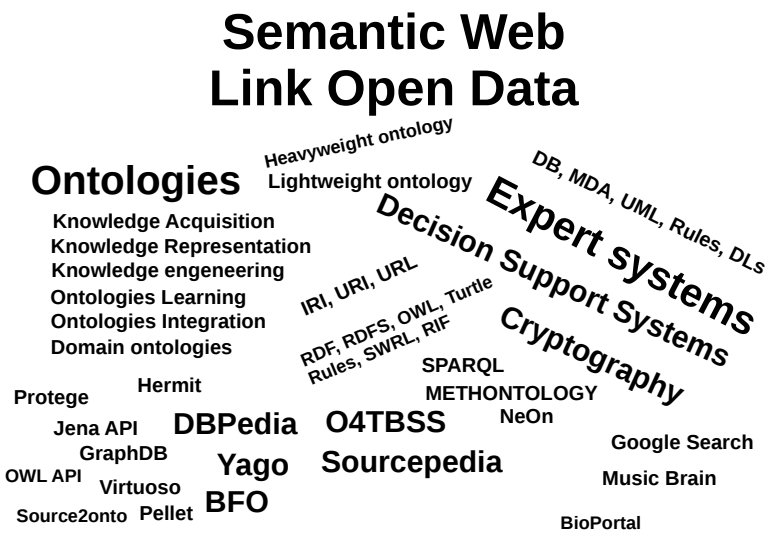
Semantic Web and knowledge engineering

Identify things and represent simple facts



Dr. Azanzi Jiomekong

University of Yaounde I, Department of Computer Science

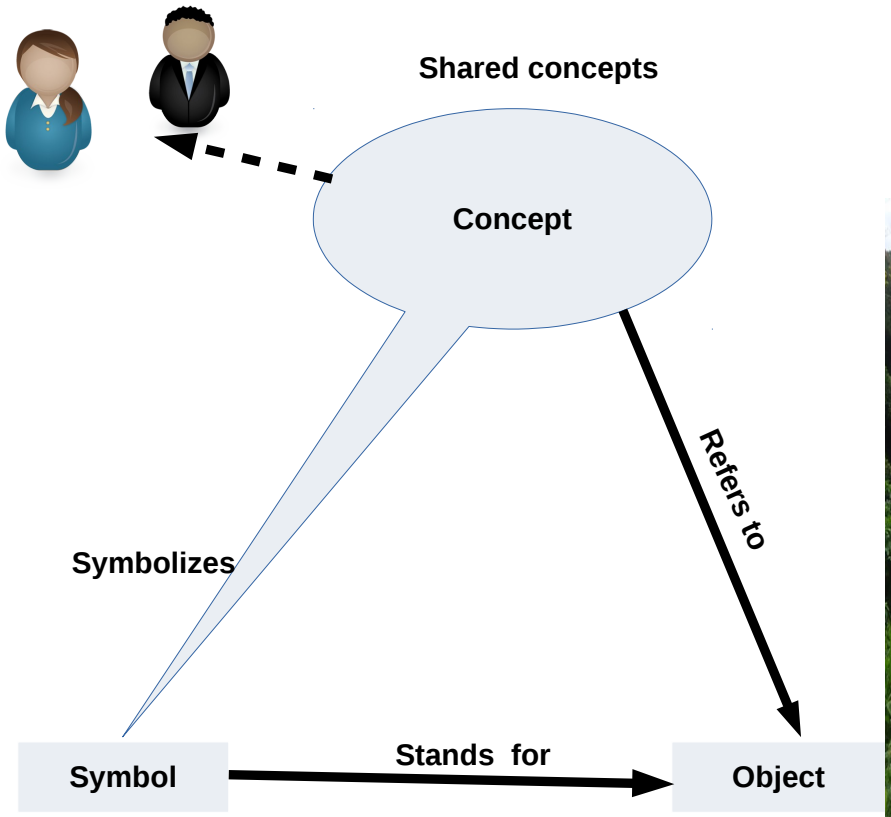
10 avril 2023

Semiotic triangle

个 個 असल

1

Semiotic triangle



Chutes jumelles Ekom Kam

Questions

- How to identify things ?
  Uniform Resource Identifier - URI

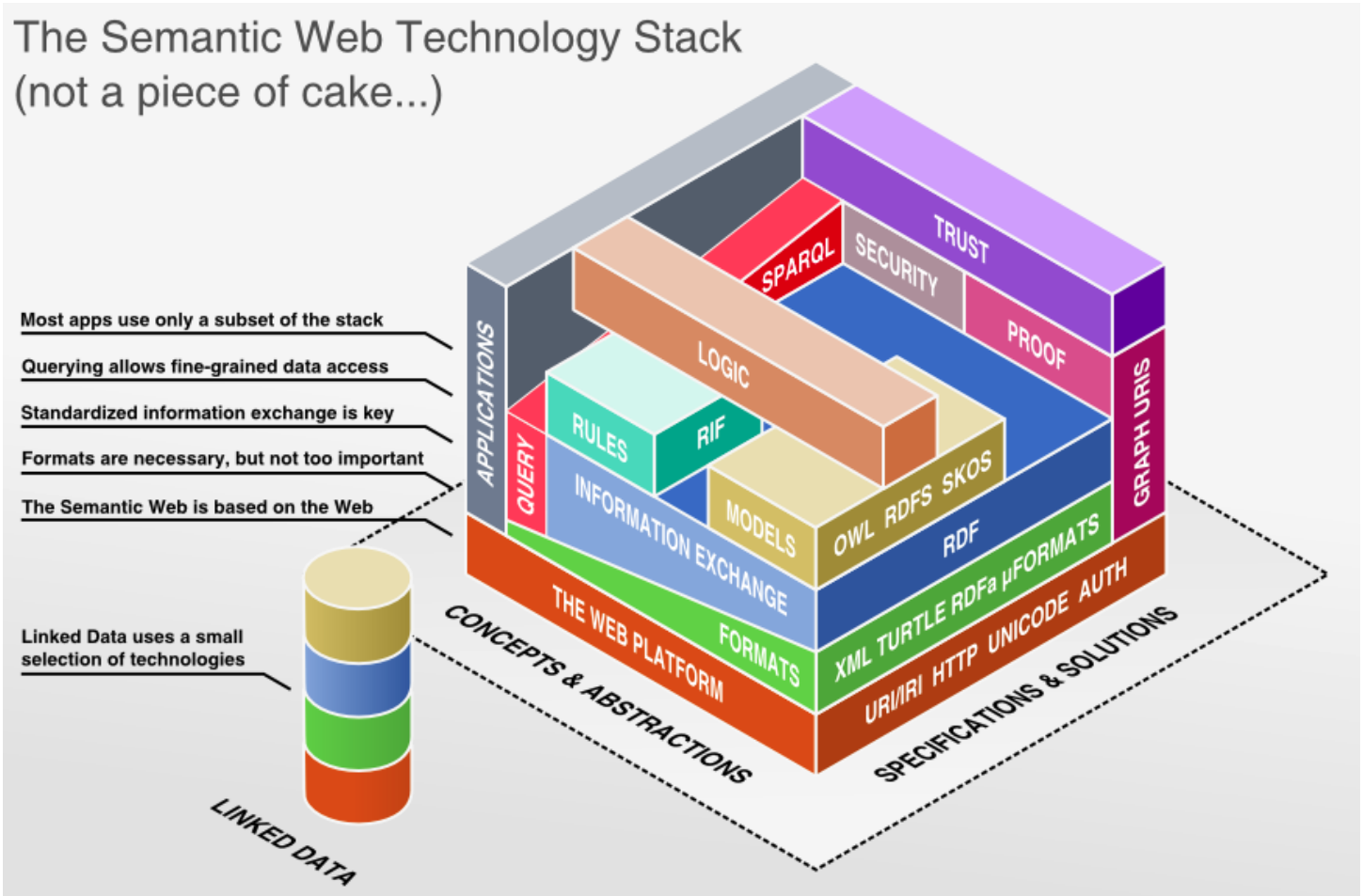
- How to represent things ?
  RDF Graph

- How to encode and represent facts ?
  Graph serialization

The responses to these questions are given in this course

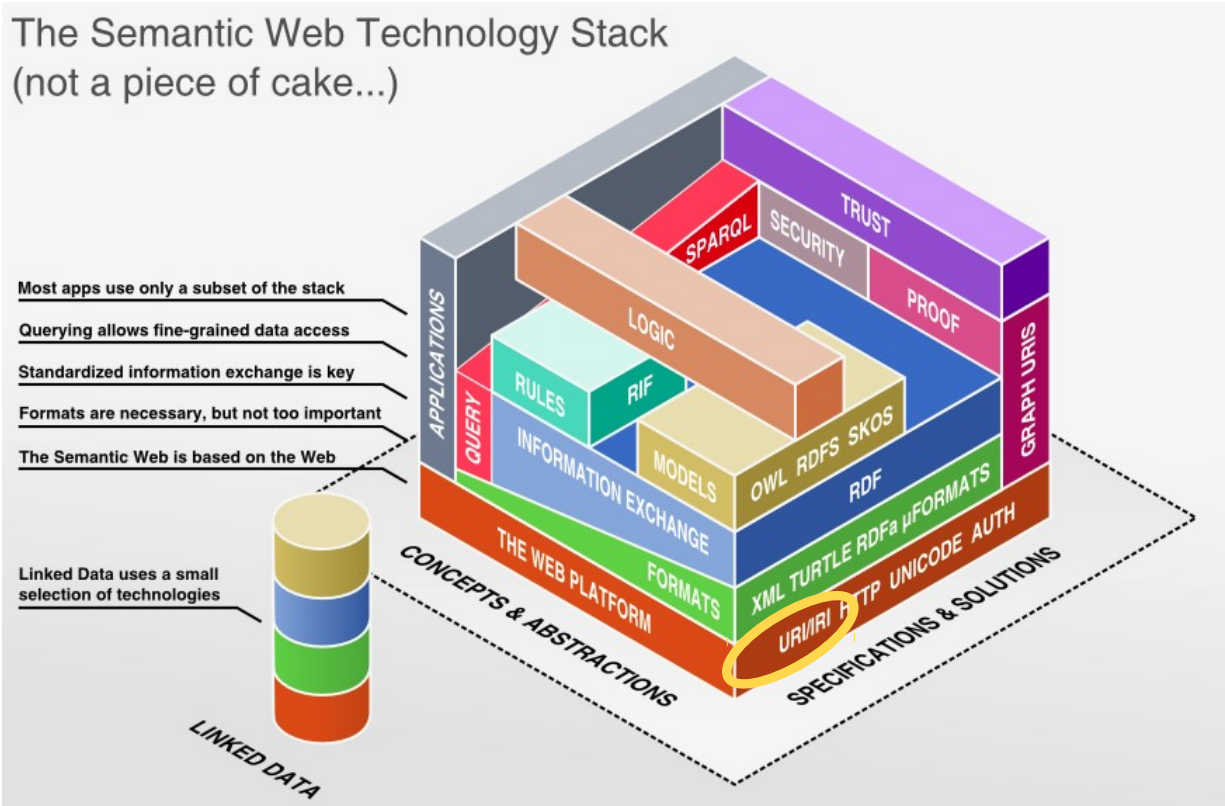
Semantic Web Stack

Represents Semantic Web Basic Architecture



How to identify things ?

