Semantic Web and Linked Data Graphs
or how to link data and schemas on the web

Fabien Gandon, http://fabien.info, @fabien_gandon
This proposal describes a new system at CERN. It discusses a solution based on a hierarchical system. The proposal includes "hypertext" and hypermedia. The "hypertext" includes a document by Tim Berners-Lee that refers to ACM and etc.

Keywords: HyperCard, ENQUIRE, VAX/NOTES, uucp, IBM GroupTalk, CERN, CERNDOC, DD division, MIS, OC group, RA section.
semantic web
mentioned by Tim BL
in 1994 at WWW

IDENTIFYING EVERYTHING ON THE WEB
A WEB OF LINKED DATA

User Interface & Applications

Trust

Proof

Unifying Logic

Crypto

SPARQL

Ontology: OWL

RDF-S

Rule: RIF

XML

URI/IRI

RDF

W3C®
PUBLISHED SEMANTICS OF SCHEMAS
What are the three keystones of the Web architecture?
HTTP

URI/IRI

address

communication

identification

representations
propose your own languages (XML)

- structure data using tags in a textual format
- open standard family languages
- composable languages
- open non-proprietary
XML 101

`<root>...</root>`

`<x>...</x>` or `<x/>`

`<a><b></a></b>`

`<x> ≠ <X>`

`<1an> <xmla> <bla bla>` and
`<a b='c'/>` or `<a b='c'/>`
example of a name card

<card>
  <name>gandon</name>
  <tel type="office">+33492965170</tel>
  <page url="fabien.info"/>
</card>
<card>
  <name>gandon</name>
  <tel type="office">+33492965170</tel>
  <page url="fabien.info"/>
</card>
opening and closing tags

<card>
  <name>gandon</name>
  <tel type="office">+33492965170</tel>
  <page url="fabien.info"/>
</card>
attributes

<card>
  <name>gandon</name>
  <tel type="office">+33492965170</tel>
  <page url="fabien.info"/>
</card>
self-closing tag

<card>
  <name>gandon</name>
  <tel type="office">+33492965170</tel>
  <page url="fabien.info"/>
</card>
Practice XML

<book>
  <AAAA>Architecture Now</AAAA>
  <author>Jodidio, Philip</author>
  <ID isbn10="3822840912"
  <DDDDD>
multiplication of supporting tools and standards

- Parsers (DOM, SAX) to access the content
- DTD / XML Schema to validate the structure
- XPath to select a part
- XPointer & XLink to link
- XQuery to query
- XSL to transform
- XProc to orchestrate
linked data principles

- Use RDF as data format
- Use HTTP URIs as names for things so that people can look up those names
- When someone looks up a URI, provide useful information (RDF, HTML, etc.) using content negotiation
- Include links to other URIs so that related things can be discovered
URL

identify what exists on the web.

http://my-site.fr
URL identify what exists on the web.

http://my-site.fr

URI identify, on the web, what exists.

http://animals.org/zebra#this
URL
identify what exists on the web.

http://my-site.fr

URI
identify, on the web, what exists.

http://animals.org/zebra#this

IRI
identify, on the web, in any language, what exists.

http://الحيوانات.tn/斑馬#this
Definition: a resource is anything that can be identified by a URI.

http://fabien.info/objects#mycar

e.g. a page, a person, a car, a dog, an idea, a country, a product, a service...
data.tatouille.fr
linked open data(sets) cloud on the Web

Linking Open Data cloud diagram, http://lod-cloud.net/
go there
!!!!
http://lod-cloud.net/
just the small part of LOD

http://lod-cloud.net/
### Thematic Content

<table>
<thead>
<tr>
<th>Domains</th>
<th>Number of datasets</th>
<th>Number of Triples</th>
<th>%</th>
<th>Out links</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media</td>
<td>25</td>
<td>1,841,852,061</td>
<td>5.82%</td>
<td>50,440,705</td>
<td>10.01%</td>
</tr>
<tr>
<td>Geography</td>
<td>31</td>
<td>6,145,532,484</td>
<td>19.43%</td>
<td>35,812,328</td>
<td>7.11%</td>
</tr>
<tr>
<td>Government</td>
<td>49</td>
<td>13,315,009,400</td>
<td>42.09%</td>
<td>19,343,519</td>
<td>3.84%</td>
</tr>
<tr>
<td>Publications</td>
<td>87</td>
<td>2,950,720,693</td>
<td>9.33%</td>
<td>139,925,218</td>
<td>27.76%</td>
</tr>
<tr>
<td>Inter-domain</td>
<td>41</td>
<td>4,184,635,715</td>
<td>13.23%</td>
<td>63,183,065</td>
<td>12.54%</td>
</tr>
<tr>
<td>Life Sciences</td>
<td>41</td>
<td>3,036,336,004</td>
<td>9.60%</td>
<td>191,844,090</td>
<td>38.06%</td>
</tr>
<tr>
<td>Users’ content</td>
<td>20</td>
<td>134,127,413</td>
<td>0.42%</td>
<td>3,449,143</td>
<td>0.68%</td>
</tr>
</tbody>
</table>

**Total**

|                  | 295                | 31,634,213,770    | 503,998,829 |

![Pie chart showing distribution of domains](image)
surf on the Web of data
Practice with BBC

Great White Shark

http://www.bbc.co.uk/nature/life/Great_white_shark
http://www.bbc.co.uk/nature/life/Great_white_shark.rdf

Reference: "Current and future uses of Semantic Web technologies at the BBC"
http://raimond.me.uk/slides/ismart2013/
a Web approach to data publication

« http://fr.dbpedia.org/resource/Paris »
a Web approach to data publication

« http://fr.dbpedia.org/resource/Paris »
a Web approach to data publication
a Web approach to data publication
a Web approach to data publication
a Web approach to data publication
a Web approach to data publication
linked data
content negotiation

- mechanism defined in the HTTP protocol specification
- serve different representation of a resource at the same URI
- user agents inform the servers of media types preferences (format, language, etc.)

Accept-Language: fr; q=1.0, en; q=0.5
Accept: text/html; q=1.0, text/*; q=0.8, image/gif; q=0.7, image/jpeg; q=0.6, image/*; q=0.5, */*; q=0.1

- servers select the most suited representation
a Web approach to data publication

Accept: text/html

GET

HTTP URI
a Web approach to data publication

HTTP URI

Accept: text/html

GET

303 redirect

URL of HTML
a Web approach to data publication

GET

Accept: application/rdf+xml

URL of RDF/XML

303 redirect

HTTP URI
DBpedia demo
Practical Session

1. Find “London” on DBpedia.org
   e.g. Google: "london site:dbpedia.org"
   make sure you are on the English chapter (dbpedia.org) as there are many others (fr.dbpedia.org, de.dbpedia.org)

2. Find dbp:populationDemonym and give its value

3. Find rdf:type

4. Click on value
   yago:WikicatCapitalsInEurope

5. Find “Vienna” and get its URI
   (careful: with content negotiation and redirection, the URL of the page you are currently viewing may be different from the URI of the resource it describes)

6. Native name of Vienna?
use CURL to get data
Practical Session

Do you have CURL? (windows=no, mac=yes, linux=?)

CURL: [http://curl.haxx.se/](http://curl.haxx.se/)

Installation wizard: [http://curl.haxx.se/dlwiz/?type=bin](http://curl.haxx.se/dlwiz/?type=bin)


Practical Session ++
Do you have CURL? (windows=no, mac=yes, linux=?)
CURL:  [http://curl.haxx.se/](http://curl.haxx.se/)
Installation wizard:  [http://curl.haxx.se/dlwiz/?type=bin](http://curl.haxx.se/dlwiz/?type=bin)

1. **HTML and RDF for PARIS:**

   ```
http://dbpedia.org/resource/Paris
   
http://dbpedia.org/resource/Paris
   ```

2. **HTML and RDF:** [http://ns.inria.fr/fabien.gandon#me](http://ns.inria.fr/fabien.gandon#me)

3. **HTML and RDF for ‘Vienna’ on Dbpedia**

4. **HTML and RDF for great white shark at BBC**
   [http://www.bbc.co.uk/nature/life/Great_white_shark](http://www.bbc.co.uk/nature/life/Great_white_shark)

5. **HTML and RDF for a protein**
   [http://purl.uniprot.org/uniprot/P43121](http://purl.uniprot.org/uniprot/P43121)

6. **What is the topic and format of data obtained with**
   ```
curl -o json.txt -L -H "Accept: application/json"
https://www.wikidata.org/wiki/Special:EntityData/Q551861
   ```

7. **What is the topic and format of data obtained with**
   ```
curl -o turtle.txt -L -H "Accept: text/turtle"
http://dx.doi.org/10.1007/3-540-45741-0_18
   ```
LINKED OPEN DATA
On the web
Machine-readable data
Non-proprietary format
RDF standards
Linked RDF
IS YOUR DATA 5 ☆?
This is a beta Web application and is intended as an open directory listing of vendors, government authorities and Open Government deployments worldwide. The goal of this directory is to better connect suppliers and consumers of Linked Data. This application was developed by the W3C Government Linked Data Working Group per our charter.

Regardless of W3C affiliation, we invite independent consultants, SMEs, multi-national corporations and academic research groups to enter their details. At this time a username & password is required to enter your details. Please email support at 3roundstones dot com with subject “Directory”. Thanks.

