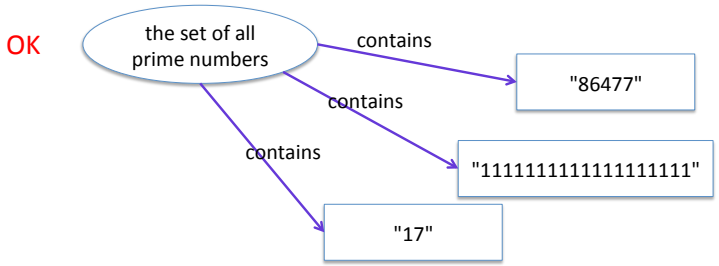
XML builtin datatypes are of common use, but not mandatory

prefix my: http://cui.unige.ch/TypeSystem#

"4.5+3i+2j-5k"^^my:quaternion

Remark. A literal may not be the subject of a triple (values cannot be described, they are supposed to be known)





Exercises

1. Draw an RDF graph that represents the following situation
- Bob has a cat. The name of this cat is Felix and he is 6 years old. Felix has two friends: Tiger and Einstein.

2. Add the facts
- Bob is married with Alice since 2008-08-01
- Bob has two other cats

Practical syntax for RDF

How to represent a RDF graph with characters (in a text file)

RDF data can be expressed with different notations

- XML (for machine interchange)
- N3 and Turtle (human readable)

XML Syntax

Principle: there are node elements and property elements

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:sw="http://cui.unige.ch/sw-course/">
<rdf:Description rdf:about="http://cui.unige.ch/sw-course/Geneva">
   <sw:population>466536</sw:population>
   <sw:neighbour>
      <rdf:Description
         rdf:about="http://cui.unige.ch/sw-course/Vaud">
      </rdf:Description>
   </sw:neighbour>
</rdf:Description>
 ...
</rdf:RDF>
```


N3 notation

An N3 file has

1. prefix definitions
2. triples or abbreviated triples

```
@prefix sw: <http://cui.unige.ch/sw-course/> .
@prefix xsd: <http://www.w3c.org/2001/XMLSchema#> .
sw:Geneva sw:population "466536"^^xsd:integer .
sw:Geneva sw:neighbour sw:Vaud .
sw:Vaud sw:official-language <http://id.loc.gov/vocabulary/iso639-2/
fra> .
```


Abbreviations

subject $pred_1$ obj_1 ; $pred_2$ obj_2 ; ... ; $pred_n$ obj_n .

for

subject $pred_1$ obj_1 . subject $pred_2$ obj_2 . ... . subject $pred_n$ obj_n .

```
sw:Geneva
   sw:population "466536"^^xsd:integer ;
   sw:neighbour sw:Vaud .
```

Abbreviations

subject predicate obj_1 , obj_2 , ... , obj_n .

for

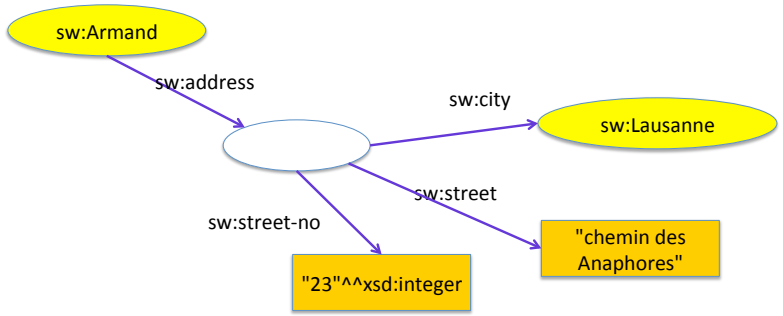
subject predicate obj_1 . subject predicate obj_2 . ... . subject predicate obj_n .

```
sw:Vaud sw:neighbour
  sw:Geneva , sw:Fribourg , sw:Valais ,
  sw:Neuchatel , sw:Bern .
```


Blank nodes

- Nodes that are anonymous, not identified by a URI
- Only locally identified

"The address of Armand is 23 chemin des Anaphores, Lausanne"



In N3, with the _: prefix

```
@prefix sw: <http://cui.unige.ch/sw-course/> .
@prefix xsd: <http://www.w3c.org/2001/XMLSchema#> .

sw:Armand sw:address _:aa .
_:aa sw:street "chemin des Anaphores" .
_:aa sw:street-no "23"^^xsd:integer .
_:aa sw:city sw:Lausanne .