Resp = Uniform Resource Identifier



Uniform Resource Identifier

How to identify things ? URI

URI,
Resource and Representation

URI
http://www.ekomKamWaterfall/index.html

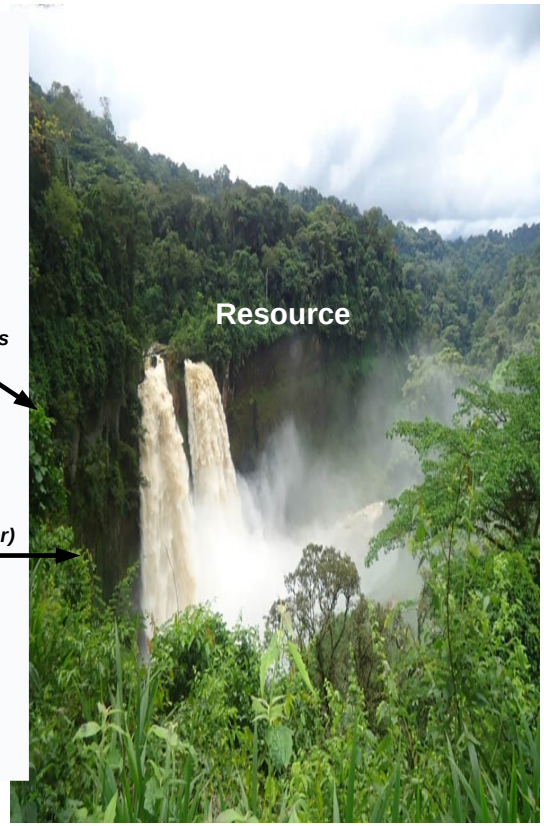
Defines

Metadata:
Content-type: text/html

Data:
<!DOCTYPE html>
<html>
<head>
<title>Ekom Kam Waterfall</title>

…
</body>
</html>

Represents (stands for)



Chutes jumelles Ekom Kam

Uniform Resource Identifier

Representation (HTML) and presentation (Web browser)
Put the previous waterfall in a Web page to have his representation

Uniform Resource Identifier

HTTP Request / Response
Use the previous image to show how HTTP works (with the representation and the presentation of information)

Uniform Resource Identifier

- What is the difference between the picture of a thing and the thing itself ?
- When we talk about a thing $\longrightarrow$ we talk about things having the same property of this thing
- We share a common concepts $\longrightarrow$ we are able to communicate information and understand information
- We must identify things that we are talking about $\longrightarrow$ we need the representation of these things in the Semantic Web

Uniform Resource Identifier

- Simple and extensible schema for worldwide unique identification of abstract physical resources (RFC 3986)

- Uniform : different types of resources identifiers all constructed according to a uniform schema

- Resource :
  - Can be every object with a clear identity (according to the content of the application)
  - Whatever may be identified via URI
  - e.g., web pages, books, locations, persons, relations among objects, abstract concepts, etc.

- Identifier : to distinguish one resource from another

Uniform Resource Identifier

- URI = 2 things : Address and Identity
- Uniform : we can different types of resources
- Identifier : we can distinguish one resource from another in a unique way
- Representation of information :
  - We have a representation of a thing in the web page
  - We have the metadata of the representation of this thing
  - The browser transform this representation in a format that a human can understand
- When a user send a request, to a server, the representation of what he wants is delivered

Uniform Resource Identifier

Designator and Designatum

- *The distinguishing characteristics of [information] resources is that all of their essential characteristics and can be conveyed in a messages* - W3C : Architecture of the WWW, volume One
- A resource can be describeb (designated) via **Metadata**
- Even if the resource itself cannot be delivered from the web server, probably its representation might be available that describes the resource suffifiently

Uniform Resource Identifier

URI,
Resource and Representation

URI

http://www.ekomKamWaterfall/index.html

Defines

Representation

Metadata:
Content-type: text/html

Data:
<!DOCTYPE html>
<html>
<head>
<title>Ekom Kam Waterfall</title>

…
</body>
</html>

Represents (stands for)

Describe **Designate**

Information about Ekom Kam Waterfall

Resource:
Ekom Kam Waterfall
Metadata

Uniform Resource Identifier

- Combines :
  - Address (Locator)
    - Uniform Resource Locator (URL, RFC 1738)
    - Denotes where a resource can be found in the Web by stating its primary access mechanism
    - Might change during life cycle
  - Identity (Name) :
    - Uniform Resource Name (URN, RFC 2141)
    - Persistent identifier for a Web resource
    - Remains unchanged during life cycle

Uniform Resource Identifier

Generic Syntax

- US-ASCII Encoding
- Percent Encoding for reserved characters, or characters that do not exist in US-ASCII encoding
  pct-encoded=”%”HEXDIG HEXDIG
- Reserved characters with special function
  reserved = gen-delims/sub-delims
  gen-delims=” : ” /” /” /” ?” /” #” /” [” /”]” /” @”
  sub-delims=”!” /”$” /”&” /”’” /”(” /”)” /” ∗” /” +” /”,” /”;” /” =”
- Permitted characters
  unreserved = *ALPHA*/*DIGIT* /” −” /”.” /”_” /” ”
- Extension to Universal Character Code (Unicode/ISO 10646) International Resource Identifier (IRI, RFC 3987), e.g., http address in Japanese language

Uniform Resource Identifier

Generic syntax

schema" ://[userinfo"@"]host[:port][path][" ?"query]["#"fragment]

- schema : e.g. http, ftp, mailto, etc.
- userinfo : e.g. username :password
- host : e.g. Domain-Name, IPV4/IPV6 Address
- port : e.g. 80 for standard HTTP port
- path : e.g. path in file system of www server
- query : e.g. parameters to be passed over to applications
- fragment : e.g. determines a specific fragment of a document

Uniform Resource Identifier

In the Semantic Web

- What if a URI for a resource does not exist ?
  - Define a URI by yourself :
    - avoid overlaps $\longrightarrow$ use your own website
    - enables documentation at the same place ($\longrightarrow$Content negociation)
  - Separate URI for resource (Designatum) and its documentation (Designator) via URI references (i.e. via "#" fragments) or content negociation

`http://www.facsciences-uy1/azanzi.foaf.rdf#me`

Uniform Resource Identifier

- Already establish in various domains :
  - The Web : URL, PRN
  - Books and publications : ISBN, ISSN
  - Digital Object Identifier : DOI

Knowledge

Facts, information or experience

How to represent simple facts ?

Example : How to represent the fact : "Dr. Jiomekong has the email fidel.jiomekong@facsciences-uy1@uninet.cm" ? :

Response :

- Natural Language representation
- Graph representation
- XML Serialization

Knowledge representation

A simple example with XML

How do I represent: ‘’Dr. Jiomekong has the email fidel.jiomekong@facsciences-uy1.uninet.cm ?

```
<emailaddr>
      <owner> Dr. Jiomekong </owner>
      <email>fidel.jiomekong@facsciences-uy1.uninet.cm</email>
</emailaddr>
```

```
<person name= ‘’Dr. Jiomekong’’>
      <owner> Dr. Jiomekong </owner>
      <email>fidel.jiomekong@facsciences-uy1.uninet.cm</email>
</person>
```

```
<person name= ‘’Dr. Jiomekong’’ email= fidel.jiomekong@facsciences-uy1.uninet.cm/>
```

Knowledge representation

A simple example with Natural Language

How do I represent: ''Dr. Jiomekong has the email fidel.jiomekong@facsciences-uy1.uninet.cm ?

Dr. Jiomekong

Has the email

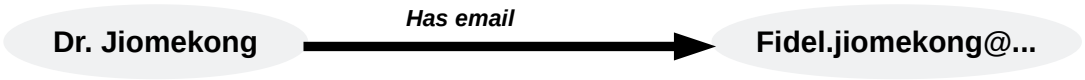
fidel.jiomekong@...

Knowledge representation

A simple example with a graph

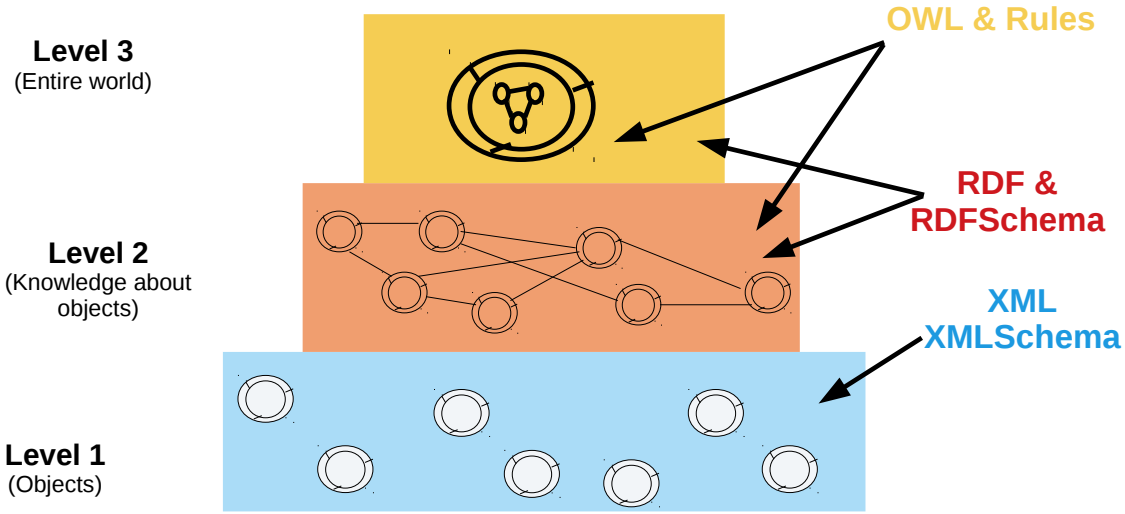
How do I represent: ''Dr. Jiomekong has the email fidel.jiomekong@facsciences-uy1.uninet.cm ?