Anyone can browse the details contained in this directory of Linked Data vendors, researchers and deployments.

**Step #1:** To get started, add your organization to the directory.

**Step #2:** Click on the “Edit” tab (found on the top of your Organization’s page) to add a product, service, or project. You may enter as many products, services and projects as you wish.

**Step #3:** Click on the “View” tab (found on the top of your Organization’s page) and click on the ‘Add deployment’ link located on the right.

If you have any feedback or questions, please email team-gld-chairs at w3 dot org.

We used Callimachus, an open source framework for creating Linked Data applications. All data in this Directory is available as RDF. Download the bulk RDF.

License
Sindice - Data Web Services

Over 10 billion pieces of reusable information can already be found across 100 million web pages which embed RDF and Microformats. Start consuming this data today with Sindice Data Web services.

LEARN MORE

Search the Semantic Web

harry
harry potter

Searching on about 37.71 million documents.

Latest News

Meet us at:

Sindice - The semantic web index

Sindice Beta 1 index
The Sindice Beta 1 index is now online. Apart from the exciting geek wizardries (e.g. (More →))

An Exciting Hard Hat area
As you might have noticed by the look of the site we’re now in a very exciting transition phase. Here is a short summary of the ma...
(More →)

Sindice in use
Sindice is really meant to be used by your project, and for us it couldn’t
Free book !!!

Linked Data: Evolving the Web into a Global Data Space,
Tom Heath and Christian Bizer,

http://linkeddatabook.com/
A WEB OF LINKED DATA
A WEB OF LINKED DATA

- SPARQL
- RDF
- Ontology: OWL
- RDF-S
- Rule: RIF
- XML
- URI/IRI
RDF: data model

Web of data

RDF

HTTP

URI

communication

reference

address

User Interface & Applications

Trust

Proof

Unifying Logic

Ontology: OWL

RDF-S

Rule: RIF

RDF

XML

URI/IRI

W3C®
RDF stands for

Resource: pages, dogs, ideas... everything that can have a URI

Description: attributes, features, and relations of the resources

Framework: model, languages and syntaxes for these descriptions
RDF is a triple model i.e. every piece of knowledge is broken down into

(subject, predicate, object)
doc.html has for author Fabien and has for theme Music
doc.html has for author Fabien

doc.html has for theme Music
(doc.html, author, Fabien)
(doc.html, theme, Music)
(subject, predicate, object)
a triple
the RDF atom
Break this statement into triples?
“Fabien is a man and is French. He was born in Orléans in 1975”
RDF is also a graph model to link the descriptions of resources.
RDF triples can be seen as arcs of a graph (vertex, edge, vertex)
( doc.html, author, Fabien )
( doc.html, theme, Music )
RDF is a model for directed labeled multigraphs.

edges have a direction: starting/head node (subject) arrival/tail node (object)
RDF is a model for directed labeled multigraphs. Edges and nodes have labels.
RDF is a model for directed labeled multigraphs

- Fabien
- doc.html
- Music

Author: several edges/arcs between nodes/vertices

Contains: theme
several points of views on a graph
What is the mathematical structure built by the RDF triples?
(give the type of structure and its definition/explanation)
identify what exists on the web

http://my-site.fr

identify, on the web, what exists

http://animals.org/this-zebra
http://ns.inria.fr/fabien.gandon#me

http://inria.fr/schema#author

http://inria.fr/rr/doc.html

http://inria.fr/schema#theme

Music
open and link data in a global giant graph
in RDF values of properties can also be literals i.e. strings of characters
"Margot is a journalist woman, 32 years old, married to Arthur who is a man with whom she had two children: Marie who is a woman and Simon who is a man". For each person we also explicitly specify the name.

see the graph
namespace (XML)

- prevent name collision
- prefixes associate a tag to a URI e.g.

```xml
<ugb:score xmlns:ugb='http://www.ugb.sn/'>&gt;18&lt;/ugb:score&gt;
<bla:score xmlns:bla='http://www.ugb.sn/'>&gt;18&lt;/bla:score&gt;
```
- definitions are inherited in XML tree
- default namespace xmlns = "..."
namespace

**Definition:** abstract space gathering names in a same set

http://inria.fr/sujets#compilation
http://mit.edu/org/Lab
e.g. a dictionary, a library index, a glossary, a standard, a thesaurus,...
**NAMESPACES**
- A collection of names identified by a URI
- Names belonging to a same namespace start with the same URI.

**PREFIXES**
- Local shortcut to declare an namespace in a file, document, etc.
- Locally use prefix instead of repeating namespace.
- Representation languages provide prefix declaration means.

**QUALIFIED NAMES**
- prefix + ":" + local name
- e.g. `dc:title` instead of `<http://purl.org/dc/elements/1.1/title>`

Namespaces, Prefixes, Qualified Names (in general)
The web site prefix.cc is a useful service to find the namespaces usually associated to a prefix. Use it to find the RDF namespace.
<RDF/> has an XML syntax
<RDF /> : graphs serialized in XML trees
<RDF/> : graphs serialized in XML trees

  <rdf:Description rdf:about="http://inria.fr/rr/doc.html">
    <inria:author rdf:resource="http://ns.inria.fr/fabien.gandon#me"/>
    <inria:theme>Music</inria:theme>
  </rdf:Description>
</rdf:RDF>
<RDF/>
: one root


<rdf:Description
    rdf:about="http://inria.fr/rr/doc.html">
    <inria:author rdf:resource="http://ns.inria.fr/fabien.gandon#me"/>
    <inria:theme>Music</inria:theme>
</rdf:Description>

</rdf:RDF>
RDF : resource descriptions

  <rdf:Description rdf:about="http://inria.fr/rr/doc.html">
    <inria:author rdf:resource="http://ns.inria.fr/fabien.gandon#me"/>
    <inria:theme>Music</inria:theme>
  </rdf:Description>
</rdf:RDF>
links between resources

```xml
<RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:inria="http://inria.fr/schema#">
  <rdf:Description rdf:about="http://inria.fr/rr/doc.html">
    <inria:author rdf:resource="http://ns.inria.fr/fabien.gandon#me"/>
    <inria:theme>Music</inria:theme>
  </rdf:Description>
</RDF>
```
<RDF /> : literal values


<rdf:Description
 rdf:about="http://inria.fr/rr/doc.html">
 <inria:author rdf:resource="http://ns.inria.fr/fabien.gandon#me"/>
 <inria:theme>Music</inria:theme>
</rdf:Description>

</rdf:RDF>
RDF/XML: many syntactic variations

```xml
<rdf:RDF (…) >
  <rdf:Description rdf:about="http://inria.fr/rr/doc.html">
    <inria:author rdf:resource="http://ns.inria.fr/fabien.gandon#me"/>
  </rdf:Description>
</rdf:RDF>

<rdf:RDF (…) >
  <rdf:Description rdf:about="http://inria.fr/rr/doc.html">
    <inria:theme>Music</inria:theme>
  </rdf:Description>
</rdf:RDF>
```
RDF/XML: many syntactic variations

```xml
<rdf:RDF (…) >
  <rdf:Description rdf:about="http://inria.fr/rr/doc.html">
    <inria:theme>Music</inria:theme>
    <inria:author>
      <rdf:Description rdf:about="http://ns.inria.fr/fabien.gandon#me"/>
    </inria:author>
  </rdf:Description>
</rdf:RDF>
```
RDF/XML: many syntactic variations

<rdf:RDF (> 

<rdf:Description rdf:about="http://inria.fr/rr/doc.html"> 
  <inria:theme>Music</inria:theme> 
  <inria:author> 
    <rdf:Description>
      rdf:about="http://ns.inria.fr/fabien.gandon#me"> 
      <inria:firstName>Fabien</inria:firstName> 
    </rdf:Description> 
  </inria:author> 
</rdf:Description> 
</rdf:RDF>
Fill the blanks

