

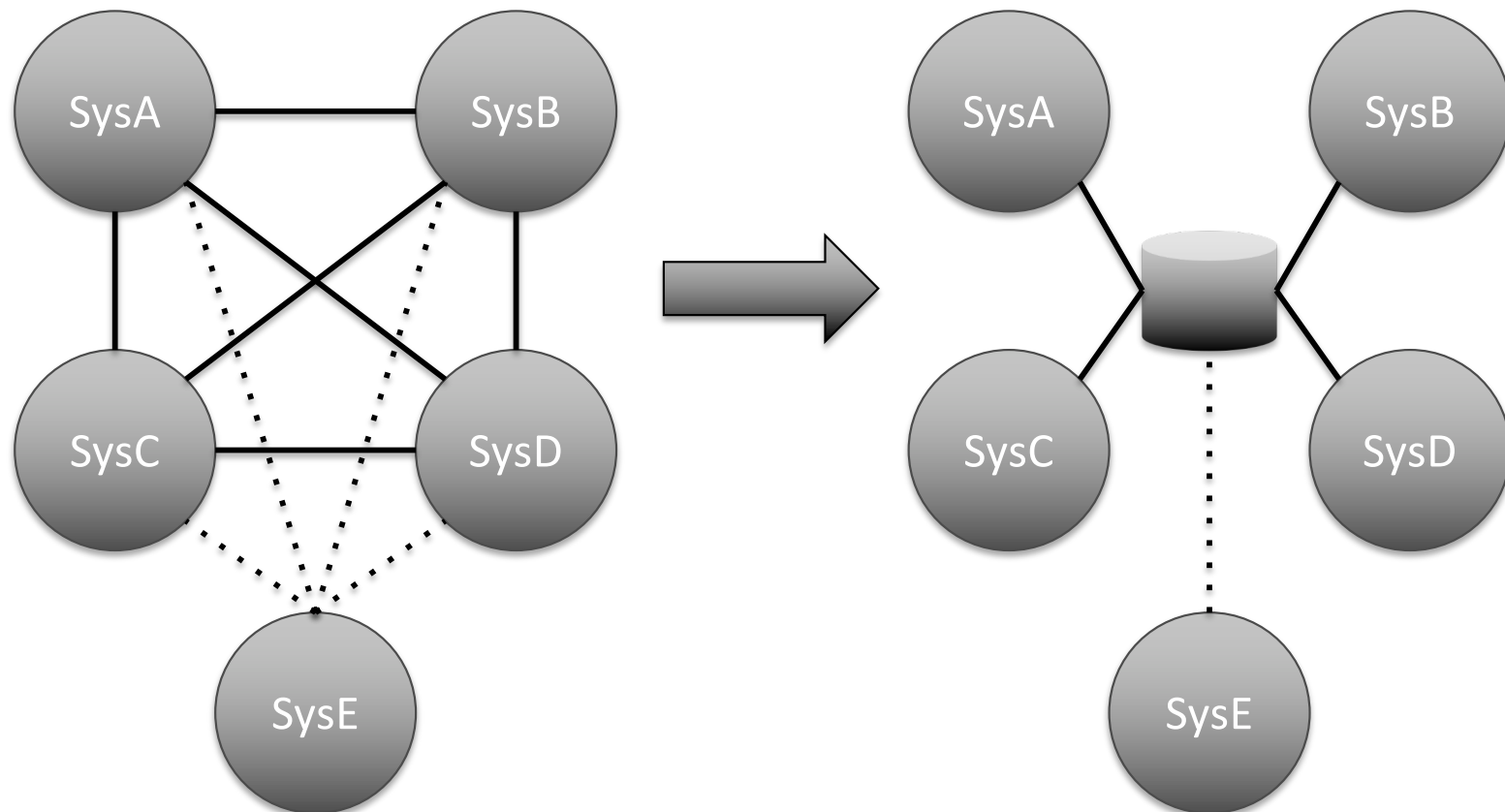
Open Information Systems 2019-2020

Lecture 2: RDF and RDF(S)