Simple and intuitive knowledge representation with **directed graphs**



Knowledge representation

In the Semantic Web



Knowledge representation

In the Semantic Web

Three steps of Semantic knowledge representation : XML & XMLSchema, RDF & RDFSchema, OWL & Rules :

- XML & XMLSchema :
  - Available in the Web
  - Used to describe simple objects
  - Are not connected to each other
  - It is difficult to decide if they represent the same or difference
- RDF & RDFSchema :
  - Give the possibility to define relations among objects
  - First knowledge, first facts that we can express
  - Used to take objects into relations
  - Used to construct little world that is open and based on RDF and RDFS

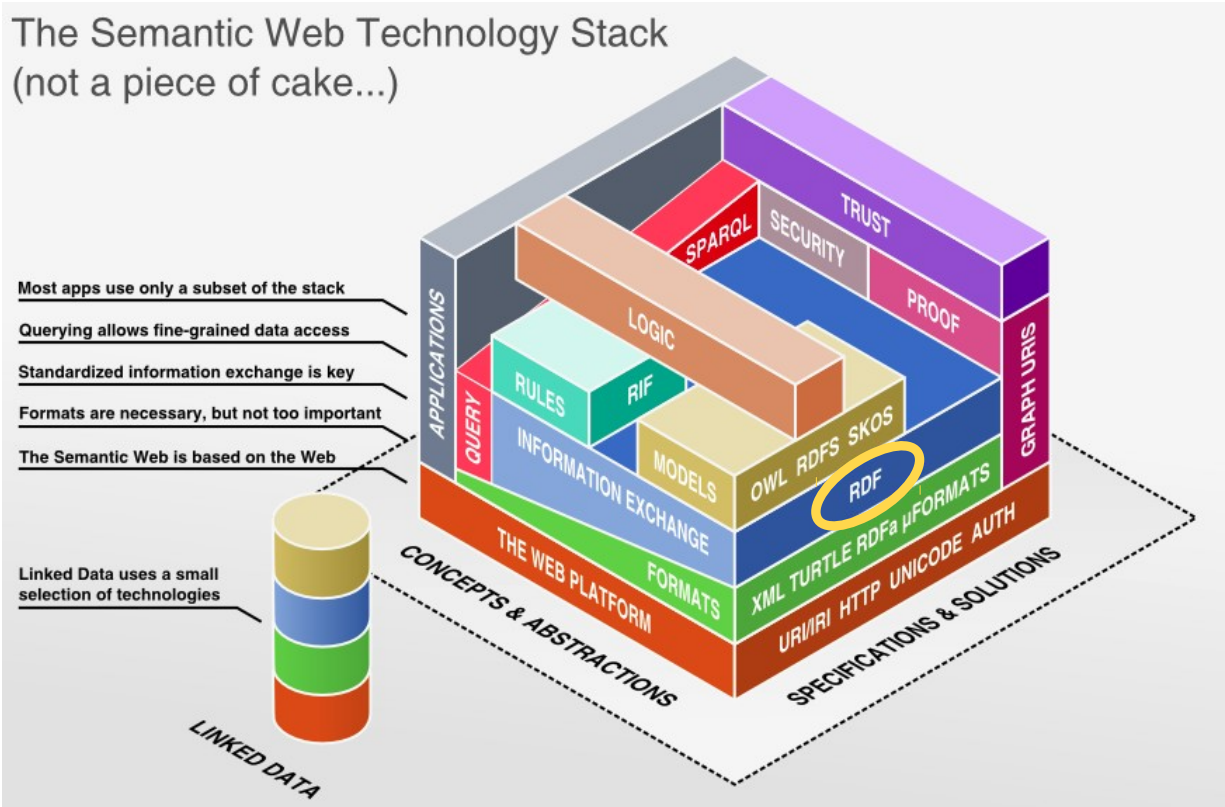
Knowledge representation

In the Semantic Web

Three steps of Semantic knowledge representation : XML & XMLSchema, RDF & RDFSchema, OWL & Rules :

- OWL & Rules :
  - Use if we want to close our world
  - Used to define entire world with things inside that world
  - Enable the Semantic expressivity with OWL and Rules
  - Used to formulate axioms that contains knowledge, restrictions, constrainst about things we are modeling

Resource Description Framework



Resource Description Framework

- An RDF Model is a set of RDF statements
- Statement :=(subject, property, object)
- Subject is a resource
- Property is a resource
- Object is either a resource or a literal
- Knowledge is expressed as a list of statements
- All RDF statements follow the same simple schema

Resource Description Framework

Resource

- Can be everything
- Must be uniquely identified and be referencable
- ...simply via URI

All objects that can be addressed via URI

Resource Description Framework

Description

- =Description of resource
- ...via representing properties and relationships among resources
- ...relationships can be represent as graphs

Properties / attributes for the description of ressources

Resource Description Framework

Framework

- =Combination of Web based protocols (URI, HTTP, XML, ...)
- Based on formal model (semantics)
- Defines all allowed relationships among resources

Resource Description Framework

How do I represent: ‘’Dr. Jiomekong has the email fidel.jiomekong@facsciences-uy1.uninet.cm ?

Dr. Jiomekong **Subject**

Has the email **Property**

fidel.jiomekong@... **Object**

Resource Description Framework

- RDF
- Subject - Property - Value (the value of that property)
- The value of the property can be another subject or a value
- Constituents of RDF language : Resource, literal and blank nodes
- Several representation of RDF language :
  - Node-Edge-Node Triple : graph representation
  - N3 Notation
  - Turtle (Terse RDF Triple Language) - simplification of N3
  - RDF XML-Serialization - is the standard

Resource Description Framework

Example of a statement

Suppose I want to make the following statement :
This is a lecture and this lecture has a name which is "Semantic Web and Applications". The property of this lecture is the lecture name and the lecture name contains a certain amount of time comprises four hours per week (Property for the duration).

Represent this statement with the different formalism

Resource Description Framework

Constituents of RDF graph

- **Resources :**
  Objects that can be addressed via **URI**

- **Properties :**
  Attributes for the description of resources

- **Statements(RDF-Triple) :**

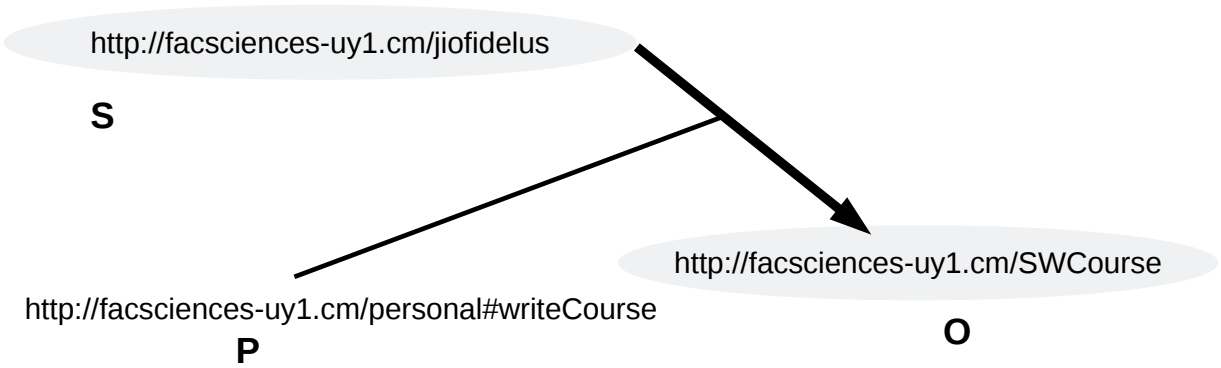
Resource	+	Property	+	Object/Value
URI		URI		URI/Literal

Resource Description Framework

Constituents of RDF graph : URI

How do I represent: ‘’Dr. Jiomekong write the course Semantic Web and applications ?

Used to reference resources uniquely

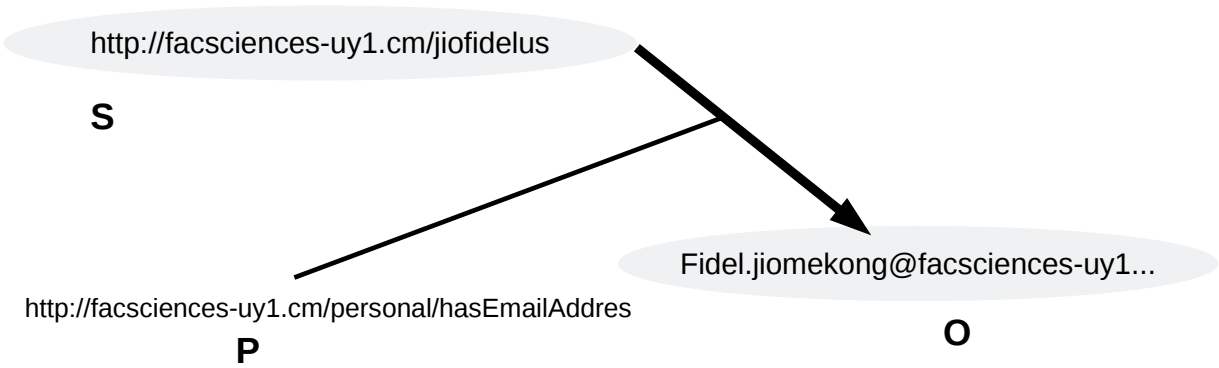


Resource Description Framework

Constituents of RDF graph : Literals

- Describe data values that do not have a seperate existence
- Strings, interpretation via datatypes

How do I represent: ''Dr. Jiomekong has the email fidel.jiomekong@facsciences-uy1.uninet.cm ?



Resource Description Framework

Constituents of RDF graph : Literals

- Typed literals can be expressed via XML Schema datatypes
- Namespace for typed literals :
  `http://www.w3c.org/2001/XMLSchema#`
  e.g. :
  "Semantics"'http ://www.w3c.org/2001/XMLSchema#string
- Language Tags denote the (natural) language of the text :
  e.g. : "Language"@en, "Language"@fr

Resource Description Framework

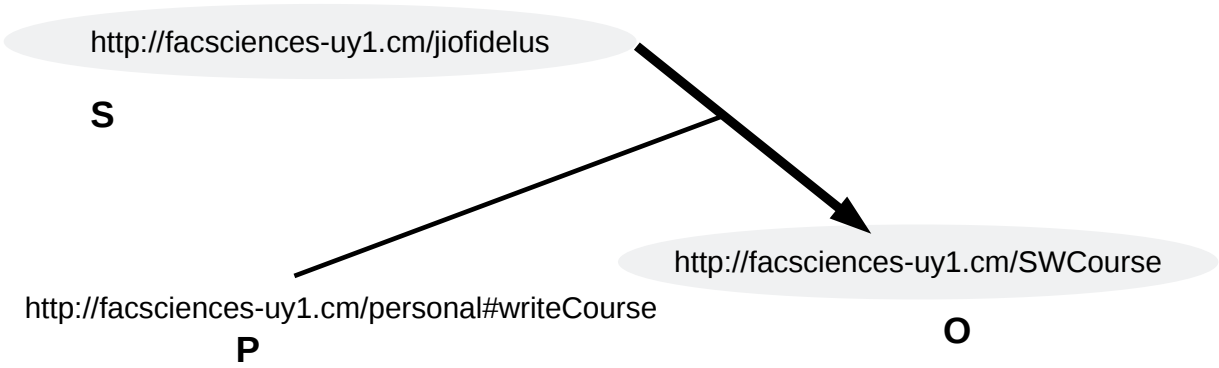
Constituents of RDF graph : Blank Nodes

Denote existence of an individual with specific attributes, but without providing an identification or reference