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [  <!ENTITY vocabulaire "http://www.unice.fr/voc">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" ]>
  <AAA rdf:about="#Margot">
    <voc:name>Margot</voc:name>
    <voc:age rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">32</voc:age>
    <BBB rdf:resource="#Arthur"></BBB>
    <voc:hasChild rdf:resource="#Simon"></voc:hasChild>
    <voc:hasChild>
      <rdf:Description rdf:about="#Marie">
        <voc:name>Marie</voc:name>
        <rdf:type CCC="&vocabulaire;#Woman"></rdf:type>
      </rdf:Description>
    </voc:hasChild>
    <DDD rdf:resource="&vocabulaire;#Journalist"></DDD>
  </AAA>  (…)
```
Fill the blanks

(...)

<EEEE rdf:about="#Arthur">
  <voc:name>Arthur</voc:name>
  <voc:hasChild rdf:resource="#Simon"></voc:hasChild>
  <voc:hasChild rdf:resource="#Marie"></voc:hasChild>
</EEEE>

<voc:Man rdf:about="#Simon">
  <voc:name>Simon</voc:name>
</voc:Man>
</rdf:RDF>
RDF has other syntaxes
(Turtle, TriG, N-Triples, N-Quads, JSON, RDFa)
RDF has a minimalist syntax (N-Triples)
just a list of triples: simple to load / parse
- URI between angle brackets  
- literal values between double quotes  
- triplets separated by a point  

(but verbose)

N-Triples
RDF popular and concise syntax (Turtle/N3)
RDF very concise syntax (Turtle/N3)

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

<http://inria.fr/rr/doc.html>
  inria:author <http://ns.inria.fr/fabien.gandon#me> ;
  inria:theme "Music" .
RDF prefix declaration

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

<http://inria.fr/rr/doc.html>
  inria:author <http://ns.inria.fr/fabien.gandon#me> ;
  inria:theme "Music" .
RDF URI between angle brackets or qualified names

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

<http://inria.fr/rr/doc.html>
  inria:author <http://ns.inria.fr/fabien.gandon#me> ;
  inria:theme "Music" .
RDF one (.) or many properties (;) or values (,)

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

<http://inria.fr/rr/doc.html>
  inria:author <http://ns.inria.fr/fabien.gandon#me> ;
Fill the blanks

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix voc: <http://www.unice.fr/voc#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://www.unice.fr/data#Margot> a voc:Journalist AAA BBB ;
  voc:age "32"^^xsd:string ;
  voc:hasChild <http://www.unice.fr/data#Marie>, <CCC> ;
  voc:hasSpouse <http://www.unice.fr/data#Arthur> ;
  voc:name "Margot" .

<http://www.unice.fr/data#Arthur> DDD voc:Man ;
  voc:hasChild <http://www.unice.fr/data#Marie>,
    <http://www.unice.fr/data#Simon> ;
  voc:name "Arthur" .

<http://www.unice.fr/data#Marie> a voc:Woman ;
  voc:name "Marie" .

<EEE> a FFF ;
  GGG HHH .
What is the historical syntax of RDF?
validate & transform

Tim Berners-Lee

Biography

A graduate of Oxford University, Tim Berners-Lee invented the World Wide Web, an Internet-based hypermedia initiative for global information sharing while at CERN, the European Particle Physics Laboratory, in 1989. He wrote the first web client and server in 1990. His specifications of URIs, HTTP and HTML were refined as Web technology spread.

He is the 3Com Founders Professor of Engineering in the School of Engineering with a joint appointment in the Department of Electrical Engineering and Computer Science at the Laboratory for Computer Science and Artificial Intelligence (CSAIL) at the Massachusetts Institute of Technology (MIT) where he also heads the Decentralized Information Group (Digi). He is also a Professor in the Electronics and Computer Science Department at the University of Southampton, UK.

He is the Director of the World Wide Web Consortium (W3C), a Web standards organisation founded in 1994 which develops interoperable technologies (specifications, guidelines, software, and tools) to lead the Web to its full potential. He was a Director of the Web Science Trust (WST) launched in 2009 to promote research and education in Web Science, the multidisciplinary study of humanity connected by technology.

Tim is a Director of the World Wide Web Foundation, launched in 2009 to coordinate efforts to further the potential of the Web to benefit humanity.

He has promoted open government data globally, is a member of the UK's Transparency Board, and president of London's Open Data Institute.

In 2001 he became a Fellow of the Royal Society. He has been the recipient of several international awards including the Japan Prize, the Prince of Asturias Foundation Prize, the Millennium Technology Prize and Germany's Die Quadriga award. In 2004 he was knighted by H.M. Queen Elizabeth and in 2007 he was awarded the Order of Merit. In 2009 he was elected a foreign associate of the National Academy of Sciences. He is the author of 'Weaving the Web'.

On March 18, 2013, Tim, along with Vinton Cerf, Robert Kahn, Louis Pouzin and Marc Andreessen, was awarded the Queen Elizabeth Prize for Engineering for "ground-breaking innovation in engineering that has been of global benefit to humanity."
Practice

1. Get the RDF data from:
   http://ns.inria.fr/fabien.gandon#me
2. What is the syntax used?
3. Validate it and see the graph:
   http://www.w3.org/RDF/Validator/
4. Translate into Turtle/N3:
   http://rdf-translator.appspot.com/
5. Visualize it also with:
   https://graves.cl/visualRDF/
6. Adapt to your data and do it again
writing rules for RDF triples

- the subject is always a resource (never a *literal*)
- properties are binary relations and their types are identified by IRIs
- the value is a resource or a literal
XML schema datatypes & literals
standard literals are xsd:string

type literals with datatypes from XML Schema

```
<rdf:Description rdf:about="#Fabien">
  <teaching rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">true</teaching>
  <birth rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1975-07-31</birth>
</rdf:Description/>

#Fabien teaching "true"^^xsd:boolean ;
birth "1975-07-31"^^xsd:date .
```
XML Schema datatypes
blank nodes (bnodes)
handy anonymous nodes (existential quantification)
there exist a resource such that... \{ \exists r ; \ldots \}

```xml
<rdf:Description rdf:about="http://bu.ch/123.html ">
  <author>
    <rdf:Description>
      <surname>Doe</surname>
      <firstname>John</firstname>
    </rdf:Description>
  </author>
  <title>My Life</title>
</rdf:Description>
```
<rdf:Description rdf:about="#car91">
  <ex:weight rdf:parseType="Resource">
    <rdf:value rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal">1.5</rdf:value>
    <ex:unit rdf:resource="http://unit.org/ton"/>
  </ex:weight>
</rdf:Description>

#car91 ex:weight [ ex:unit <http://unit.org/ton> ;
  rdf:value 1.5 ] .
<Book>
<title xml:lang='fr'>Seigneur des anneaux</title>
<title xml:lang='en'>Lord of the rings</title>
</Book>

literals with languages and without are disjoint

"Fabien" ≠ "Fabien"@en ≠ "Fabien"@fr
typing resources

using URIs to identify the types

```xml
<urn://~fgandon> rdf:type <http://www.inria.fr/schema#Person>
```

a resource can have several types

```xml
<urn://~fgandon> rdf:type <http://www.inria.fr/schema#Person>
<urn://~fgandon> rdf:type <http://www.inria.fr/schema#Researcher>
<urn://~fgandon> rdf:type <http://www.mit.edu/schema#Lecturer>
```

```xml
<rdf:Description rdf:about="urn://~fgandon">
  <rdf:type rdf:resource="http://www.inria.fr/schema#Person" />
  <name>Fabien</name>
</rdf:Description>
```

```xml
<in:Person rdf:about="urn://~fgandon">
  <name>Fabien</name>
</in:Person>
```
question:

<?xml version="1.0"?>
  <rdf:Description rdf:about="http://example.org/doc.html">
    <rdf:type rdf:resource="http://example.org/schema#Report"/>
    <exs:theme rdf:resource="http://example.org#Music"/>
    <exs:theme rdf:resource="http://example.org#History"/>
    <exs:nbPages rdf:datatype="http://www.w3.org/2001/XMLSchema#int">23</exs:nbPages>
  </rdf:Description>
</rdf:RDF>

means...?
question:

<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:exs="http://example.org/schema#">
  <rdf:Description rdf:about="http://example.org/doc.html">
    <rdf:type rdf:resource="http://example.org/schema#Report"/>
    <exs:theme rdf:resource="http://example.org#Music"/>
    <exs:theme rdf:resource="http://example.org#History"/>
    <exs:nbPages rdf:datatype="http://www.w3.org/2001/XMLSchema#int">23</exs:nbPages>
  </rdf:Description>
</rdf:RDF>
Visit Victor Hugo

1. See HTML data from:
   http://id.loc.gov/authorities/names/n79091479.html

2. Get RDF data from:
   http://id.loc.gov/authorities/names/n79091479.rdf

3. What is the syntax?

4. Translate into Turtle/N3:
   http://rdf-translator.appspot.com/

5. Any remark?
bags = unordered groups

```xml
<rdf:Description rdf:about="#">
  <author>
    <rdf:Bag>
      <rdf:li>Ivan Herman</rdf:li>
      <rdf:li>Fabien Gandon</rdf:li>
    </rdf:Bag>
  </author>
</rdf:Description>
```

```sparql
<#> author [ a rdf:Bag ; rdf:li "Ivan Herman" ; rdf:li "Fabien Gandon" . ] .

<#> author _:a
  _:a rdf:_1 "Ivan Herman"
  _:a rdf:_2 "Fabien Gandon"
```
sequence
ordered group of resources or literals

<rdf:Description rdf:about="#partition">
  <contains>
    <rdf:Seq>
      <rdf:li rdf:about="#C"/>
      <rdf:li rdf:about="#C"/>
      <rdf:li rdf:about="#C"/>
      <rdf:li rdf:about="#D"/>
      <rdf:li rdf:about="#E"/>
    </rdf:Seq>
  </contains>
</rdf:Description>
alternatives
e.g. title of a book in different languages

```xml
<rdf:Description rdf:about="#book">
  <title>
    <rdf:Alt>
      <rdf:li xml:lang="fr">l’homme qui prenait sa femme pour un chapeau</rdf:li>
      <rdf:li xml:lang="en">the man who mistook his wife for a hat</rdf:li>
    </rdf:Alt>
  </title>
</rdf:Description>
```