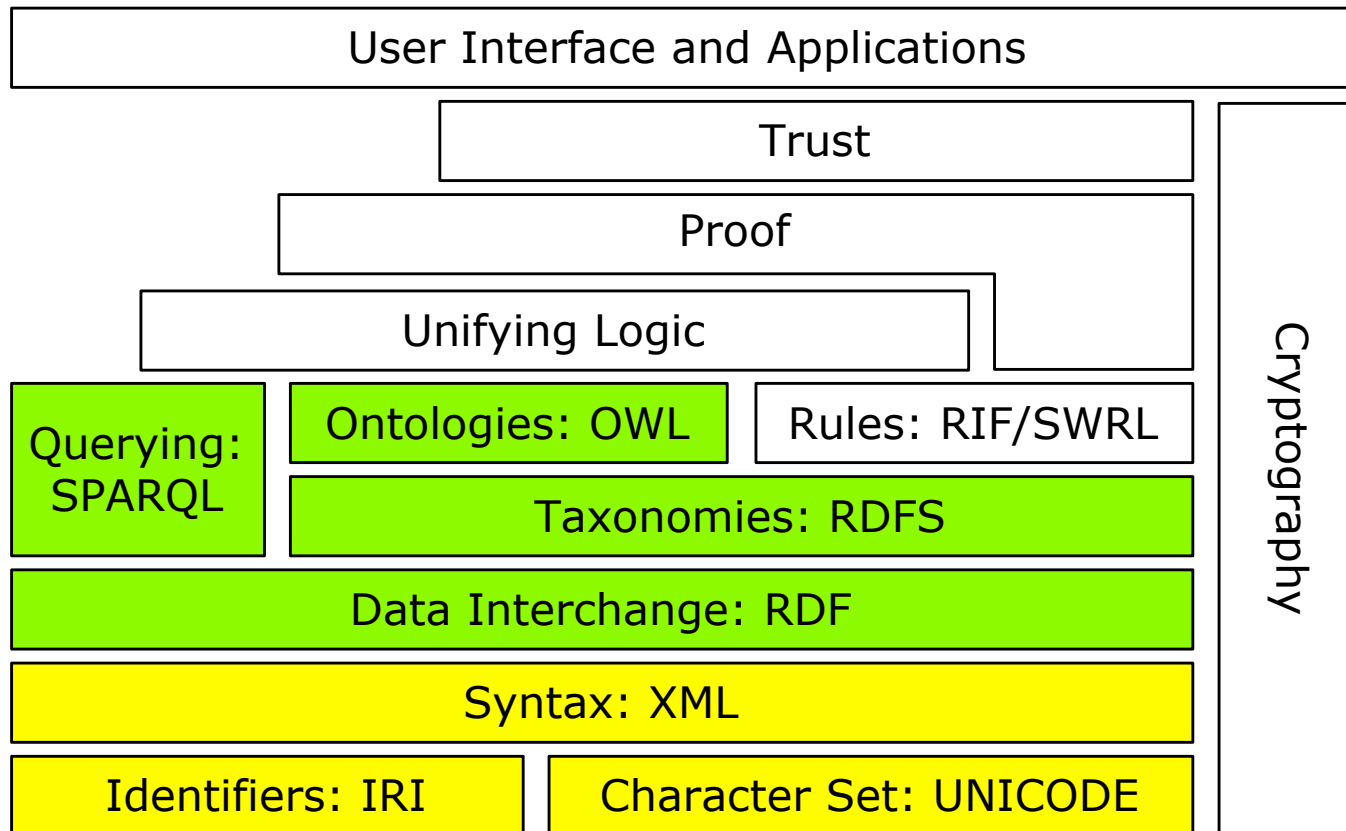
Christophe Debruyne

The problem

Traditional



The Semantic Web Stack



Different stacks exist, but the gist remains the same: a layered approach.

Resource Description Framework

We will familiarize ourselves with both the RDF/XML and TURTLE serialization of RDF. The former is useful for context and terminology, while the latter is more often adopted in practice.