How do I represent: ''Existence of things of type person''

Resource Description Framework

RDF representation : Node-Edge-Node Triple



Resource Description Framework

RDF representation : N3 Notation

Simple listing of triples

```
{ http://facsciences-uy1.cm/jiofidelus ,
http://facsciences-uy1.cm/personal#writeCourse ,
http://facsciences-uy1.cm/SWCourse
}
```

Resource Description Framework

RDF representation : Turtle

Simple listing of triples

- Terse RDF triple Language
- Extension of N3
- URIs in angle brackets
- Literals in quotation marks
- Triple ends with a period
- White space will be ignored

```
<Subject> <Property> <Object>.
<Subject> <Property> "Object".
```

Resource Description Framework

RDF representation : Turtle


```
<http://facsciences-uy1.cm/Jiomekong>
<http://facsciences-uy1.cm/personal#writeCourse>
<http://facsciences-uy1.cm/SWCourse>.


<http://facsciences-uy1.cm/Jiomekong>
<http://facsciences-uy1.cm/personal#hasEmailAddress>
"fidel.jiomekong@facsciences-uy1.uninet.cm".
```

Resource Description Framework

RDF representation : Turtle


```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix ex: <http://example.org/stuff/1.0/>.

<http://www.w3.org/TR/rdf-syntax-grammar>
dc:title "RDF/XML Syntax Specification (Revised)"@en;
ex:editor[
        ex:fullname "Dave Beckett" ^^xsd:string;
        ex:homePage <http://purl.org/net/dajobe/>
].
```

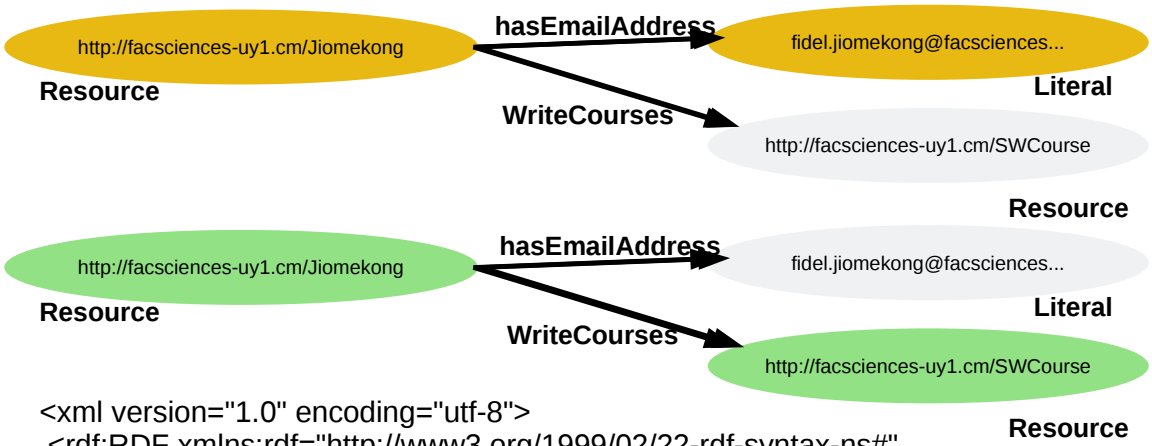
Resource Description Framework

RDF representation : RDF XML-Serialization


```
<xml version="1.0" encoding="utf-8">
<rdf:RDF xmlns:rdf="http://www3.org/1999/02/22-rdf-syntax-ns#"
                 xmlns:pers="http://facsciences-uy1.cm/personal#">

<rdf:Description rdf:about="http://facsciences-uy1.cm/Jiomekong">
        <pers:hasEmailAddress>fidel.jiomekong@...</pers:hasEmailAddress>
</rdf:Description>

<rdf:Description rdf:about="http://facsciences-uy1.cm/Jiomekong">
        <pers:writeCourse>
                <rdf:Description rdf:about="http://facsciences-uy1.cm/SWCourse">
                </rdf:Description>
        </pers:writeCourse>
</rdf:Description>
</rdf:RDF>
```

Resource Description Framework

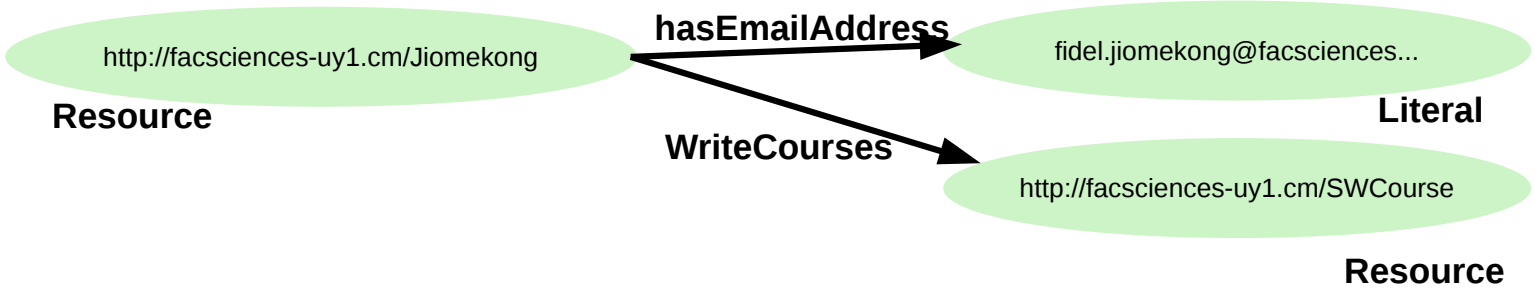
RDF representation : RDF XML-Serialization




```
<xml version="1.0" encoding="utf-8">
 <rdf:RDF xmlns:rdf="http://www3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:pers="http://facsciences-uy1.cm/personal#">

<rdf:Description rdf:about="http://facsciences-uy1.cm/Jiomekong">
      <pers:hasEmailAddress>fidel.jiomekong@...</pers:hasEmailAddress>
</rdf:Description>

<rdf:Description rdf:about="http://facsciences-uy1.cm/Jiomekong">
      <pers:writeCourse>
            <rdf:Description rdf:about="http://facsciences-uy1.cm/SWCourse">
            </rdf:Description>
      </pers:writeCourse>
</rdf:Description>
</rdf:RDF>
```

Resource Description Framework

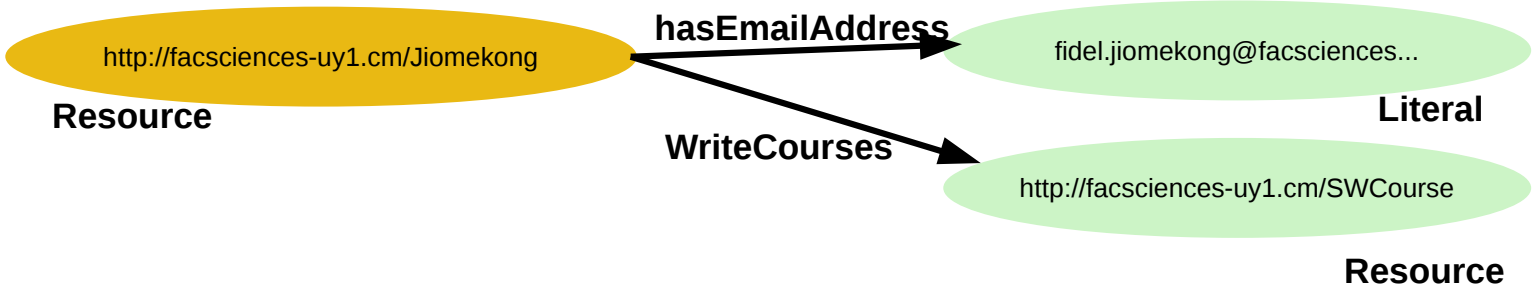
RDF representation : RDF XML-Serialization




```
<xml version="1.0" encoding="utf-8">
 <rdf:RDF xmlns:rdf="http://www3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:pers="http://facsciences-uy1.cm/personal#">

<rdf:Description rdf:about="http://facsciences-uy1.cm/Jiomekong"
      pers:hasEmailAddress= ‘’fidel.jiomekong@...’’ >
      <pers:writeCourse rdf:resource=
"http://facsciences-uy1.cm/SWCourse">
</rdf:Description>
</rdf:RDF>
```

Resource Description Framework

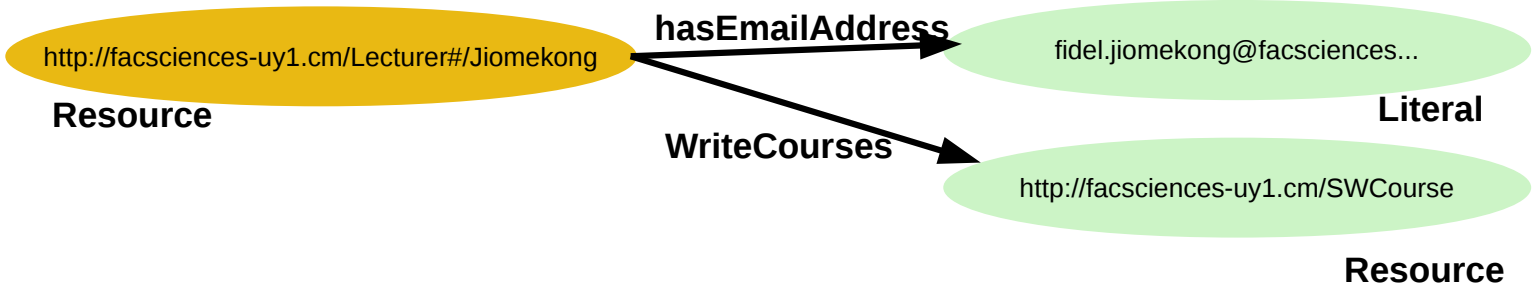
RDF representation : RDF XML-Serialization




```
<xml version="1.0" encoding="utf-8">
 <rdf:RDF xmlns:rdf="http://www3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:pers="http://facsciences-uy1.cm/personal#"
          xml:base=''http://facsciences-uy1.cm/Lecturer'' >

<rdf:Description rdf:about="http://facsciences-uy1.cm/Jiomekong"
      pers:hasEmailAddress= ''fidel.jiomekong@...'' >
      <pers:writeCourse rdf:resource=
"http://facsciences-uy1.cm/SWCourse">
</rdf:Description>
</rdf:RDF>
```