```
<#book>
  title [a rdf:Alt ;
            rdf:li "l’homme..."@fr ;
            rdf:li "the man..."@en .]
</#book>
```
<rdf:Description rdf:about="#week">
  <dividedIn rdf:parseType="Collection">
    <rdf:Description rdf:about="#monday"/>
    <rdf:Description rdf:about="#tuesday"/>
    <rdf:Description rdf:about="#wednesday"/>
    <rdf:Description rdf:about="#thursday"/>
    <rdf:Description rdf:about="#friday"/>
    <rdf:Description rdf:about="#saturday"/>
    <rdf:Description rdf:about="#sunday"/>
  </dividedIn>
</rdf:Description>
The 3 winners of a race

To represent the fact a race had exactly three winners of a race sorted according to the arrival order, you preferably use:

1. alternatives (rdf:Alt)
2. a sequence (rdf:Seq)
3. a collection (rdf:List)
4. a group/set (rdf:Bag)
reification of a statement (~deprecated)

– a triple is reified by a statement
– the statement turns the triple into a resource
– the statement resource can be described

```
<rdf:Statement rdf:nodeID="decFab">
  <rdf:subject rdf:resource="http://inria.fr/doc.html" />
  <rdf:predicate rdf:resource="&dc;creator"/>
  <rdf:object rdf:resource="urn://~fgandon"/>
</rdf:Statement>

<rdf:Description rdf:nodeID="decFab">
  <dc:creator rdf:resource="http://inria.fr/"/>
</rdf:Description>
```
RDF (named) graphs group triples in graphs named by IRIs
RDF 1.1 extends Turtle and N-Triples for named graphs
@prefix rdf:  
  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix inria: <http://inria.fr/schema#> .

GRAPH <http://inria.fr/people> {
  <http://inria.fr/rr/doc.html> inria:author  
    <http://ns.inria.fr/fabien.gandon#me> .
}

GRAPH <http://inria.fr/topics> {
}
<http://inria.fr/rr/doc.html>

<http://inria.fr/schema#author>
<http://ns.inria.fr/fabien.gandon#me>
<http://inria.fr/people> .

<http://inria.fr/rr/doc.html>

<http://inria.fr/schema#theme> "Music"
<http://inria.fr/topics> .
named graph

What is the syntax of the following RDF statement? What does mean?

```sparql
@prefix dcterms: <http://purl.org/dc/terms/>.
GRAPH <http://inria.fr/topics/algebre>
{
  <http://inria.fr/rr/doc.html>
    dcterms:subject
      <http://data.bnf.fr/ark:/12148/cb121105993> .
}
```
Visit Leukocyte surface antigen CD53

1. See HTML data from:
   http://www.uniprot.org/uniprot/Q61451

2. Get RDF data from:
   http://www.uniprot.org/uniprot/Q61451.rdf

3. What is the syntax?

4. Translate into Turtle/N3:
   http://rdf-translator.appspot.com/

5. Any remark?
RDF in JSON (latter in the course)
RDF Semantics: subset of First Order Logic (FOL)

- Binary predicates
- Existential Quantification
- Conjunctions

**BUT:**
- no n-ary predicates
- no disjunction
- no negation
- no universal quantification

**SO:** RDF is monotonous
what is true and what can be inferred remains true as we add new statements

RDF formal semantics - theory models [http://www.w3.org/TR/rdf-mt/](http://www.w3.org/TR/rdf-mt/)
Chocolat Noir
Eclats de Noisettes

MAÎTRE CHOCOLATIER SUISSE

DEPUIS 1901

All Natural
sans additif artificiel, sans arôme artificiel

NET WT. 3.5 oz
100 g

ex:ingredients

ex:weight

dc:creator

rdf:about

rdf:label

rdf:type
openmodel

- extensible vocabulary based on URIs
- anyone can say anything about anything
link to the world
ACCESSING DATA ON THE WEB
Query RDF data
SPARQL Protocol and RDF Query Language
SPARQL in 3 parts
part 1: query language
part 2: result format
part 3: access protocol
SPARQL query

SELECT . . .
FROM . . .
WHERE { { . . . } }
query syntax based on Turtle

URI: <http://ns.Inria.fr/fabien.gandon#me>

QName: rdf:type foaf:name

Variable: ?x ?name $test

Blank Node: _:b1 _:b2

Literal: "Victor Hugo"@fr
          3.14
          true
          "12"^^xsd:integer
query syntax based on Turtle
e.g. persons at least 18-year old

PREFIX ex: <http://inria.fr/schema#>
SELECT ?person ?name
WHERE {
    ?person ex:name ?name .
    FILTER (?age > 17)
}
\[
\begin{align*}
\text{left}(x, y) & \quad \text{right}(z, v) \\
\text{left}(y, z) & \quad \text{right}(z, u) \\
\text{right}(u, v) & \\
\end{align*}
\]

\[
\begin{align*}
\text{left}(x, ?p) & \quad \text{left}(?p, z) \\
\end{align*}
\]
graph mapping / projection

classical three clauses:

– Select: clause to select the values to be returned
– Where: triple/graph pattern to match
– Filter: constraints expressed using test functions (XPath 2.0 or external)
SPARQL triples

• triples and question marks for variables:
  \(?x\) rdf:type ex:Person

• graph patterns to match:
  SELECT ?subject ?property ?value
  WHERE {?subject ?property ?value}

• a pattern is, by default, a conjunction of triples

  SELECT ?x WHERE
  { ?x rdf:type ex:Person .
  ?x ex:name ?name . }
question:

- Query:
  ```sparql
  SELECT ?name WHERE {
    ?x name ?name .
    ?x email ?email .
  }
  ```

- Base:
  ```sparql
  _:a name "Fabien" x2
  _:b name "Thomas"
  _:c name "Lincoln"
  _:d name "Aline"
  _:b email <mailto:thom@chaka.sn>
  _:a email <mailto:Fabien.Gandon@inria.fr>
  _:d email <mailto:avalandre@pachinko.jp>
  _:a email <mailto:bafien@fabien.info>
  ```

- Results ?
prefixes
to use namespaces:

```sparql
PREFIX mit: <http://www.mit.edu#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?student
WHERE {
  ?student mit:registeredAt ?x .
}
```

Base namespace: BASE <...>
compare...

prefix foaf: <http://xmlns.com/foaf/0.1/>

select * where {
  ?x foaf:name ?n
}

select * where {
  ?x <http://xmlns.com/foaf/0.1/name> ?n
}
Test online

• Connect to:
  https://corese.inria.fr/srv/template

• Query:
  ```sql
  prefix v: <http://www.inria.fr/2015/humans#>
  select * where
  {
    ?x a v:Person .
  }
  ```
to respect an attribute e.g. @fr , ^^xsd:integer

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?x ?f WHERE {
  ?x foaf:name "Fabien"@fr ; foaf:knows ?f .
}

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?x WHERE {
  ?x foaf:name "Fabien"@fr ;
  foaf:age "21"^^xsd:integer .
}

specify the language and the type of literals
same shortcuts as Turtle

triples with a common subject:

```
SELECT ?name ?fname
WHERE {
  ?x a Person;
  name ?name ;
  firstname ?fname ;
  author ?y . }
```

list of values

```
?x firstname "Fabien", "Lucien" .
```

blank node

```
[ firstname "Fabien"] OR [ ] firstname "Fabien"
```
Query DBpedia
Test on DBpedia

• Connect to:
  
  http://dbpedia.org/snorql/ or
  http://fr.dbpedia.org/sparql or ...
  http://wiki.dbpedia.org/Internationalization/Chapters

• Query:

  SELECT * WHERE {
  }

  LIMIT 10
SELECT ?president  ?name WHERE

  FILTER (lang(?name)="en") 
}

American presidents on Wikidata
https://query.wikidata.org/
dataset

PREFIX mit: <http://www.mit.edu#>
SELECT ?student
FROM NAMED <http://www.mit.edu/data1.rdf>
FROM NAMED <http://www.mit.edu/data2.rdf>
WHERE {
  GRAPH ?g {
    ?student mit:registeredAt ?x .
  }
}
alternative graph patterns

PREFIX mit: <http://www.mit.edu#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?student ?name
WHERE {
  ?student mit:registeredAt ?x .
  {
    {?
  }
  UNION
  {
  }
}
}
PREFIX mit: <http://www.mit.edu#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?student ?name
WHERE {
  ?student mit:registeredAt ?x .
  OPTIONAL {?student foaf:name ?name .}
}
question:

```
SELECT * WHERE {
  ?x :hasCreated ?doc .
  OPTIONAL {
    ?x :isMemberOf ?org
  }
}
```

Results

(1) x = :John ; doc = :d1 ; org = unbound ; age = unbound
(2) x = :John ; doc = :d2 ; org = unbound ; age = unbound
(3) x = :Jack ; doc = :d3 ; org = unbound ; age = unbound
(4) x = :Jim ; doc = :d4 ; org = :assoc ; age = 45
sort, filter and limit answers

PREFIX mit: <http://www.mit.edu#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?student ?name
WHERE {
    ?student mit:registeredAt ?x .
    ?student foaf:name ?name .
    FILTER (?age > 22)
}

ORDER BY ?name

LIMIT 20

OFFSET 20

students older than 22 years sorted by name
results from number #21 to #40
Result Modifiers

```
SELECT * WHERE
SELECT DISTINCT ?x ?y WHERE
ORDER BY ?x DESC(?y)
LIMIT 10
OFFSET 10
```
**question:**

```
select distinct ?x ?z
where {
  ?y :friend ?z
}
```

<table>
<thead>
<tr>
<th></th>
<th>:Jules :friend :Jim</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>:Jim :friend :Jack</td>
</tr>
<tr>
<td></td>
<td>:Jules :friend :James</td>
</tr>
<tr>
<td></td>
<td>:James :friend :Jack</td>
</tr>
</tbody>
</table>

**Result with distinct**

(1) \( x = :Jules \); \( z = :Jack \)

**Result without distinct**

(1) \( x = :Jules \); \( z = :Jack \)

(1) \( x = :Jules \); \( z = :Jack \)
question:

```
select ?doc ?date
where {
  ?pers :author ?doc
  ?doc :date ?date
}
order by ?date
desc(?doc)
```

Result

(1) doc = :d3 ; date = 2007-12-31
(2) doc = :d1 ; date = 2007-12-31
(3) doc = :d2 ; date = 2008-01-01
operators

• Inside the FILTER:
  – Comparators: <, >, =, <=, >=, !=
  – Tests on variables: isURI(?x), isBlank(?x), isLiteral(?x), bound(?x)
  – Regular expression regex(?x, "A.*")
  – Attributes and values: lang(), datatype(), str()
  – Casting: xsd:integer(?x)
  – External functions and extensions
  – Boolean combinations: &&, ||

• In the where WHERE: @fr, ^^xsd:integer

• In the SELECT: distinct
meaning?

prefix foaf: <http://xmlns.com/foaf/0.1/>
select ?x where {
  ?x foaf:knows ?y ;
  foaf:knows ?z .
filter (?y != ?z)
}
results?

select * where {
  ?x ex:age ?a
  filter (?a <= 30)
}

  ex:John ex:age "18" .
  ex:Jim  ex:age "20" .
  ex:Jack ex:age "22" .
  ex:Jude ex:age "35" .
other functions (v 1.1)
isNumeric(Val) test it is a numeric value
coaalesce(val,..., val) first valid value
IRI(Str)/URI(Str) to build an iri/uri from a string
BNODE(ID) to build a blank node
RAND() random value between 0 and 1
ABS(Val) absolute value
CEIL(Val), FLOOR(Val), ROUND(Val)
NOW() today’s date
DAY(Date), HOURS(Date), MINUTES(Date), MONTH(Date), SECONDS(Date), TIMEZONE(Date), TZ(Date), YEAR(Date) to access different parts of a date
MD5(Val), SHA1(Val), SHA256(Val), SHA384(Val), SHA512(Val) hash functions
string / literal functions (v1.1)

- **STRDT(value, type)**: build a typed literal
- **STRLANG(value, lang)**: build a literal with a language
- **CONCAT(lit1,...,litn)**: concatenate a list of literal
- **CONTAINS(lit1,lit2)**, **STRSTARTS(lit1,lit2)**, **STRENGS(lit1,lit2)**: to test string inclusion
- **SUBSTR(lit, start [,length])**: extract a sub string
- **ENCODE_FOR_URI (Str)**: encodes a string as URI
- **UCASE (Str)**, **LCASE (Str)**: uppercase and lowercase
- **STRLEN (Str)**: length of the string
e.g. DBpedia
question:

PREFIX ex: <http://www.example.abc#>

SELECT ?person

WHERE {
  FILTER(! (?type = ex:Man ))
}
minus
substract a pattern: remove from the results of PAT1 the results of PAT2 PAT1 minus {PAT2}

PREFIX ex: <http://www.exemple.abc#>
SELECT ?x
WHERE {
  { ?x rdf:type ex:Person }
  minus { ?x rdf:type ex:Man }
}

Remove results that are compatible: same variables have same values and at least one common variable
Quizz:

Could this query return \texttt{ex:a c:memberOf ex:b} \ ?

\begin{verbatim}
select * where {
  ?x c:memberOf ?org .
  minus { ex:a c:memberOf ex:b }
}
\end{verbatim}
not exists
check the absence of a pattern in the graph
PAT1 . filter(! exists {PAT2})

PREFIX ex: <http://www.exemple.abc#>

SELECT ?x

WHERE {
  filter (not exists {
    {?y ex:memberOf <Hell>}
  })
}
not exist vs. minus

Same results:

?x c:memberOf ?org . filter(! exists {?x c:author ?doc })
?x c:memberOf ?org . minus {?x c:author ?doc }

Different results:

?x c:memberOf ?org . filter(! exists {?y c:author ?doc })
?x c:memberOf ?org . minus {?y c:author ?doc }

Example: integrity constraint, coherence constraint, etc.
if... then... else

prefix foaf: <http://xmlns.com/foaf/0.1/>

select * where {
  filter ( 
    if (langMatches( lang(?name), "FR"),
      ?age>=18, ?age>=21) )
}
test a value is in/not in a list

prefix foaf: <http://xmlns.com/foaf/0.1/>
select * where {
  ?x foaf:name ?n .
  filter (?n in ("fabien", "olivier", "catherine") )
}
values
pre-defined bindings

select ?person where {
  ?person name ?name .
VALUES (?name)
  { "Peter" "Pedro" "Pierre" }
bind
allows a value to be assigned to a variable

PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>

SELECT ?title ?price
{
    ?x ns:discount ?discount
    BIND (?p*(1-?discount) AS ?price)}
  {?x dc:title ?title .}
} FILTER(?price < 20)
aggregates

Aggregation functions:
group by + count, sum, min, max, avg, group_concat or sample

Additional filter after aggregate: having()
aggregates

• Return one result when there is no group by
  select (\text{min}(\text{?price}) \text{ as } \text{?min}) \text{ where } \{ \text{?x ex:price } \text{?price} \}

• Return a result for each « group by » key
  select ?class (\text{min}(\text{?price}) \text{ as } \text{?min})
  \text{ where } \{ \text{?x a } \text{?class ; ex:price } \text{?price} \}
  \text{ group by } \text{?class}

• Count the number of results
  select (\text{count}(*) \text{ as } \text{?count}) \text{ where } \{ \text{?x ex:price } \text{?price} \}

• Count the number of distinct results
  select (\text{count(distinct}*) \text{ as } \text{?count})
  \text{ where } \{ [ \text{a } \text{?class ; ex:price } \text{?price } ] \}
aggregates

ex. average scores, grouped by the subject, but only where the mean is greater than 10

SELECT (AVG(?score) AS ?average)
WHERE { ?student score ?score . }
GROUP BY ?student
HAVING(AVG(?score) > 10)
aggregates

ex. members of organizations and number of other members they are linked to through them

select ?x (count(distinct ?y) as ?count) where {
  ?x c:memberOf ?o .
  ?y c:memberOf ?o
}
group by ?x
concatenate grouped values

select ?x (group_concat(?p) as ?prices)
where { ?x c:price ?p } group by ?x

?x = <book1> ; ?prices = "12 8 10"
?x = <pen8> ; ?prices = "5 6"

select ?x (group_concat(?p ; separator='; ') as ?prices) where { ?x c:price ?p } group by ?x

?x = <book1> ; ?prices = "12; 8; 10"
?x = <pen8> ; ?prices = "5; 6"
meaning?

prefix ex: <http://example.org/>
select ?x (count(?doc) as ?c)
where { ?x ex:author ?doc }
group by ?x
order by desc(count(?doc))
meaning?

```sparql
select distinct ?x ?c where {
  ?x a foaf:Person
  optional { ?x foaf:name ?n }
  optional { ?x foaf:mbox ?m }
  optional { ?x foaf:knows ?y }
  bind (if (bound(?n), 1, 0) + if (bound(?m), 1, 0) + if (bound(?y), 1, 0) as ?c)
}
order by desc(?c)
```
FLINT editor
prefix foaf: <http://xmlns.com/foaf/0.1/>

select ?f_of_f where {
  ?x foaf:name "Fabien Gandon" ;
}

paths
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>: sequence</td>
</tr>
<tr>
<td></td>
<td>: alternative</td>
</tr>
<tr>
<td>+</td>
<td>: one or several</td>
</tr>
<tr>
<td>*</td>
<td>: zero or several</td>
</tr>
<tr>
<td>?</td>
<td>: optional</td>
</tr>
<tr>
<td>^</td>
<td>: reverse</td>
</tr>
<tr>
<td>!</td>
<td>: negation</td>
</tr>
<tr>
<td>{min,max}</td>
<td>: length</td>
</tr>
</tbody>
</table>
paths

/ : sequence
| : alternative
+ : one or several
* : zero or several
? : optional
^ : reverse
! : negation
{min,max} : length

prefix foaf: <http://xmlns.com/foaf/0.1/>;
select ?network_fab where {
  ?x foaf:name "Fabien Gandon" ;
  foaf:knows* ?network_fab .
}
paths

/ : sequence
| : alternative
+ : one or several
* : zero or several
? : optional
^ : reverse
! : negation
{min,max} : length

prefix ex: <http://example.org/voc#>
select * where {
  ?x ^ex:hasParent ?y
}
paths

/ : sequence
| : alternative
+ : one or several
* : zero or several
? : optional
^ : reverse
! : negation
{min,max} : length

prefix ex: <http://example.org/voc#>
select * where {
  ?x !ex:hasParent ?y }

paths

/ : sequence
| : alternative
+ : one or several
* : zero or several
? : optional
^ : reverse
! : negation
{min,max} : length

prefix ex: <http://example.org/voc#>
prefix rdf: <...>
prefix rdfs: <...>
select * where {
  ?x rdf:type/rdfs:subClassOf* ex:Person}
prefix rdf: <...>

select ?val where {
  ?list rdf:rest*/rdf:first ?val
}

/ : sequence
| : alternative
+ : one or several
* : zero or several
? : optional
^ : reverse
! : negation
{min,max} : length
quizz

What expression should we use to find the ?x related to ?y by paths composed of any composition of properties foaf:knows or/or rdfs: seeAlso?

- ?x (foaf:knows | rdfs:seeAlso)+ ?y
- ?x foaf:knows+ | rdfs:seeAlso+ ?y
- ?x (foaf:knows / rdfs:seeAlso)+ ?y
select expression

select ?x (year(?date) as ?year)
where {
  ?x birthdate ?date .
}
meaning?

prefix foaf: <http://xmlns.com/foaf/0.1/>

select ?x (if (bound(?n), ?n, "JohnDoe") as ?m)

where {
  ?x foaf:knows ?y
  optional { ?y foaf:name ?n }
}
prefix ex: <http://example.org/>
select ?x (avg(?a) as ?b)
where {
  ?y ex:age ?a
}
group by ?x
subquery / nested query