_:aa acts like an internal variable, within the RDF file/graph.
It is invisible from the outside (no URI).

Possible abbreviation:
sw:Armand sw:address
  [sw:street "chemin des Anaphores" ;
   sw:street-no "23"^^xsd:integer ;
   sw:city sw:Lausanne] .
```

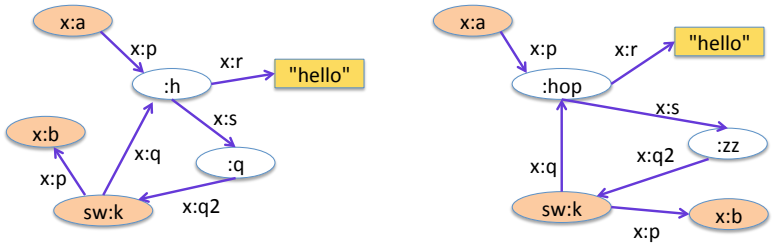

In XML: *nodeID* instead of *about*

```
<rdf:Description rdf:about="http://sw.unige.ch/Armand"
   <sw:address rdf:nodeID="abc"/>
</rdf:Description>

<rdf:Description rdf:nodeID="abc">
   <sw:street>chemin des Anaphores </sw:street>
   <sw:street-no
      rdf:datatype="http://www.w3.org/2001/XMLSchema#int>
       23
   </sw:street-no>
   <sw:city>
      <rdf:Description about="http://sw.unige.ch/Lausanne" />
   </sw:city>
</rdf:Description>
```

Graph equivalence

- The internal identifiers of blank node are interchangeable
- Two RDF graphs have the same meaning if their only differences are the blank node identifiers.



Graph equivalence

The official definition

Two RDF graphs G and G' are equivalent if there is a bijection M between the sets of nodes of the two graphs, such that:

- M maps blank nodes to blank nodes.
- $M(lit)=lit$ for all RDF literals lit which are nodes of G.
- $M(uri)=uri$ for all RDF URI references uri which are nodes of G.
- The triple (s, p, o) is in G if and only if the triple ($M(s)$, p, $M(o)$) is in G'

In fact, M shows how each blank node in G can be replaced with a new blank node to give G'.

RDF standard vocabuary

A standard vocabulary for defining

- resource typing
- data structures (containers and collections)
- RDF graph schemas
  - resource classification (schemas)
  - constraints on properties

This vocabulary has URIs of the form

http://www.w3.org/1999/02/22-rdf-syntax-ns#*name*

the usual prefix definition is

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

rdf:type

Assign a type to a resource

- Felix *is a* cat and Joe *is a* mouse

  ex:Felix rdf:type ex:Cat
  ex:Joe rdf:type ex:Mouse

- My car is red (a set-theoretical view)

  ex:myCar rdf:type ex:RedThings

  ex:myCar ex:color "red" (generally a better choice)

Containers

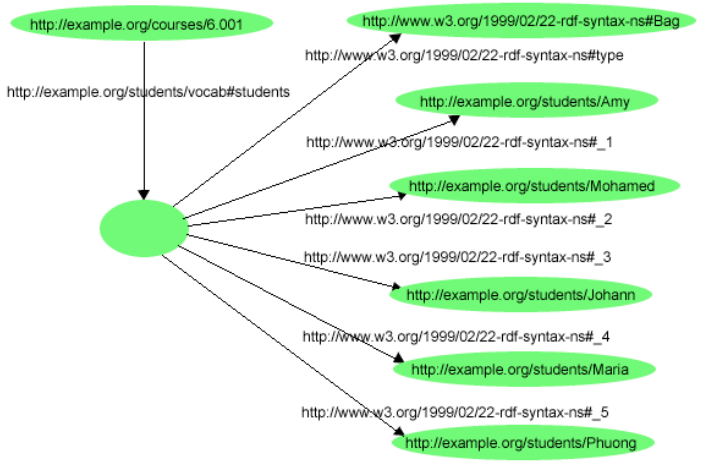
To consider a group of resources as a whole

- assign global properties to the group

Three types of containers

- rdf:Bag (a set with repetitions)
- rdf:Seq (an ordered set)
- rdf:Alt (represents choices)

Properties rdf:_1, rdf:_2, rdf:_3, ... to link a container with its first, second, third, ... member.



http://www.w3.org/TR/2004/REC-rdf-primer-20040210/#collections

RDF/XML (abbreviation)