Resource Description Framework

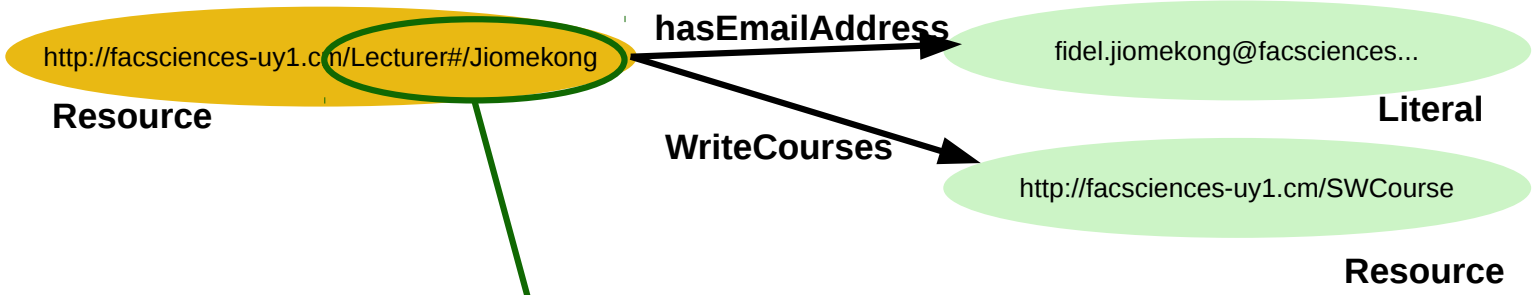
RDF representation : Turtle representation




```
@prefix rdf: <http://www3.org/1999/02/22-rdf-syntax-ns#>
@prefix pers: <http://facsciences-uy1.cm/personal#>
@base <http://facsciences-uy1.cm/Lecturer>.

:Jiomekong pers:hasEmailAddress ''fidel.jiomekong@...'' ;
      pers:writeCourse <http://facsciences-uy1.cm/SWCourse>.
```

Resource Description Framework

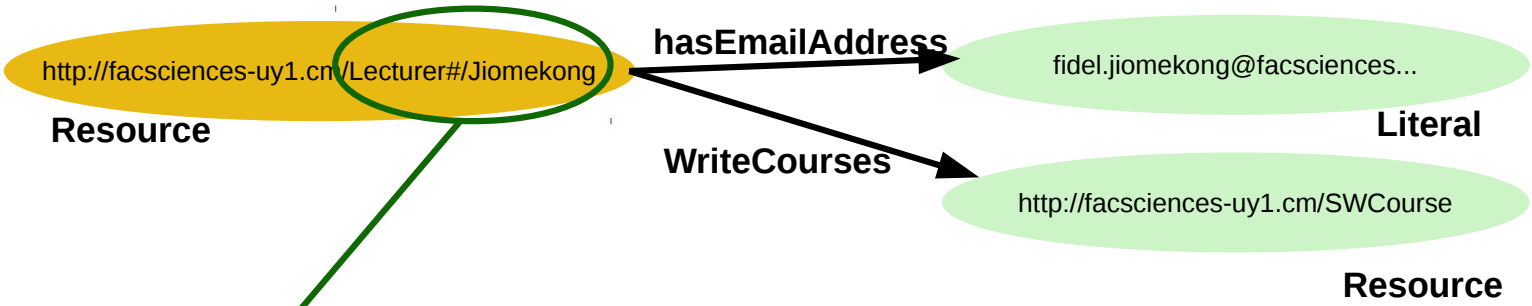
RDF representation : RDF XML-Serialization




```
<xml version="1.0" encoding="utf-8">
 <rdf:RDF xmlns:rdf="http://www3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:pers="http://facsciences-uy1.cm/personal#"
          xml:base=''http://facsciences-uy1.cm/Lecturer'' >

<rdf:Description rdf:ID="Jiomekong"
      pers:hasEmailAddress= ''fidel.jiomekong@...'' >
      <pers:writeCourse rdf:resource=
"http://facsciences-uy1.cm/SWCourse">
</rdf:Description>
</rdf:RDF>
```

Resource Description Framework

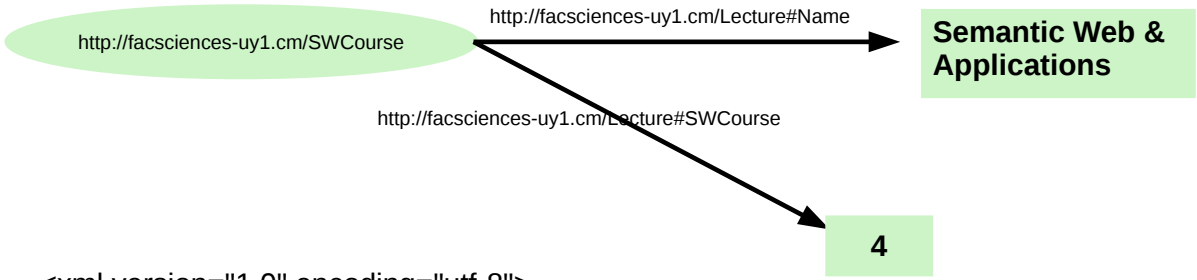
RDF representation : Turtle Serialization




```
@prefix rdf: <http://www3.org/1999/02/22-rdf-syntax-ns#>
@prefix pers: <http://facsciences-uy1.cm/personal#>
@base <http://facsciences-uy1.cm/Lecturer>.

:Jiomekong pers:hasEmailAddress ‘’fidel.jiomekong@...’’ ;
        pers:writeCourse <http://facsciences-uy1.cm/SWCourse>.
```

Resource Description Framework

RDF representation : RDF XML-Serialization

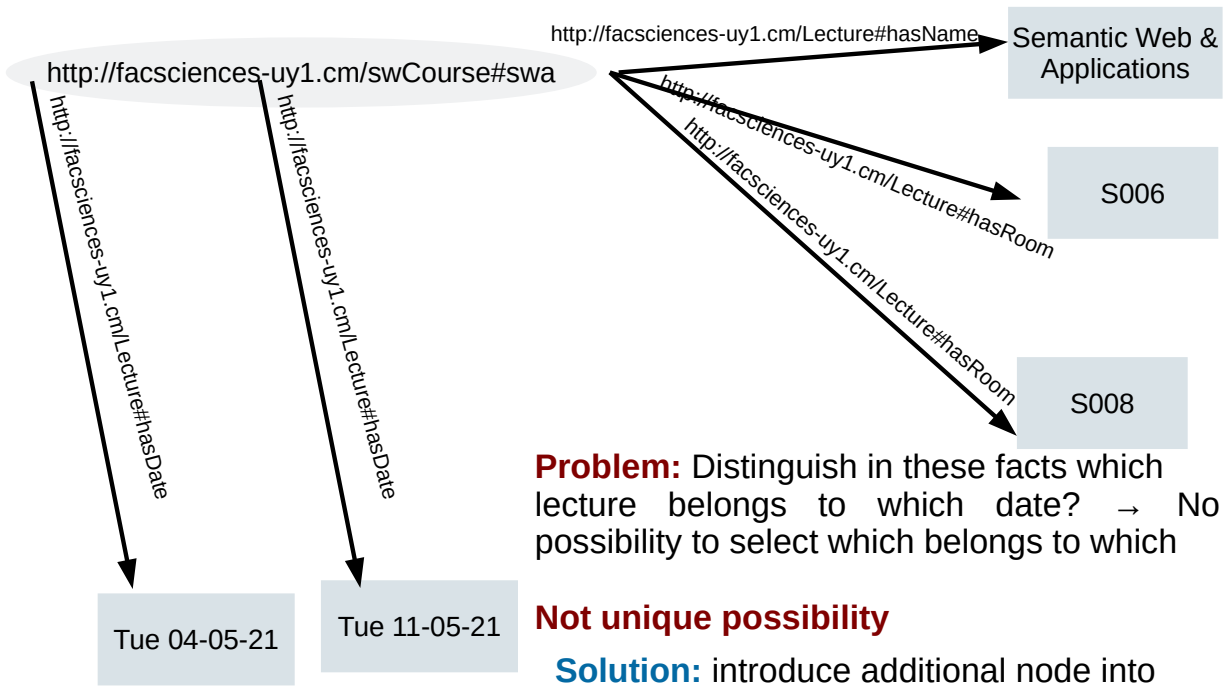



```
<xml version="1.0" encoding="utf-8">
 <rdf:RDF xmlns:rdf="http://www3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:pers="http://facsciences-uy1.cm/personal#"
          xmlns:lv=''http://facsciences-uy1.cm/Lecturer#'' >

<rdf:Description rdf:about="http://facsciences-uy1.cm/SWCourse"
     <lv:Name rdf:DataType=''http://www.w3c.org/2001/XMLSchema#string''>
          Semantic Web & Applications
     </lv:Name>

     <lv:SWCourse rdf:DataType=''http://www.w3c.org/2001/XMLSchema#Integer''>
          4
     </lv:SWCourse>
</rdf:Description>
</rdf:RDF>
```

Resource Description Framework

Blank nodes

- A lecture takes place twice a week in two different rooms
- How to model this in RDF ?



Problem: Distinguish in these facts which lecture belongs to which date? → No possibility to select which belongs to which

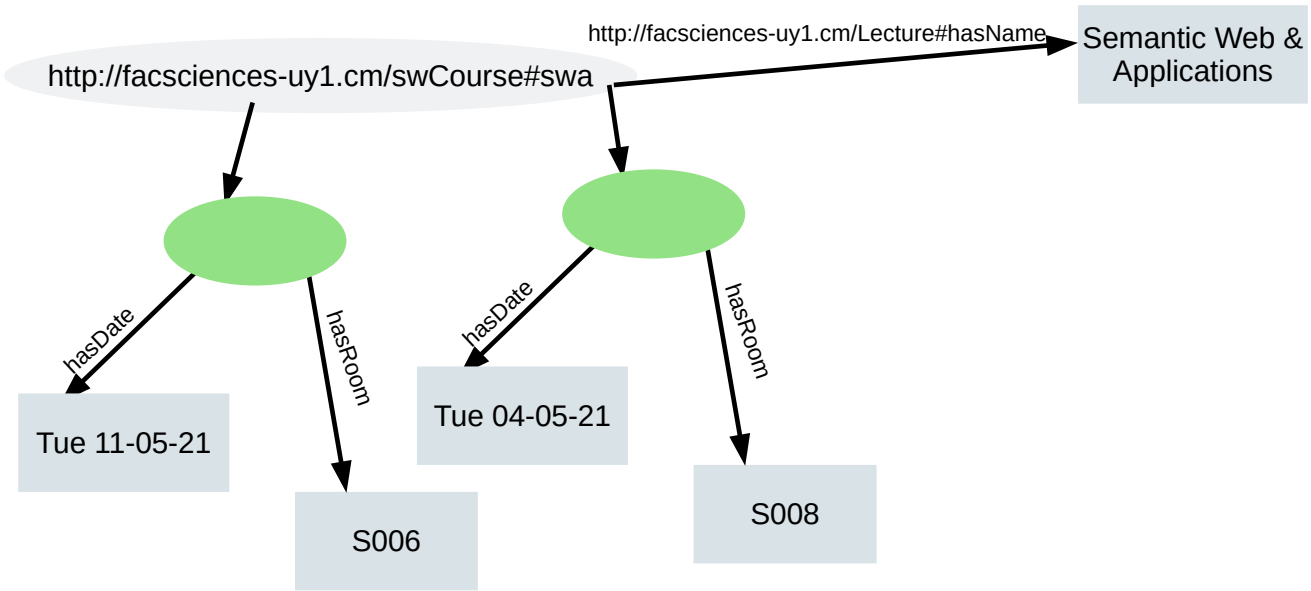
Not unique possibility

Solution: introduce additional node into the graph

Resource Description Framework

Blank nodes

- A lecture takes place twice a week in two different rooms
- How to model this in RDF ?