Resource Description Framework

- RDF is **not an ontology language but a data model (!!!)**
- RDF is a W3C Recommendation
- RDF is designed to be **read by computers**
- RDF is for describing resources on the Web
- RDF uses URIs to identify and reference resources on the Web

RDF/XML is just one way of serializing RDF. Other serializations format include TURTLE and N3. NQuads and Trig even support (named) graphs.

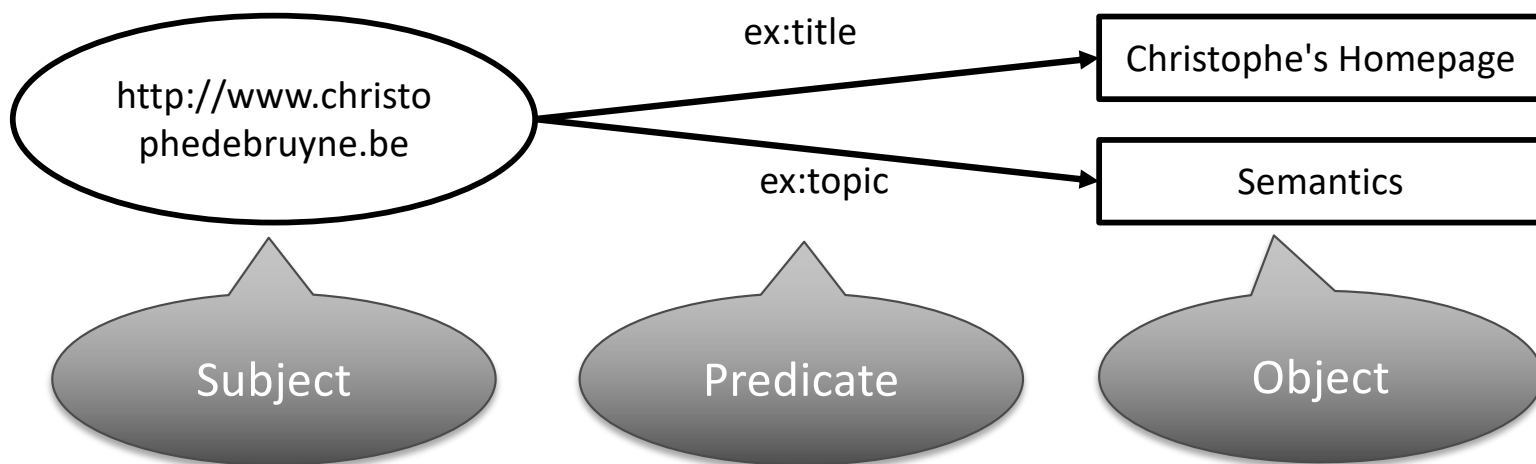
Example

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

  <rdf:Description rdf:about="http://www.christophedebruyne.be">
    <ex:title>Christophe's Homepage</ex:title>
    <ex:topic>Semantics</ex:topic>
  </rdf:Description>

</rdf:RDF>
```

RDF
Namespace



Example

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

  <rdf:Description rdf:about="http://www.christophedebruyne.be">
    <ex:title>Christophe's Homepage</ex:title>
    <ex:topic>Semantics</ex:topic>
  </rdf:Description>

</rdf:RDF>
```

RDF/XML

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

<http://www.christophedebruyne.be> ex:title "Christophe's Homepage" .
<http://www.christophedebruyne.be> ex:topic "Semantics" .
```

TURTLE

Example

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

  <rdf:Description rdf:about="http://www.christophedebruyne.be">
    <ex:title>Christophe's Homepage</ex:title>
    <ex:topic>Semantics</ex:topic>
  </rdf:Description>

</rdf:RDF>
```