```
<?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>
```


Remarks

- Bag, Seq, Alt are indications about the intended meaning
- There is not specific way to "close" a container, i.e. to say that is doesn't have any other member.
  - the Bag of students in the previous example may have more than 5 members, in reality

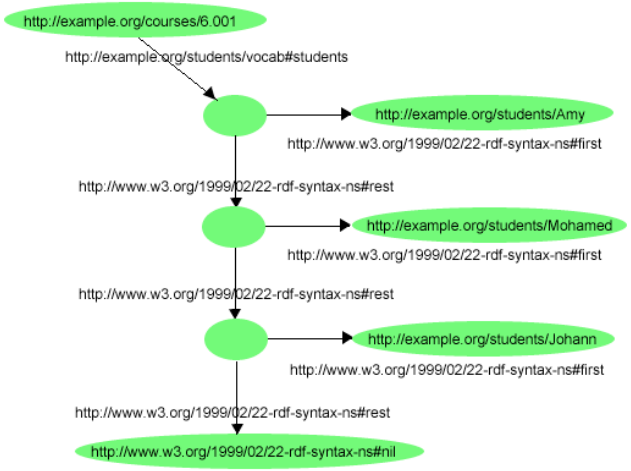
Collections

Closed collections: all the members are known

Use the first/rest representation technique:

A collection is made of
- a first element (any resource)
- a rest, which is a collection

rdf:nil is the empty collection



http://www.w3.org/TR/2004/REC-rdf-primer-20040210/#collections

in RDF/XML

```
<?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>
```

rdf:parseType="Collection" instructs parsers to generate first/rest properties

in N3

(*object1 object2*) is short for:

[rdf:first *object1*; rdf:rest [rdf:first *object2*; rdf:rest rdf:nil]]

( ) is short for the resource:

rdf:nil

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix s: <http://example.org/vocab#> .
@prefix c: <http://example.org/courses/> .
@prefix std: <http://example.org/students/>.

c:6.001 s:students (std:Amy, std:Mohamed, std:Johann) .

Reification

Ralph Swick says that Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila . »

« Albert says that document 345 confirms that Ralph Swick says that Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila ».

Reification

Ralph Swick says that Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila . »

Something like

Ralph Swick says X

X = (http://www.w3.org/Home/Lassila ex:creator "Ora Lassila")

Reification

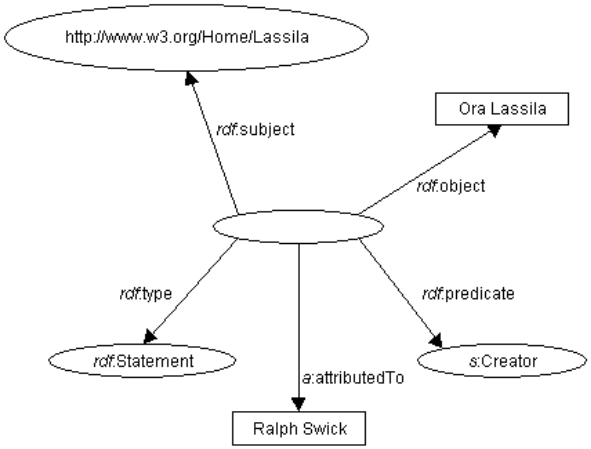
Ralph Swick says that Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila . »

in RDF

X a:attributedTo "Ralph Swick".

X rdf:type rdf:Statement .
X rdf:predicate s:creator .
X rdf:subject http://www.w3.org/Home/Lassila .
X rdf:object "Ora Lassila" .

Reified statement



Albert says that doc342 confirms that Ralph Swick says that Ora Lassila is the creator of the resource.