The blank node represent a place and a time

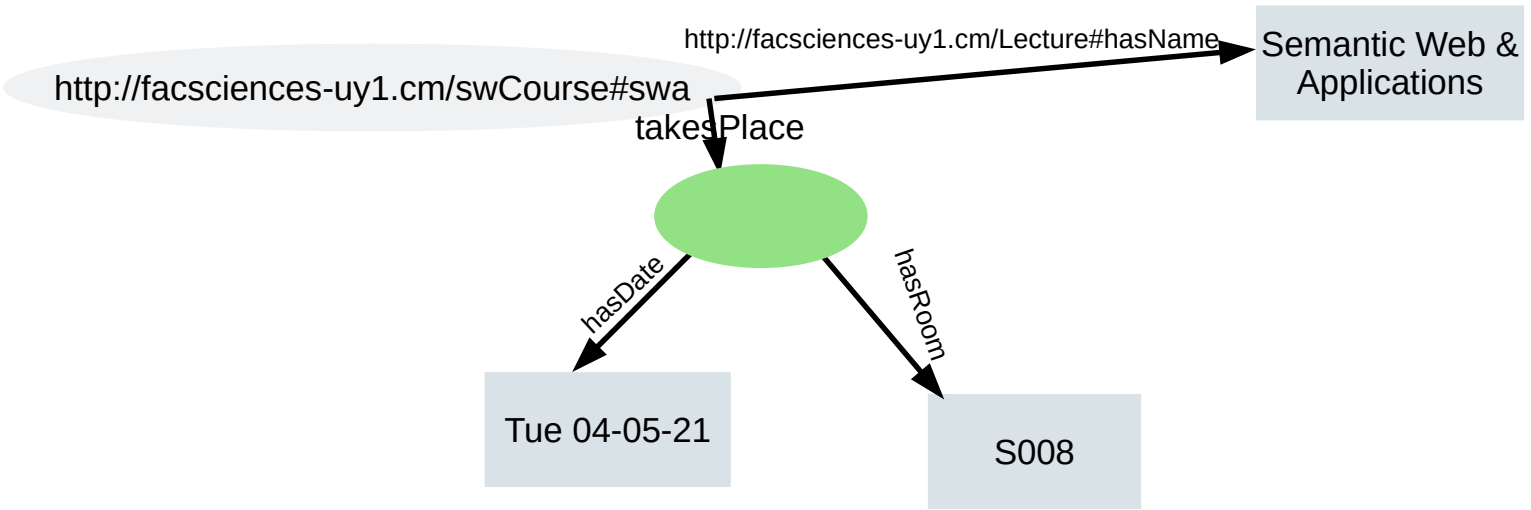
Resource Description Framework

Blank nodes

- Bnodes
- Also called Empty node
- Object which do not have URI associated with
- Used to make a statement about things that exist without telling which individual directly are addressed with this information
- Existential assertion without denoting specific individual (There exist at least one)
- Used to represent multivalue relations

Resource Description Framework

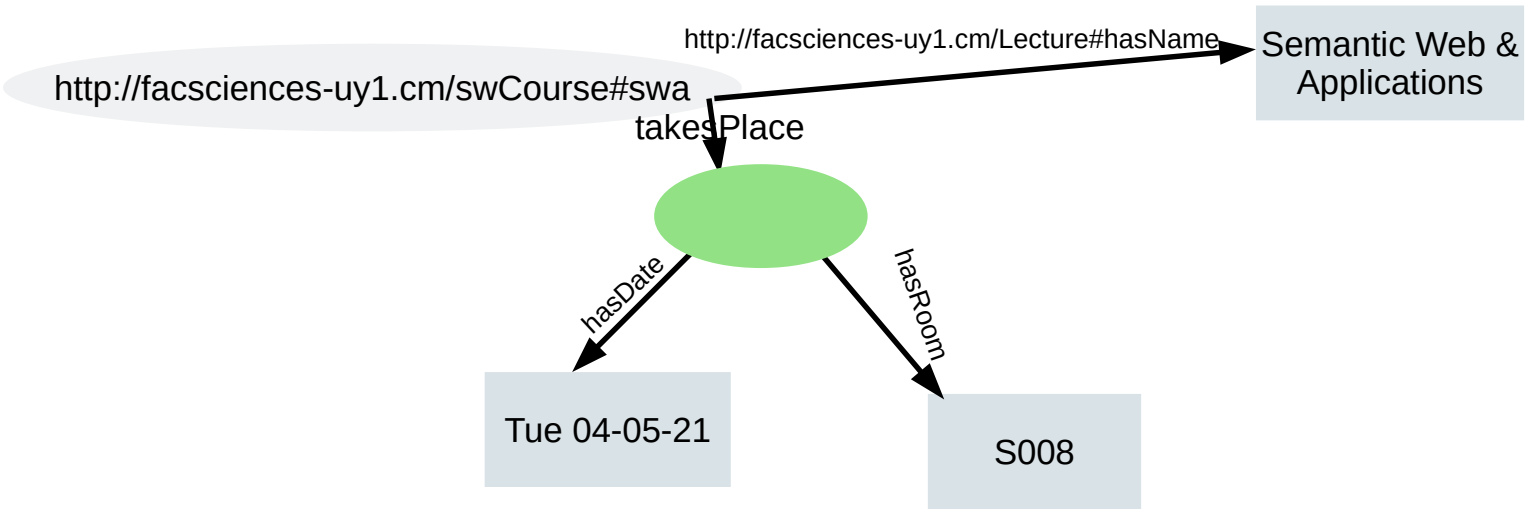
Anonymous Blank Nodes




```
<xml version="1.0" encoding="utf-8">
 <rdf:RDF xmlns:rdf="http://www3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:fs-uy1=”http://facsciences-uy1.cm/Lecturer#” >

<rdf:Description rdf:about="http://facsciences-uy1.cm/swa" >
      <fs-uy1:takesPlace= rdf:parseType=”Resource”/>
           <fs-uy1:hasDate>Tue 04-05-21</fs-uy1:hasDate>
           <fs-uy1:hasRoom>S008</fs-uy1:hasRoom>
      <fs-uy1:takesPlace>
</rdf:Description>
</rdf:RDF>
```

Resource Description Framework

Anonymous Blank Nodes : Turtle representation



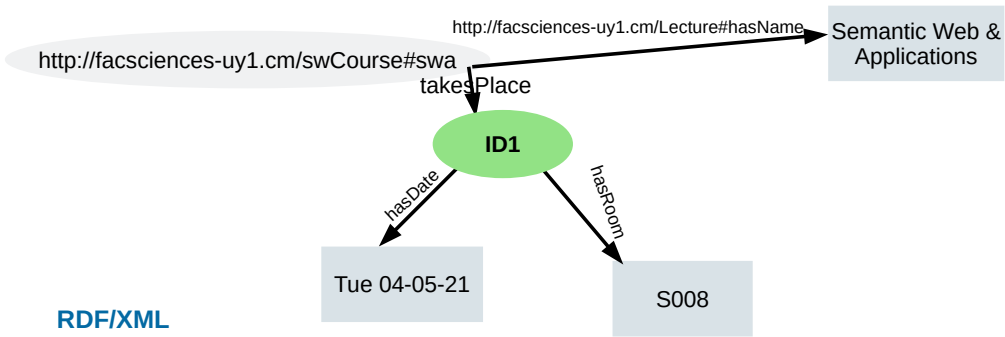
@prefix rdf: <http://www3.org/1999/02/22-rdf-syntax-ns#>
@prefix fs-uy1: <http://facsciences-uy1.cm/Lecture#>.

<http://facsciences-uy1.cm/swCourse#swa>fs-uy1:takesPlace[
fs-uy1:hasDate ‘’Tue 04-05-21’’ ;
fs-uy1:hasRoom ‘’S008’’].

Resource Description Framework

Deferencable Blank Nodes

It might be important to dereference a node $\longrightarrow$ Give this blank node a name



RDF/XML

```
<rdf:Description rdf:about="http://facsciences-uy1.cm/swCourse#swa" >
      <fs-uy1:takesPlace= rdf:nodeID="ID1"/>
</rdf:Description>

<rdf:Description rdf:nodeID="ID1" >
      <fs-uy1:takesPlace= rdf:nodeID="ID1"/>
            <fs-uy1:hasDate>Tue 04-05-21</fs-uy1:hasDate>
            <fs-uy1:hasRoom>S008</fs-uy1:hasRoom>
      <fs-uy1:takesPlace>
</rdf:Description>
</rdf:RDF>
```

Turtle

```
@prefix rdf: <http://www3.org/1999/02/22-rdf-syntax-ns#>
@prefix fs-uy1: <http://facsciences-uy1.cm/Lecture#>.

<http://facsciences-uy1.cm/swCourse#swa>fs-uy1:takesPlace _:ID1.
_:ID1 fs-uy1:hasDate ''Tue 04-05-21'' ;
      fs-uy1:hasRoom ''S008''.
```

Resource Description Framework

Lists

- Aggregation of facts
- Use to talk about single individual and aggregate into sets
- General Data structure to enumerate any resources or literals
- Only shortcuts, no additional semantic expressivity

Resource Description Framework

Lists

Two types of lists :

- **Container :**
  - open list, can be extends afterwards
  - extension (new entries) possible
  - rdf :sequential : the sequence of the list is important

- **Collection :**
  - simple linear list
  - closed list, no further extension is possible
  - rdf :nil is for close lit

Resource Description Framework

Lists : RDF Container

- The root node is assigned a container-type via **rdf :type**

- **rdf :Bag**
  - unordered set of elements
  - there is no given order of elements