RDF/XML

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

<http://www.christophedebruyne.be>
  ex:title "Christophe's Homepage" ;
  ex:topic "Semantics" .
```

TURTLE – more terse

Example

@prefix ex: <http://www.example.org/ont#> .

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

<http://www.christophedebruyne.be>
 ex:title "Christophe's Homepage" ;
 ex:topic "Semantics" .

HTTP GET http://www.example.org/ont#

@base <http://www.example.org/ont> .

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

<#title> a rdf:Property .

<#topic> a rdf:Property .

RDF Triples

	SUBJECT	PREDICATE	OBJECT
RESOURCE IDENTIFIED WITH URI	Y	Y	Y
ANONYMOUS RESOURCE (NO URI OR BLANK NODE)	Y	N	Y
LITERAL	N	N	Y

Example

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

  <rdf:Description rdf:about="http://www.christophedebruyne.be">
    <ex:title>Christophe's Homepage</ex:title>
    <ex:topic>Semantics</ex:topic>
  </rdf:Description>

</rdf:RDF>
```

RDF/XML

In RDF/XML

- Property names must be associated with a schema
- Qualify property names with a namespace prefix

RDF Description Elements

- **rdf:about** refers to an existing resource
- **rdf:ID** creates a new resource (a named node)

RDF Description Elements

- **rdf:about** refers to an existing resource
- **rdf:ID** creates a new resource (a named node)

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

  <rdf:Description rdf:ID="Christophe">
    <ex:name>Christophe Debruyne</ex:name>
  </rdf:Description>

  <rdf:Description rdf:about="http://www.christophedebruyne.be">
    <ex:author rdf:resource="#Christophe" />
  </rdf:Description>

</rdf:RDF>
```

The base URI is used to resolve relative URIs in a document.

RDF Description Elements

This notion of `rdf:about` and `rdf:ID` does not exist in TURTLE.

```
@prefix ex: <http://www.example.org/ont#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
  
<http://www.example.org/example#Christophe>  
  ex:name "Christophe Debruyne" .  
  
<http://www.christophedebruyne.be>  
  ex:author <http://www.example.org/example#Christophe> .
```

Properties have a value which can be either a literal or a resource.

Base URIs

Relative URIs are resolved relative to the current base URI. The current base URI is the URI of the document. You can “overwrite” the base URI by providing an `xml:base` directive in RDF XML, or `@base` directive in TURTLE

RDF/XML

- `xml:base="http://www.example.org/example"`
- `rdf:ID="foo"` → `http://www.example.org/example#foo`
- `rdf:about="#foo"` → `http://www.example.org/example#foo`
- `rdf:about="foo"` → `http://www.example.org/example/foo`

TURTLE

- `@base <http://www.example.org/example> .`
- `<#foo>` → `http://www.example.org/example#foo`
- `<foo>` → `http://www.example.org/example/foo`

RDF Property Elements

```
<rdf:Description rdf:about="http://www.christophedebruyne.be">  
  <ex:title>Christophe's Homepage</ex:title>  
  <ex:topic>Semantics</ex:topic>  
</rdf:Description>
```

```
<rdf:Description rdf:about="http://www.christophedebruyne.be">  
  <ex:title>Christophe's Homepage</ex:title>  
  <ex:author>  
    <rdf:Description>  
      <ex:name>Christophe Debruyne</ex:name>  
    </rdf:Description>  
  </ex:author>  
</rdf:Description>
```

← Blank Node! (See later)

Can somebody see something strange in this example?

RDF Property Elements

```
@prefix ex: <http://www.example.org/ont#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
<http://www.christophedebruyne.be>  
  ex:author [ ex:name "Christophe Debruyne" ] ;  
  ex:title  "Christophe's Homepage" ;  
  ex:topic  "Semantics" .
```

"Inline declaration of a blank node. A new blank node identifier will be generated for each [...] pair.

```
@prefix ex: <http://www.example.org/ont#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
<http://www.christophedebruyne.be>  
  ex:author _:foo ;  
  ex:title  "Christophe's Homepage" ;  
  ex:topic  "Semantics" .  
  
_:foo ex:name "Christophe Debruyne" .
```

Here, we declare our own blank node identifiers, which we can use to refer to blank nodes from various places in the graph.

Reusing nodes in RDF/XML?