```sparql
select ?name where {
  {select (max(?age) as ?max)
    where { ?person age ?age }
  }
  ?senior age ?max
  ?senior name ?name
}
```
Service Clause
remote access to a SPARQL endpoint

prefix r: <http://fr.dbpedia.org/resource/>
prefix p: <http://fr.dbpedia.org/property/>
prefix o: <http://dbpedia.org/ontology/>
prefix geo: <http://rdf.insee.fr/def/geo#>

select * where {
  service <http://fr.dbpedia.org/sparql> { r:Auguste p:succ ?s ; o:wife ?w }
}

prefix geo: <http://rdf.insee.fr/def/geo#>

select * where {
  service <http://rdf.insee.fr/sparql> {
    ?region rdf:type geo:Region ; geo:nom "Bourgogne" ; ?p ?v
  }
}
SPARQL result
failure/ success
values found
result formats

- a **binding** i.e. list of all the selected values (SELECT) for each answer found; (stable XML format ; e.g. for XSLT transformations)
- RDF **sub-graphs** for each answer found (RDF/XML format ; e.g. for application integration)
- JSON (eg. ajax web applications)
- CSV/TSV (eg. export)
example of binding

results for previous query in XML

```xml
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="student"/>
  </head>
  <results ordered="false" distinct="false">
    <result>
      <binding name="student">
        <uri>http://www.mit.edu/data.rdf#ndieng</uri>
      </binding>
    </result>
    <result>
      <binding name="student">
        <uri>http://www.mit.edu/data.rdf#jdoe</uri>
      </binding>
    </result>
  </results>
</sparql>
```
example of JSON result for Internet Media Type application/sparql-results+json

```json
{
  "head": { "vars": [ "book", "title" ] },
  "results": {
    "bindings": [
      {
        "title": { "type": "literal", "value": "Harry Potter and the Half-Blood Prince" }
      },
      {
        "title": { "type": "literal", "value": "Harry Potter and the Deathly Hallows" }
      },
      {
        "title": { "type": "literal", "value": "Harry Potter and the Philosopher's Stone" }
      }
    ]
  }
}
```
example of CSV result
for Internet Media Type text/csv

uri,name
http://fabien.info,Gandon
http://inria.fr,Inria

PS: same principle for TSV
Check the existence of at least one answer/result

does not list all the results but just checks if there exists at least one (true/false)

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
ASK { ?person foaf:age 111 . }
free description

PREFIX mit: <http://www.mit.edu#>
DESCRIBE ?student
{ ?student rdf:type mit:Student . }
construct RDF as result

PREFIX mit: <http://www.mit.edu#>
PREFIX corp: <http://mycorp.com/schema#>

CONSTRUCT

{ ?student rdf:type corp:FuturExecutive . } 

WHERE

{ ?student rdf:type mit:Student . }

prefix dbpedia-owl: <http://dbpedia.org/ontology/>
prefix dbpedia-pro: <http://dbpedia.org/property/>
prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix geonames: <http://sws.geonames.org/>
prefix wgs84_pos: <http://www.w3.org/2003/01/geo/wgs84_pos#>
prefix gephi: <http://gephi.org/>
prefix owl: <http://www.w3.org/2002/07/owl#>

construct {
  ?city1 gephi:label ?city_name1 ;
  gephi:longitude ?long1 ;
  gephi:latitude ?lat1 ;
  gephi:image ?im1 ;
  gephi:language ?language ;
  gephi:populationTotal ?population1 .
}

?city2 gephi:label ?city_name2 ;
  gephi:longitude ?long2 ;
  gephi:latitude ?lat2 ;
  gephi:image ?im2 ;
  gephi:language ?language ;
  gephi:populationTotal ?population2 .

where {
  service <http://live.dbpedia.org/sparql/> {
    
  }
}
RDF Datasets

• Query an RDF base with several graphs
• Named graphs with URIs
• Default graph
• Identify, characterize, etc. the graphs that are queried
select ?g where
{ graph ?g { ?p inria:author ?doc } }
select ?g from named <http://g1> where {
  graph ?g {
    ?p inria:author ?doc
  }
}
select ?g from <http://g1> where
{ ?p inria:author ?doc }
select * where { graph <http://g1> { ?p inria:author ?doc } }
Select ?g where { ?g inria:date 2016 .
graph ?g { ?p inria:author ?doc } }

http://ns.inria.fr/fabien.gandon#me

http://inria.fr/rr/doc1.html

http://inria.fr/rr/doc2.html

https://www.w3.org/People/Berners-Lee/card#i

http://g1

http://g2

inria:author

inria:author

inria:date 2016

inria:date 2015
Quizz

On which graph(s) is calculated ?x ?p ?y

On which graph(s) is calculated graph ?g { ?y ?q ?z } 

prefix ex:  &lt;http://example.org/&gt;
select *
from ex:g1
from named ex:g2
where {
    graph ?g { ?y ?q ?z} 
}
SPARQL Update
Update language for RDF graphs
CRUD: Create Read Update Delete
SPARQL Update
Add or remove triples or graphs

LOAD <http://example.org/dataset>

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX ex:   <http://example.org/>

INSERT DATA {
  ex:Fab foaf:name "Fabien" ;
  foaf:knows ex:Cathy, ex:Olivier .
}
SPARQL Update
Add and remove triples

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX ex: <http://example.org/>

INSERT DATA {
  ex:Fab foaf:name "Fabien" ;
  foaf:knows ex:Cathy, ex:Olivier .
}

DELETE DATA {
  ex:Fab foaf:knows ex:Bill .
}
SPARQL Update

Search and add triples

PREFIX ex: <http://example.org/>

INSERT {
  ?x a ex:Artist
}

WHERE {
  ?x a ex:Musician
}
SPARQL Update

Search and remove triples

PREFIX ex: <http://example.org/>
DELETE {
  ?x a ex:Musician
}
WHERE {
  ?x a ex:Musician
}
SPARQL Update

Search, remove and add triples

PREFIX ex: <http://example.org/>

DELETE {
    ?x a ex:Musician
}

INSERT {
    ?x a ex:Artist
}

WHERE {
    ?x a ex:Musician
}
SPARQL Update

Search, remove and add triples

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX ex: <http://example.org/>

DELETE { ?person foaf:firstName 'Fred' }

INSERT { ?person foaf:firstName 'Frederique' }

WHERE

{ ?person a ex:Woman .

  ?person foaf:firstName 'Fred'
}


SPARQL Update
USING like FROM :source graph
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
DELETE { ?person foaf:firstName 'Fred' }
USING <g1>
WHERE { ?person foaf:firstName 'Fred' }
SPARQL Update

WITH target graph

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

WITH <g1>

INSERT { ?person foaf:firstName 'Frederique' } 

WHERE { ?person foaf:firstName 'Fred' }
SPARQL Update on graphs

LOAD <documentURI> [ INTO GRAPH <uri> ]
CLEAR [ SILENT ] (GRAPH <uri> | DEFAULT | NAMED | ALL )
DROP [ SILENT ] (GRAPH <uri> | DEFAULT | NAMED | ALL )
CREATE [ SILENT ] GRAPH <uri>
meaning?

prefix ex: <http://example.org/>
insert { ?y ex:hasParent ?x }
where { ?x ex:hasChild ?y }
prefix ex: <http://example.org/>
delete { ?x ex:age ?a }
insert { ?x ex:age ?i }
where {
  select ?x (xsd:integer(?a) as ?i)
  where {
    ?x ex:age ?a
    filter(datatype(?a) = xsd:string)
  }
}
SPARQL protocol

exchange queries and their results through the web
HTTP

SPARQL
Linked Data Platform

HTTP access to LD resources & containers
get, post, put, delete resources from LD servers.

GET /people/fab HTTP/1.1
Host: data.inria.fr

PUT http://data.inria.fr/people/fab HTTP/1.1
Host: data.inria.fr
Content-Type: text/turtle

<fab> a foaf:Person ;
  rdfs:label "Fabien" ;
  foaf:mbox <fabien.gandon@inria.fr> .
Corese KGram

Loaded files:

/Users/ereal_5/Desktop/demoquery/history.ttl

Logs:

Reset...
done.

Loading ttl file from path: /Users/ereal_5/Desktop/demoquery/history.ttl
SEMANTIC WEB
do not read the following sign
you loose
we identify and interpret information, machines don’t.
180°C + Allons, Enfants de la Patrie! Le jour de gloire est arrivé = ?

publish the data schemas
know the meaning of data
to find out what can be done with it.
what is the last document you read?
documents
your answer relies on a shared ontology you infer from it we all understood
Document

Book

Novel  Short story
kind of

"document"

"book"
"livre"

"novel"
"roman"

"short story"
"nouvelle"
formalized ontological knowledge
languages to formalize ontologies
stack of standards

RDFS

communication

HTTP

web de données

URI

référence

adresse

W3C®
RDFS means RDF Schema
RDFS provides primitives to write lightweight ontologies.
RDFS to define classes of resources and organize their hierarchy
RDFS to define relations between resources, their signature and organize their hierarchy.
F^O \rightarrow R \iff G_F \leq G_R

mapping modulo an ontology
an old schema of RDFS

W3C http://www.w3.org/TR/2000/CR-rdf-schema-20000327/
RDFS means « RDF Schema »

- standard Vocabulary to write lightweight ontologies
- written in RDF
- query RDFS (schemas) in SPARQL
associate a namespace with a vocabulary

complete URI of classes, properties, etc. or shortcuts:

```xml
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  (...)
</rdf:RDF>
```
associate a namespace with a vocabulary
complete URI of classes, properties, etc. or shortcuts:

```xml
  xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  (...)
</rdf:RDF>

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@base <http://inria.fr/2005/humans.rdfs>
  (...)
```
RDFS: meta-ontology / meta-vocabulary

Standard RDF classes and properties to define our own classes and properties.

A schema to describe schemas.

Query a schema in SPARQL
RDFS Classes
meta-classes and some of their links

-> sub-type / subsumption
-------> typing/ instance
RDFS Classes
meta-classes and some of their links

[rdfs:Class] -> [rdfs:Resource]
[rdfs:Data] -> [rdfs:Resource]
[rdf:HTML] -> [rdfs:Literal]
[rdf:XMLLiteral] -> [rdfs:Literal]

→ sub-type / subsumption
-----→ typing/ instance
RDFS Classes
meta-classes and some of their links

- rdfs:Literal
- rdfs:Class
- rdfs:Resource
- rdf:Property
- rdf:ContainerMembershipProperty
- rdf:HTML
- rdf:langString
- rdf:Datatype
- rdf:XMLLiteral
- rdf:Statement
- rdf:List
- rdf:nil
- rdf:Container
- rdf:Alt
- rdf:Bag
- rdf:Seq

sub-type / subsumption
----- typing/ instance
RDFS properties
meta-properties and some of their links

sub-type / subsumption
-------> typing/ instance
RDFS properties
meta-properties and some of their links

- rdfs:Resource
- rdfs:Literal
- rdfs:Class
- rdfs:subClassOf
- rdf:Property
- rdfs:domain
- rdfs:range
- rdfs:subPropertyOf

→ sub-type / subsumption
-----→ typing/ instance
RDFS properties
meta-properties and some of their links
RDFS properties
meta-properties and some of their links

sub-type / subsumption
------> typing/ instance
RDFS properties
meta-properties and some of their links

- rdfs:Resource
- rdfs:Literal
- rdfs:Class
- rdfs:subClassOf
- rdf:Property
- rdf:List
- rdf:Container
- rdf:ContainerMembershipProperty
- rdfs:member
- rdf:_1
- rdf:_2
- rdf:_3

→ sub-type / subsumption
-----→ typing/ instance
RDFS properties
meta-properties and some of their links
RDFS contains primitives to... (several answers possible)

- describe classes of resources
- describe formulas of calculation for values of properties
- describe types of properties of resources
- document definitions in natural language
- sign and authenticate the authors of the definitions of classes and properties
semantics
everything is a resources

IF x p y THEN
x rdf:type rdfs:Resource

IF x p y THEN
y rdf:type rdfs:Resource
Define classes of resources

- declare and name classes
- organize their hierarchy
- multiple inheritance hierarchy
instances of rdfs:Class
the class of classes is in RDFS namespace.

```xml
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdfs:Class rdf:ID="Man">
    <rdfs:subClassOf rdf:resource="#Person"/>
    <rdfs:subClassOf rdf:resource="#Male"/>
  </rdfs:Class>
</rdf:RDF>
```
instances of rdfs:Class
the class of classes is in RDFS namespace.

  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdfs:Class rdf:ID="Man">
    <rdfs:subClassOf rdf:resource="#Person"/>
    <rdfs:subClassOf rdf:resource="#Male"/>
  </rdfs:Class>
</rdf:RDF>
instances of rdfs:Class

the class of classes is in RDFS namespace.

```xml
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdfs:Class rdf:ID="Man">
    <rdfs:subClassOf rdf:resource="#Person"/>
    <rdfs:subClassOf rdf:resource="#Male"/>
  </rdfs:Class>
</rdf:RDF>
```
instances of **rdfs:Class**
the class of classes is in RDFS namespace.

```xml
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">  
  <rdfs:Class rdf:ID="Man">  
    <rdfs:subClassOf rdf:resource="#Person"/>  
    <rdfs:subClassOf rdf:resource="#Male"/>  
  </rdfs:Class>  
</rdf:RDF>
```

@prefix **rdfs**: <http://www.w3.org/2000/01/rdf-schema#> .  
@base <http://inria.fr/2005/humans.rdfs> .  
<Man> a **rdfs:Class** ;  
  **rdfs:subClassOf** <Person>, <Male> .
semantics

1. Every class is a subclass of rdfs:Resource
   \[\text{IF } c \text{ rdf:type rdfs:Class } \implies c \text{ rdfs:subClassOf rdfs:Resource}\]

2. Type propagation
   \[\text{IF } c_2 \text{ rdfs:subClassOf } c_1 \text{ AND } x \text{ rdf:type } c_2
   \implies x \text{ rdf:type } c_1\]

3. Reflexivity of subsumption
   \[\text{IF } c \text{ rdf:type rdfs:Class}
   \implies c \text{ rdfs:subClassOf } c\]

4. Transitivity of subsumption
   \[\text{IF } c_2 \text{ rdfs:subClassOf } c_1 \text{ AND } c_3 \text{ rdfs:subClassOf } c_2
   \implies c_3 \text{ rdfs:subClassOf } c_1\]
What is defined and derived from these definitions?

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> 
@base <http://inria.fr/2005/humans.rdfs> 

<B> rdfs:subClassOf <A> . 
<C> rdfs:subClassOf <A> . 
<D> rdfs:subClassOf <B> . 
<D> rdfs:subClassOf <C> .
types of relations (properties) between resources

- declare and name the types of relationships/arcs/properties
- organize their hierarchy
- multiple inheritance hierarchy
instances of rdf:Property

the class of properties was placed in the RDF namespace because triples are a construction of RDF.

```xml
  xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Property rdf:ID="hasMother">
    <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  </rdf:Property>
</rdf:RDF>
```
instances of rdf:Property

the class of properties was placed in the RDF namespace because triples are a construction of RDF.

```xml
  xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">  
   <rdf:Property rdf:ID="hasMother">
     <rdfs:subPropertyOf rdf:resource="#hasParent"/>
   </rdf:Property>
</rdf:RDF>
```
instances of **rdf:Property**

the class of properties was placed in the RDF namespace because triples are a construction of RDF.

```xml
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Property rdf:ID="hasMother">
    <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  </rdf:Property>
</rdf:RDF>
```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://inria.fr/2005/humans.rdfs> .

<hasMother> a rdf:Property ;
  rdfs:subPropertyOf <hasParent> .
```
instances of `rdf:Property`

the class of properties was placed in the RDF namespace because triples are a construction of RDF.

```xml
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Property rdf:ID="hasMother">
    <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  </rdf:Property>
</rdf:RDF>
```

@prefix `rdf:` <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix `rdfs:` <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://inria.fr/2005/humans.rdfs> .

`<hasMother>` a `rdf:Property` ;
  `rdfs:subPropertyOf` `<hasParent>` .
semantics

1. Type propagation
   \[ \text{IF } p_2 \text{ rdfs:subPropertyOf } p_1 \text{ AND } x \ p_2 \ y \]
   \[ \text{THEN } x \ p_1 \ y \]

2. Reflexivity of subsumption
   \[ \text{IF } p \text{ rdf:type } \text{rdf:Property} \]
   \[ \text{THEN } p \text{ rdfs:subPropertyOfOf } p \]

3. Transitivity of subsumption
   \[ \text{IF } p_2 \text{ rdfs:subPropertyOfOf } p_1 \text{ AND } p_3 \text{ rdfs:subPropertyOfOf } p_2 \]
   \[ \text{THEN } p_3 \text{ rdfs:subPropertyOfOf } p_1 \]
What can be said about the properties defined below?