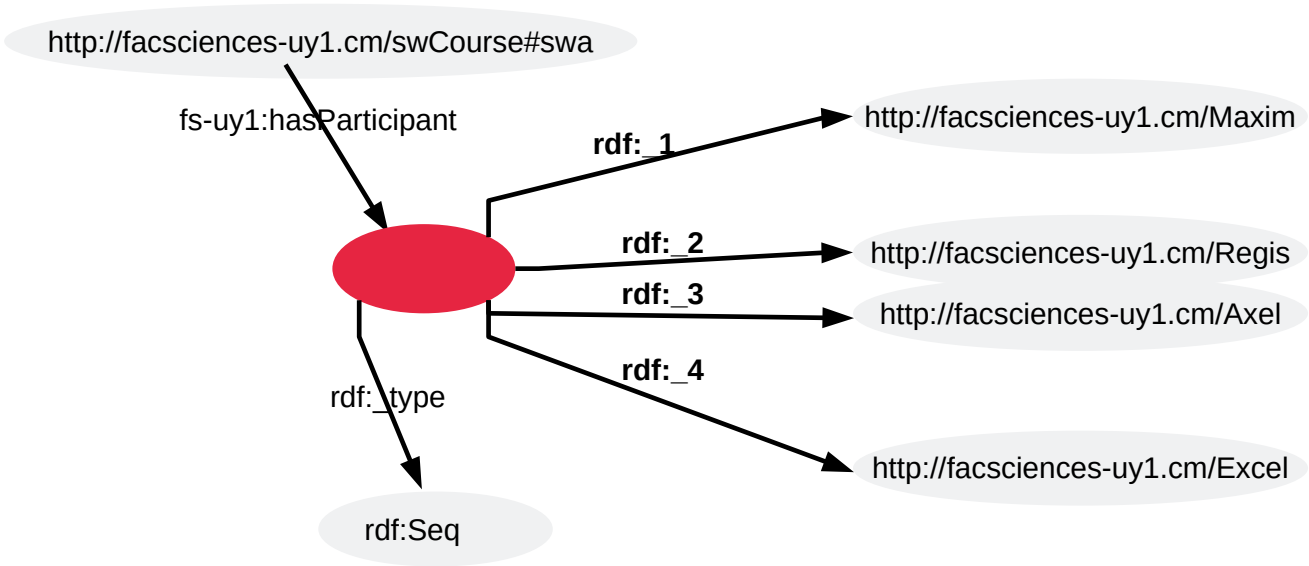
- **rdf :Seq**
  - ordered set of elements

- **rdf :Alt**
  - defines alternatives of elements
  - only one element of the given alternatives is relevant for the application

Resource Description Framework

Lists : RDF Container

RDF Container




```
@prefix rdf: <http://www3.org/1999/02/22-rdf-syntax-ns#>
@prefix fs-uy1: <http://facsciences-uy1.cm/Lecture#>.
```

Resource Description Framework

Lists : RDF Container - Turtle representation

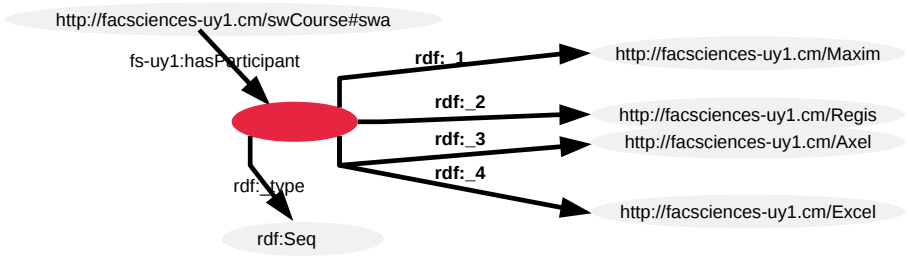
Resource Description Framework

Lists : RDF Container - RDF/XML representation

Resource Description Framework

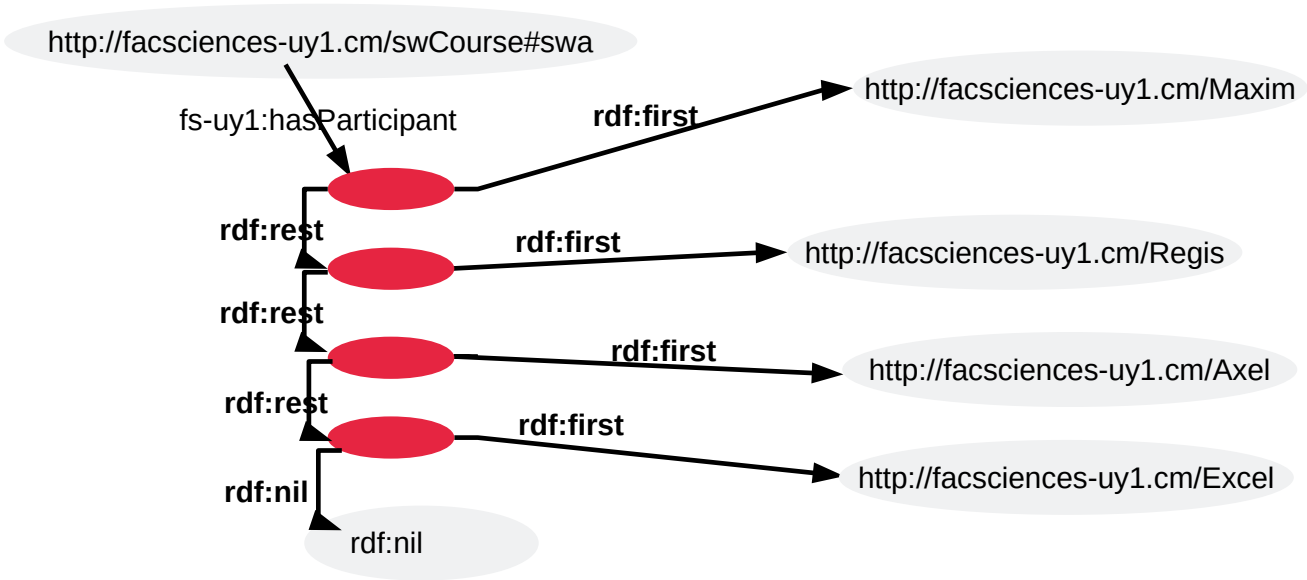
Lists : RDF Container

RDF Container




```
@prefix rdf: <http://www3.org/1999/02/22-rdf-syntax-ns#>
@prefix fs-uy1: <http://facsciences-uy1.cm/Lecture#>
@base <http://facsciences-uy1.cm/> .

:swa fs-uy1:hasParticipants[
     a rdf:Seq;
     rdf:_1 <Maxim>;
     rdf:_2 <Regis>;
     rdf:_3 <Axel>;
     rdf:_4 <Excel>.
].
```

RDF Container: rdf:collection

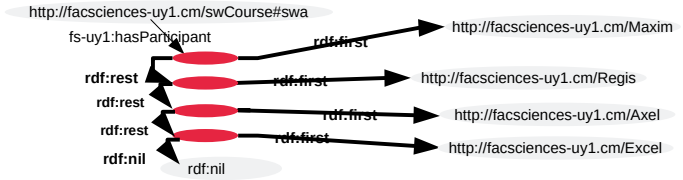



```
@prefix rdf: <http://www3.org/1999/02/22-rdf-syntax-ns#>
@prefix fs-uy1: <http://facsciences-uy1.cm/Lecture#>.
```

Resource Description Framework

Lists : RDF-Collection

RDF Container: rdf:collection




```
@prefix rdf: <http://www3.org/1999/02/22-rdf-syntax-ns#>
@prefix fs-uy1: <http://facsciences-uy1.cm/Lecture#>
@base <http://facsciences-uy1.cm/> .

:swa fs-uy1:hasParticipants[
     rdf:first <Maxim>; rdf:rest[
     rdf:first <Regis>; rdf:rest[
     rdf:first <Axel>; rdf:rest[
     rdf:first <Excel>;
     rdf:rest rdf:nil.
]]]].
```

```
@prefix rdf: <http://www3.org/1999/02/22-rdf-syntax-ns#>
@prefix fs-uy1: <http://facsciences-uy1.cm/Lecture#>
@base <http://facsciences-uy1.cm/> .

:swa fs-uy1:hasParticipants
(<Maxim> <Regis> <Axel> <Excel>).
```

Resource Description Framework

Lists : RDF Collection - Turtle representation

Resource Description Framework

Lists : RDF Collection - Turtle representation

Resource Description Framework

Reification

- Consider the following statement of Sherlock Holmes : Dear Watson, ... I suppose that the Gardner has killed the Butler" :
  - There are several facts in there
  - How to represent these fact ?
  - How to address the fact that Sherlock Holmes is referring to ?
  - $\longrightarrow$ Connected facts together to a new fact : "the Gardner has killed the butler" and "Holmes suppose that the Gardner has killed the butler"

- **Solution :** Use RDF statement/RDF Reification in which a fact may become a subject

Resource Description Framework

RDF-Reification

- Used for meta-modeling
- Used to model facts about facts
- Permits interleaving statements :
  i.e. to make statements about statements
- Example :
  "Dear Watson, ... I suppose that the Gardner has killed the Butler"
  - **Part 1 :** The Gardener has killed the Butler
    exv :Gardener exv :hasKilled exv :Butler.
  - **Part 2 :** Sherlock Holmes supposes
    exv :SherlockHolmes exv :supposes ? ? ? ?.

Resource Description Framework

RDF-Reification

- **rdf :Statement**
  Defines an RDF Statement, consisting of Subject, Predicate and Object

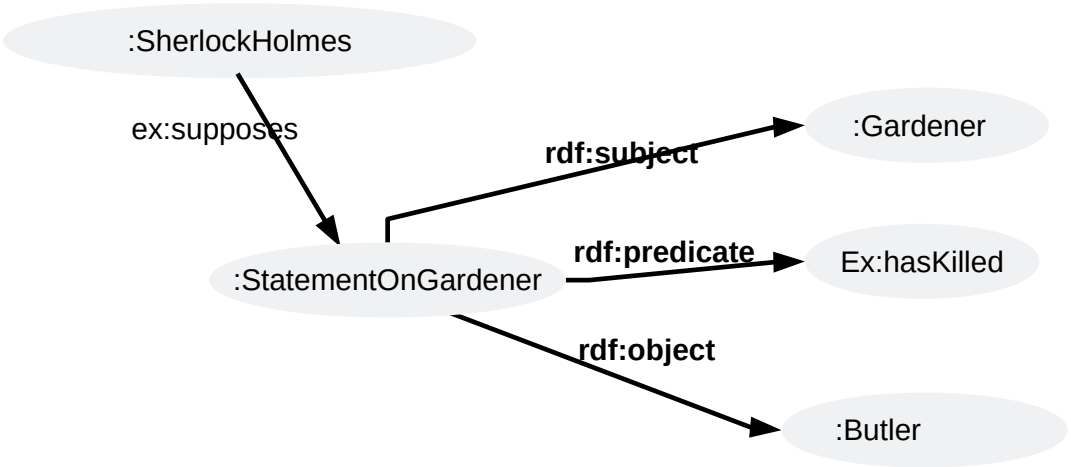
- **rdf :subject** - the described resource