```
@prefix ex: <http://www.example.org/cats#> .
```

```
ex:Victor ex:ownedBy _:foo .
```

```
ex:Bettina ex:ownedBy _:foo .
```

```
ex:Gaston ex:ownedBy _:foo .
```

```
_:foo ex:name "Christophe" .
```

```
<rdf:Description rdf:about="http://www.example.org/cats#Bettina">
```

```
  <ex:ownedBy>
```

```
    <rdf:Description rdf:nodeID="A0">
```

```
      <ex:name>Christophe</ex:name>
```

```
    </rdf:Description>
```

```
  </ex:ownedBy>
```

```
</rdf:Description>
```

```
<rdf:Description rdf:about="http://www.example.org/cats#Victor">
```

```
  <ex:ownedBy rdf:nodeID="A0"/>
```

```
</rdf:Description>
```

```
<rdf:Description rdf:about="http://www.example.org/cats#Gaston">
```

```
  <ex:ownedBy rdf:nodeID="A0"/>
```

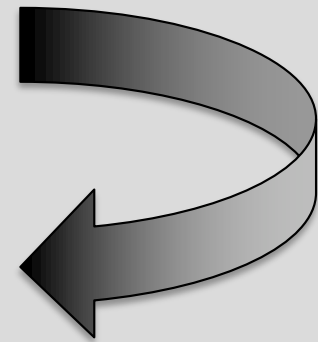
```
</rdf:Description>
```

RDF Abbreviated Syntax

- Property elements converted to attributes (only for literal values that occur once).

```
<rdf:Description rdf:about="http://www.christophedebruyne.be">  
  <ex:title>Christophe's Homepage</ex:title>  
  <ex:topic>Semantics</ex:topic>  
</rdf:Description>
```

```
<rdf:Description  
  rdf:about="http://www.christophedebruyne.be"  
  ex:title="Christophe's Homepage"  
  ex:topic="Semantics" />
```



Declaring types of things

- Types can be declared next to properties
- And things can have more than one type
- Arguably "ontology-language-ish"

```
<rdf:Description rdf:about="http://www.christophedebruyne.be">  
  <rdf:type rdf:resource="http://www.example.org/ont#Webpage"/>  
</rdf:Description>
```

```
@prefix ex: <http://www.example.org/ont#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
<http://www.christophedebruyne.be> rdf:type ex:Webpage .
```

```
@prefix ex: <http://www.example.org/ont#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
<http://www.christophedebruyne.be> a ex:Webpage .
```

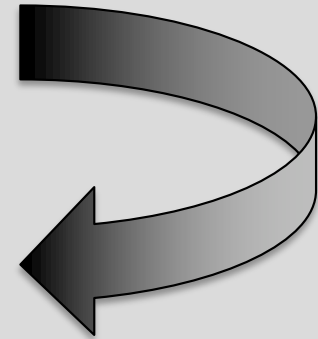
RDF Abbreviated Syntax

- Value of **rdf:type** used directly as an element name

```
<rdf:Description rdf:about="http://www.christophedebruyne.be">  
  <rdf:type rdf:resource="http://www.example.org/ont#Webpage"/>  
  <ex:title>Christophe's Homepage</ex:title>  
  <ex:topic>Semantics</ex:topic>  
</rdf:Description>
```

```
<s:Webpage rdf:about="http://www.christophedebruyne.be">  
  <ex:title>Christophe's Homepage</ex:title>  
  <ex:topic>Semantics</ex:topic>  
</s:Webpage>
```

xmlns:s="http://www.example.org/ont#"



Plain and Typed Literals

- Two types of literals: plain and typed
- Plain are strings that can have an optional language tag
- Typed are strings that have a type (usually XSD)
- Language tags have no meaning for typed literals

```
<rdf:Description rdf:ID="Christophe">
  <ex:name xml:lang="en">Christophe</ex:name>
  <ex:name>Christophe</ex:name>
  <ex:height rdf:datatype="http://www.w3.org/2001/XMLSchema#int">171</ex:height>
  <ex:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Christophe</ex:name>
</rdf:Description>
```

```
@prefix ex: <http://www.example.org/ont#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
<http://www.example.org/example#Christophe>
  ex:height "171"^^xsd:int ;
  ex:name "Christophe"^^xsd:string , "Christophe" , "Christophe"@en .
```

RDF Containers

- Are types of containers.
- Bag → An unordered group of resources or literals
- Sequence → An ordered group of resources or literals
- Alternative → A group of resources or literals that represent alternatives for the value of a property

Containers may contain duplicates.

RDF Containers

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://www.example.org/ont#"
  xmlns:cats="http://www.example.org/cats#">
  <rdf:Description rdf:about="http://www.example.org/cats#Christophe">
    <ex:adopted>
      <rdf:Bag>
        <rdf:li rdf:resource="http://www.example.org/cats#Victor"/>
        <rdf:li rdf:resource="http://www.example.org/cats#Bettina" />
        <rdf:li rdf:resource="http://www.example.org/cats#Gaston" />
      </rdf:Bag>
    </ex:adopted>
  </rdf:Description>
</rdf:RDF>
```

```
# prefixes omitted
cats:Christophe ex:adopted [ a rdf:Bag ;
                             rdf:_1 cats:Victor ;
                             rdf:_2 cats:Bettina ;
                             rdf:_3 cats:Gaston
                           ] .
```


RDF Containers

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://www.example.org/ont#"
  xmlns:cats="http://www.example.org/cats#">
  <rdf:Description rdf:about="http://www.example.org/cats#Christophe">
    <ex:adopted>
      <rdf:Seq>
        <rdf:li rdf:resource="http://www.example.org/cats#Victor"/>
        <rdf:li rdf:resource="http://www.example.org/cats#Bettina" />
        <rdf:li rdf:resource="http://www.example.org/cats#Gaston" />
      </rdf:Seq>
    </ex:adopted>
  </rdf:Description>
</rdf:RDF>
```

```
# prefixes omitted
cats:Christophe ex:adopted [ a rdf:Seq ;
                             rdf:_1 cats:Victor ;
                             rdf:_2 cats:Bettina ;
                             rdf:_3 cats:Gaston
                           ] .
```

RDF Containers

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://www.example.org/ont#"
  xmlns:cats="http://www.example.org/cats#">
  <rdf:Description rdf:about="http://www.example.org/cats#Bettina">
    <ex:listensTo>
      <rdf:Alt>
        <rdf:li>Bettina</rdf:li>
        <rdf:li>Cocotte</rdf:li>
        <rdf:li>Bouboule</rdf:li>
      </rdf:Alt>
    </ex:listensTo>
  </rdf:Description>
</rdf:RDF>
```

```
# prefixes omitted
cats:Bettina ex:listensTo [ a    rdf:Alt ;
                           rdf:_1 "Bettina" ;
                           rdf:_2 "Cocotte" ;
                           rdf:_3 "Bouboule"
                           ] .
```

RDF Collections – i.e., lists

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://www.example.org/ont#"
  xmlns:cats="http://www.example.org/cats#">
  <rdf:Description rdf:about="http://www.example.org/cats#Christophe">
    <ex:looksAfter rdf:nodeID="c1"/>
  </rdf:Description>
  <rdf:Description rdf:nodeID="c1">
    <rdf:first rdf:resource="http://www.example.org/cats#Victor"/>
    <rdf:rest rdf:nodeID="c2"/>
  </rdf:Description>
  <rdf:Description rdf:nodeID="c2">
    <rdf:first rdf:resource="http://www.example.org/cats#Bettina"/>
    <rdf:rest rdf:nodeID="c3"/>
  </rdf:Description>
  <rdf:Description rdf:nodeID="c3">
    <rdf:first rdf:resource="http://www.example.org/cats#Gaston"/>
    <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
  </rdf:Description>
</rdf:RDF>
```

Do the terms "first", "rest", and "nil" ring a bell?

-> LISP

Looks convoluted, but there is an easier way to model lists.

RDF Collections

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://www.example.org/ont#"
  xmlns:cats="http://www.example.org/cats#">
  <rdf:Description rdf:about="http://www.example.org/cats#Christophe">
    <ex:looksAfter rdf:parseType="Collection">
      <rdf:Description rdf:about="http://www.example.org/cats#Victor" />
      <rdf:Description rdf:about="http://www.example.org/cats#Bettina" />
      <rdf:Description rdf:about="http://www.example.org/cats#Gaston" />
    </ex:looksAfter>
  </rdf:Description>
</rdf:RDF>
```