- **rdf :predicate** - the original property

- **rdf :object** - the value of the property

Resource Description Framework

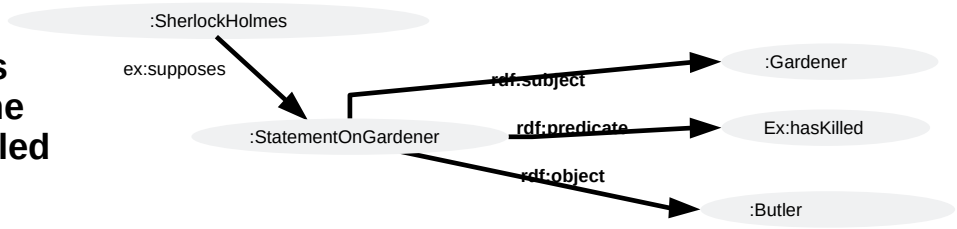
RDF-Reification



Sherlock Holmes supposes that the Gardener has killed the Butler

RDF-Reification

Sherlock Holmes supposes that the Gardener has killed the Butler




```
@prefix rdf: <http://www3.org/1999/02/22-rdf-syntax-ns#>
@ex <http://facsciences-uy1.cm/Crime#> .

:SherlockHolmes ex:supposes:StatementOnGardener .
:StatementOnGardener a rdf:Statement ;
     rdf:subject :Gardener ;
     rdf:predicate ex:hasKilled ;
     rdf:object:Butler .
```

Resource Description Framework

RDF-Reification : Turtle representation

Resource Description Framework

RDF-Reification : RDF/XML representation

Resource Description Framework

RDF-Reification

- Used to model data provenance
- Formalizing statements about **Reliability** (Trust)
- Definition of **Metadata about statements** (Assertions, Statements)
- For (general) knowledge representation (ontologies construction) Used to transform Relations into Classes

Resource Description Framework

Reification

- Used to model data provenance : Who has made a statement
- Used to formalize statements about reliability and trust

Warning : can cause the problems of decidability and type conflicts (classes, individuals, statements)

Resource Description Framework

RDF and Data Integration

Example of a Bibliography Database

Books

ID	Author	Title	Publisher	Year
ISBN 0-00-752456-Z	AJ-1254	Semantic-Aware Software	S-001	2025

Authors

ID	Name	Homepage
AJ-123	Azanzi Jiomekong	http://facsciences-uy1.cm/jiomekong

Publisher

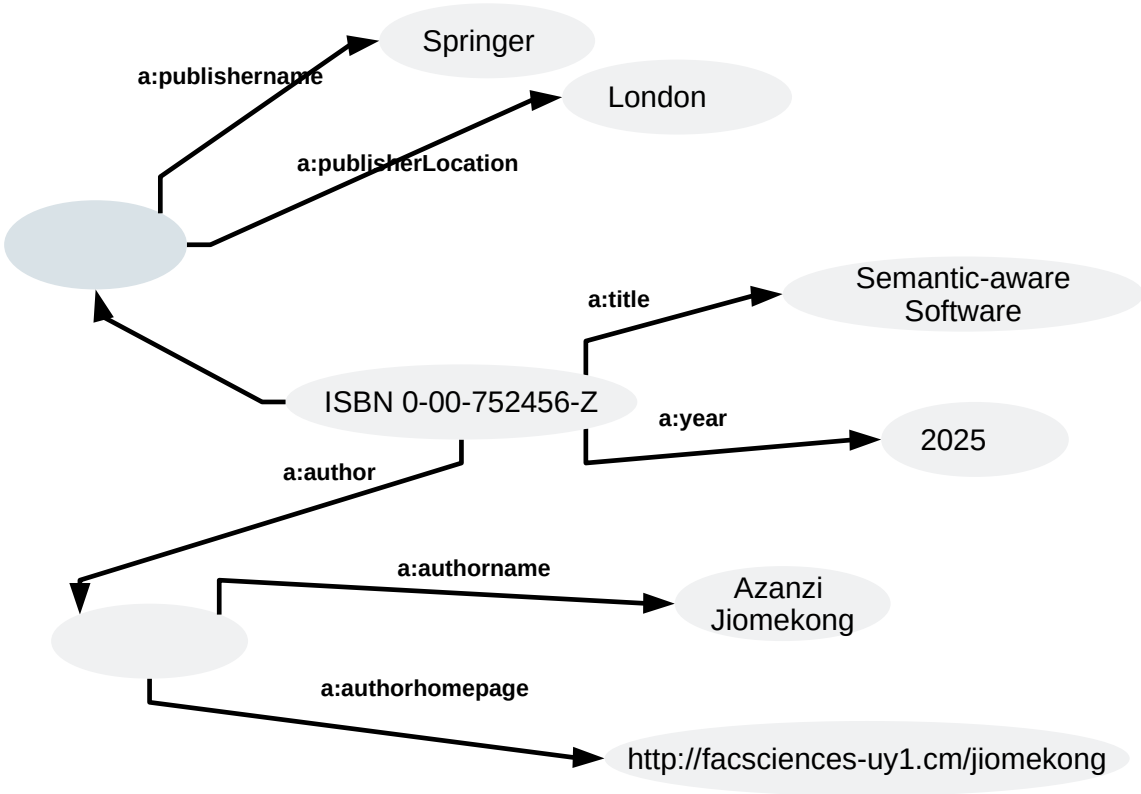
ID	Publisher	Location
S-001	Springer	London

Suppose we want to transform this database in a RDF knowledge base

Resource Description Framework

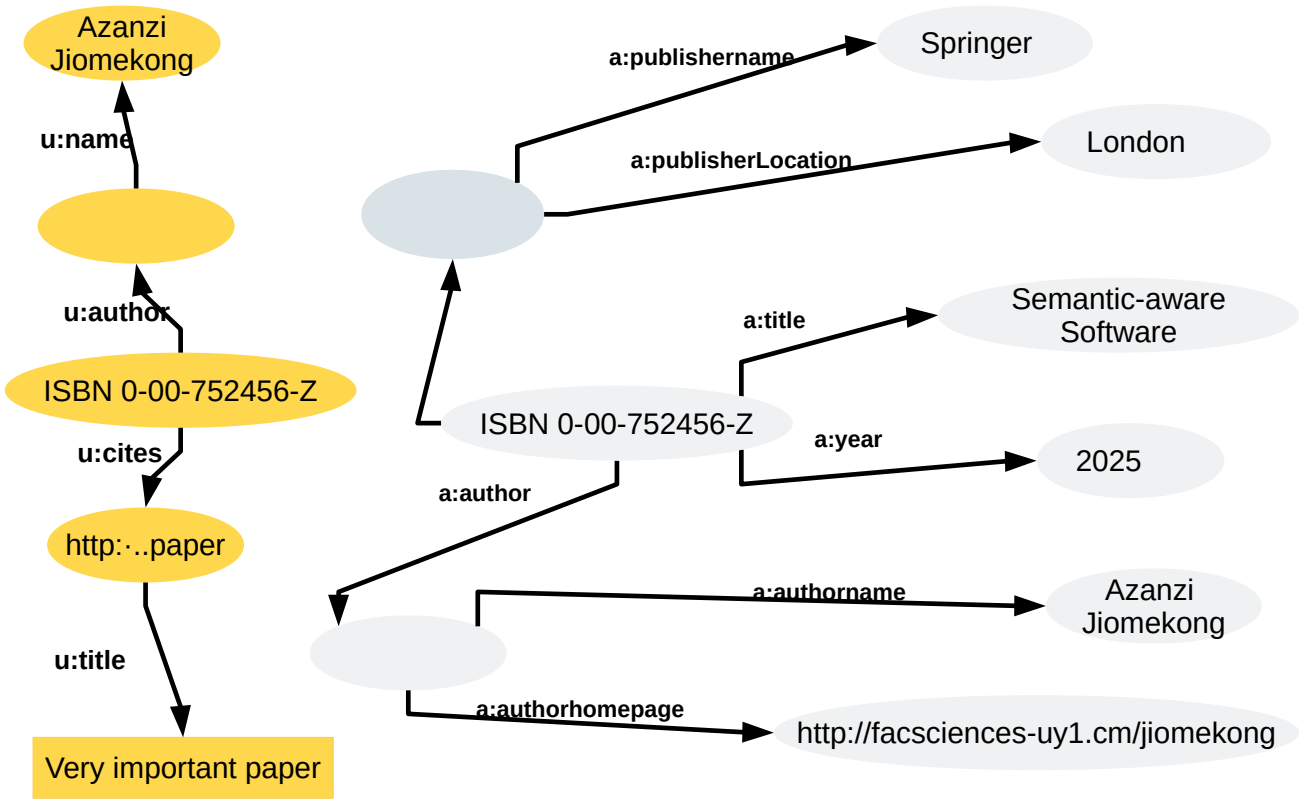
RDF and Data Integration

Database export into a set of relations



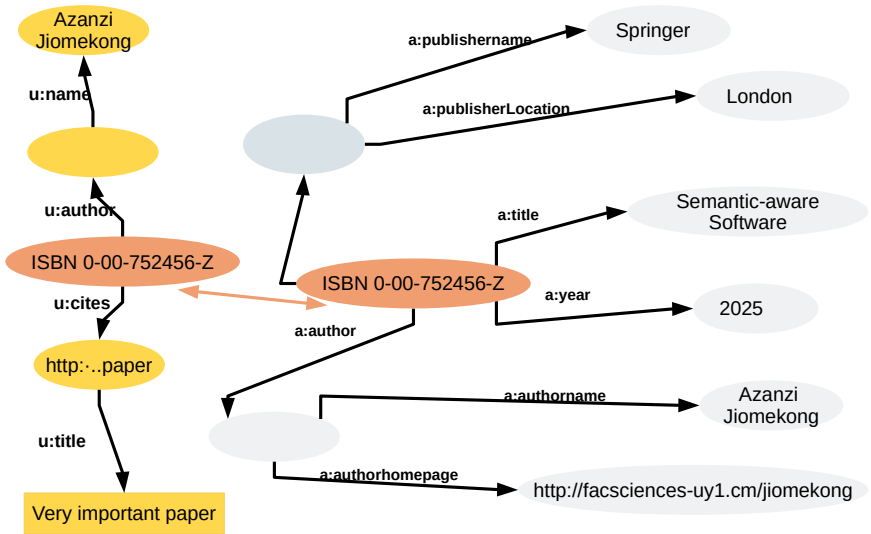
Resource Description Framework

RDF and Data Integration



Resource Description Framework

RDF and Data Integration



- Two tables with different schema
- Schema mapping is a difficult problem
- Easy to do in RDF

Summary

Element	**Description**
URI/IRI	Unique string used to identify a resource
HTTP	Protocol used to fetch resources on the Web
XML	Defines a set of rules for encoding document for data exchange
RDF	W3C standard data model for data transfert on the Web

Summary

RDF XML serialization

Element	Description
rdf :RDF	The root of an RDF document
rdf :Description	Container for the description of a resource
rdf :about	Defines the resource being described
rdf :resource	Defines a resource to identify a property
rdf :ID	Defines the ID of an element
rdf :DataType	Defines
rdf :parseType	Defines how an element should be parsed
rdf :nodeID	Defines the ID of an element node
rdf :li	Defines a list
rdf_n	Defines a node
rdf :Statement	Represent the class of RDF statement

Summary and further questions

In this course, we replied to the following questions :

- How to represent things ?
- How to encode and represent facts ?

But questions remain :

- How to represent knowledge on the Web ?
- How to store RDF and RDFS data ?

The responses to these questions are given in the next course