In TURTLE we use parentheses for describing lists.

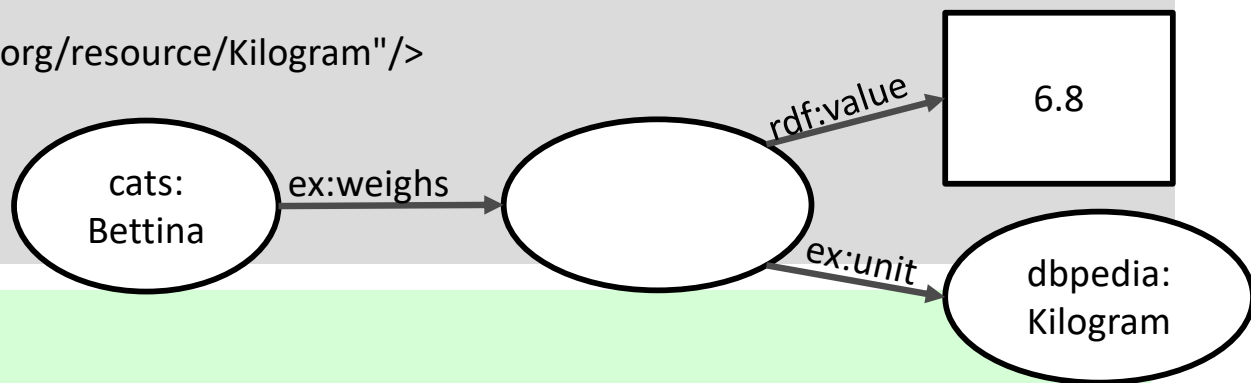
```
@prefix ex: <http://www.example.org/ont#> .
@prefix cats: <http://www.example.org/cats#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
cats:Christophe ex:looksAfter ( cats:Victor cats:Bettina cats:Gaston ).
```

Non-binary Relations

- RDF only supports binary relations

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://www.example.org/ont#"
  xmlns:cats="http://www.example.org/cats#">
  <rdf:Description rdf:about="http://www.example.org/cats#Bettina">
    <ex:weighs rdf:parseType="Resource">
      <rdf:value>6.8</rdf:value>
      <ex:unit rdf:resource="http://dbpedia.org/resource/Kilogram"/>
    </ex:weighs>
  </rdf:Description>
</rdf:RDF>
```



```
cats:Bettina ex:weighs [
  rdf:value "6.8" ;
  ex:unit  <http://dbpedia.org/resource/Kilogram>
] .
```

Non-binary Relations

- RDF only supports binary relations
- A predicate can only be applied on exactly one "value", a resource or a literal. If `rdf:parseType="Resource"` is not declared on the predicate, the parser will generate an error.
- `rdf:parseType="Resource"` instructs the parser to generate a resource and attach all values as attributes of that resource.

RDF Objects and Subjects

- Subjects of one statement can be the object of another, which creates directed labeled graphs

RDF's structure can, unlike XML, be broken down and distributed (in the same document or pointing to another). Joining of different resources using lazy loading and the URIs

@prefix ex: <http://www.example.org/ont#> .

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

@prefix dc: <http://purl.org/dc/elements/1.1/> .

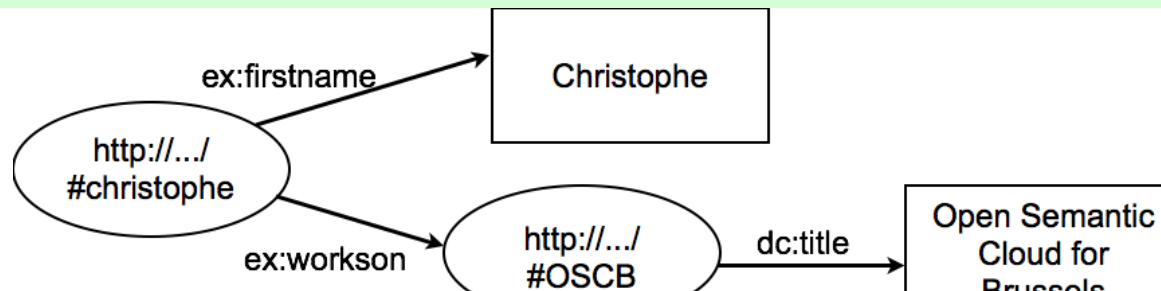
<http://www.example.org/people#christophe>

ex:firstname "christophe" ;

ex:workson <http://www.example.org/projects#OSCB> .

<http://www.example.org/projects#OSCB>

dc:title "Open Semantic Cloud for Brussels" .



Reification

- Making statements about statements

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://www.example.org/example#"
  xml:base="http://www.example.org/example">
  <rdf:Description rdf:about="http://en.wikipedia.org/wiki/Love_Me_Do">
    <ex:performedby rdf:ID="#S1"
      rdf:resource="http://en.wikipedia.org/wiki/The_Beatles"/>
  </rdf:Description>
  <rdf:Description rdf:ID="Robert">
    <ex:likes rdf:resource="#S1" />
  </rdf:Description>
</rdf:RDF>
```



Reification (II)

- Making statements about statements

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

```
ex:Robert ex:likes ex:S1 .
```

```
ex:S1    a          rdf:Statement ;  
         rdf:object  <http://en.wikipedia.org/wiki/The_Beatles> ;  
         rdf:predicate ex:performedby ;  
         rdf:subject  <http://en.wikipedia.org/wiki/Love_Me_Do> .
```



Reification

```
@prefix ex: <http://www.example.org/example#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
ex:Robert ex:likes ex:S1 .
ex:S1 a      rdf:Statement ;
      rdf:object <http://en.wikipedia.org/wiki/The_Beatles> ;
      rdf:predicate ex:performedby ;
      rdf:subject <http://en.wikipedia.org/wiki/Love_Me_Do> .
```

These two RDF documents are equivalent.

```
@prefix ex: <http://www.example.org/example#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@base <http://www.example.org/example> .
<#Robert> ex:likes <#S1> .
<#S1> a      rdf:Statement ;
      rdf:object <http://en.wikipedia.org/wiki/The_Beatles> ;
      rdf:predicate ex:performedby ;
      rdf:subject <http://en.wikipedia.org/wiki/Love_Me_Do> .
```

Conclusions

- RDF as a data model
- Two RDF serializations being RDF/XML and TURTLE
- Resources (named, and blank nodes)
- Triples: subject-predicate-object
 - Subject are resources (named or blank)
 - Predicates must have a URI
 - Objects are resources (named or blank) or literal values
- Directed labeled-graphs and descriptions can be "scattered" within and across documents
- RDF allows you to declare types, use containers and collections
- Some limitations of RDF w.r.t. reification and non-binary relationships, and how we can deal with those.

References

- XML Schema
 - <https://www.w3.org/XML/Schema>
- RDF, RDF/XML, TURTLE
 - <https://www.w3.org/TR/rdf-primer/>
 - <https://www.w3.org/TR/rdf-syntax-grammar/>
 - <https://www.w3.org/TR/turtle/>
- RDF Schema
 - <https://www.w3.org/TR/rdf-schema/>
- OWL 2.0
 - <https://www.w3.org/TR/owl2-overview/>