@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#>  .
@base <http://inria.fr/2005/humans.rdfs>  .
<P2>  rdfs:subPropertyOf  <P1>  .  
<P3>  rdfs:subPropertyOf  <P1>  .  
<P4>  rdfs:subPropertyOf  <P2>,  <P3>  .
signature: classes to which a property applies

Differences with object-oriented languages:

- RDFS does not define a class in terms of properties possessed by its instances.
- RDFS defines a property in terms of classes of resources to which it applies.
signature of a binary relation

- class of departure of the relation (domain)
- class of the arrival of the relation (range / co-domain)
domain and range

Class of departure or domain: rdfs:domain
Class of arrival, co-domain or range: rdfs:range

```xml
  xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Property rdf:ID="hasMother">
    <rdfs:subPropertyOf rdf:resource="#hasParent"/>
    <rdfs:domain rdf:resource="#Human"/>
    <rdfs:range rdf:resource="#Woman"/>
  </rdf:Property>
</rdf:RDF>
```
domain and range

Class of departure or domain: rdfs:domain
Class of arrival, co-domain or range: rdfs:range

```xml
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Property rdf:ID="hasMother">
    <rdfs:subPropertyOf rdf:resource="#hasParent"/>
    <rdfs:domain rdf:resource="#Human"/>
    <rdfs:range rdf:resource="#Woman"/>
  </rdf:Property>
</rdf:RDF>
```
domain and range
Class of departure or domain: rdfs:domain
Class of arrival, co-domain or range: rdfs:range

```xml
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Property rdf:ID="hasMother">
    <rdfs:subPropertyOf rdf:resource="#hasParent"/>
    <rdfs:domain rdf:resource="#Human"/>
    <rdfs:range rdf:resource="#Woman"/>
  </rdf:Property>
</rdf:RDF>
```
domain and range
Class of departure or domain: rdfs:domain
Class of arrival, co-domain or range: rdfs:range

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://inria.fr/2005/humans.rdfs> .

<hasMother> a rdf:Property ;
   rdfs:subPropertyOf <hasParent> ;
   rdfs:domain <Human> ;
   rdfs:range  <Woman> .
domain and range

Class of departure or domain: rdfs:domain
Class of arrival, co-domain or range: rdfs:range

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://inria.fr/2005/humans.rdfs> .

<hasMother> a rdf:Property ;
  rdfs:subPropertyOf <hasParent> ;
  rdfs:domain <Human> ;
  rdfs:range <Woman> .
domain and range

Class of departure or domain: rdfs:domain
Class of arrival, co-domain or range: rdfs:range

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://inria.fr/2005/humans.rdfs> .

<hasMother> a rdf:Property ;
   rdfs:subPropertyOf <hasParent> ;
   rdfs:domain  <Human> ;
   rdfs:range   <Woman> .
multiple domains and ranges

- conjunction of domains and ranges.
- the effective domain is the intersection of declared and inherited domains.
- the effective range is the intersection of declared and inherited ranges.
semantics

1. Type inference (domain)
   IF \( p \) rdfs:domain \( d \) AND \( x \) p \( y \)
   THEN \( x \) rdf:type \( d \)

2. Type inference (range)
   IF \( p \) rdfs:range \( r \) AND \( x \) p \( y \)
   THEN \( y \) rdf:type \( r \)
question

If I use the property author on a car what happens?
What can be said about the properties defined below?

@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@base <http://inria.fr/2005/humans.rdfs>
<P1> rdfs:subPropertyOf  <P2> .
<P2> rdfs:domain   <B> ;  rdfs:range    <C> .
<P1> rdfs:domain    <A> .
rdfs:label

A resource may have one or more labels in one or more natural language.

```
<rdf:Property rdf:ID='name'>
  <rdfs:domain rdf:resource='Person'/>
  <rdfs:range rdf:resource='&rdfs;Literal'/>
  <rdfs:label xml:lang='fr'>nom</rdfs:label>
  <rdfs:label xml:lang='fr'>nom de famille</rdfs:label>
  <rdfs:label xml:lang='en'>name</rdfs:label>
</rdf:Property>
```

```
<name> a rdf:Property ;
  range rdfs:Literal ; domain <Person> ;
  label "nom"@fr, "nom de famille"@fr, "name"@en .
```
textual labels attached to resources
any resource may have one or more labels in one or more languages

```xml
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Property rdf:ID='name'>
    <rdfs:label xml:lang='fr'>nom</rdfs:label>
    <rdfs:label xml:lang='fr'>nom de famille</rdfs:label>
    <rdfs:label xml:lang='en'>name</rdfs:label>
  </rdf:Property>
</rdf:RDF>
```
textual labels attached to resources
any resource may have one or more labels in one or more languages

```xml
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdf:Property rdf:ID='name'>
    <rdfs:label xml:lang='fr'>nom</rdfs:label>
    <rdfs:label xml:lang='fr'>nom de famille</rdfs:label>
    <rdfs:label xml:lang='en'>name</rdfs:label>
  </rdf:Property>
</rdf:RDF>
```
textual labels attached to resources
any resource may have one or more labels in one or more languages

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://inria.fr/2005/humans.rdfs> .
<name> a rdf:Property ;
    rdfs:label "nom"@fr, "nom de famille"@fr, "name"@en .
textual labels attached to resources
any resource may have one or more labels in one or more languages

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://inria.fr/2005/humans.rdfs> .
<name> a rdf:Property ;
    rdfs:label "nom"@fr, "nom de famille"@fr, "name"@en .
rdfs:comment & rdfs:seeAlso

**Comments** provide definitions and explanations in natural language

```xml
<Woman> a rdfs:Class ; rdfs:subClassOf <Person> ;
rdfs:comment "adult female person"@en ;
rdfs:comment "une adulte de sexe féminin"@fr .
</rdfs:Class>

see also...

```xml
<Man> a rdfs:Class ; rdfs:seeAlso <Woman> .
</rdfs:Class>
```
textual comments attached to resources

comments provide definitions and explanations in natural language

```xml
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdfs:Class rdf:about="#Woman">
    <rdfs:comment xml:lang='fr'>personne adulte de sexe féminin</rdfs:comment>
    <rdfs:comment xml:lang='en'>female adult person</rdfs:comment>
  </rdfs:Class>
</rdf:RDF>
```
textual comments attached to resources

comments provide definitions and explanations in natural language

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://inria.fr/2005/humans.rdfs> .

<Woman> a rdfs:Class ;
  rdfs:comment "adult femal person"@en ;
  rdfs:comment "une adulte de sexe féminin"@fr .
references between resources
invitation to check another resource

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
  <rdfs:Class rdf:about='#Man'>
    <rdfs:seeAlso rdf:resource='#Woman'/>
  </rdfs:Class>
</rdf:RDF>
```
references between resources
invitation to check another resource

```xml
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
 <rdfs:Class rdf:about="#Man">
   <rdfs:seeAlso rdf:resource="#Woman"/>
 </rdfs:Class>
</rdf:RDF>

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://inria.fr/2005/humans.rdfs> .
<Man> a rdfs:Class ; rdfs:seeAlso <Woman> .
```
What could we add to this schema (several answers)?

@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@base <http://inria.fr/2005/humans.rdfs>
<p1> a rdf:Property ; rdfs:label "age"@fr .
<c1> a rdfs:Class; rdfs:comment "un être humain"@fr .

- <p1> rdfs:label "prénom"@fr .
- <c1> rdfs:comment "a human being"@fr .
- <c1> rdfs:label "personne"@fr .
- <p1> rdfs:label "age"@en .
- <c1> rdfs:label "woman"@en .
- <c1> rdfs:label "persona"@es .
example of RDFS classes

```xml
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns="http://www.w3.org/2000/01/rdf-schema#">
  <Class rdf:ID="Man">
    <subClassOf rdf:resource="#Person"/>
    <subClassOf rdf:resource="#Male"/>
    <label xml:lang="en">man</label>
    <comment xml:lang="en">an adult male person</comment>
  </Class>
</rdf:RDF>
```

<Man> a Class ; subClassOf <Person>, <Male> .
example of RDFS properties

```xml
 xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns  ="http://www.w3.org/2000/01/rdf-schema#">
 <rdf:Property rdf:ID="hasMother">
   <subPropertyOf rdf:resource="#hasParent"/>
   <range rdf:resource="#Female"/>
   <domain rdf:resource="#Human"/>
   <label xml:lang="en">has for mother</label>
   <comment xml:lang="en">to have for parent a female.</comment>
 </rdf:Property>
</rdf:RDF>
```

```sparql
<hasMother> a rdf:Property ;
   subPropertyOf <hasParent> ;
   range <Female> ;
   domain <Human> .
```
example of RDF using this schema
usage and references to schemas
in a resource description

```xml
<rdf:RDF xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:h="http://inria.fr/2005/humans.rdfs#"
  xml:base="http://inria.fr/2005/humans.rdfs-instances" >
  <rdf:Description rdf:ID="Lucas">
    <h:hasMother rdf:resource="#Laura"/>
  </rdf:Description>
</rdf:RDF>
```
usage and references to schemas
in a resource description

<rdf:RDF xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:h="http://inria.fr/2005/humans.rdfs#"
xml:base="http://inria.fr/2005/humans.rdfs-instances" >
 <rdf:Description rdf:ID="Lucas">
   <h:hasMother rdf:resource="#Laura"/>
 </rdf:Description>
</rdf:RDF>
usage and references to schemas
in a resource description

<rdf:RDF xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:h="http://inria.fr/2005/humans.rdfs#"
    xml:base="http://inria.fr/2005/humans.rdfs-instances" >
    <rdf:Description rdf:ID="Lucas">
        <h:hasMother rdf:resource="#Laura"/>
    </rdf:Description>
</rdf:RDF>
usage and references to schemas in a resource description

<rdf:RDF xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:h="http://inria.fr/2005/humans.rdfs#"
  xml:base="http://inria.fr/2005/humans.rdfs-instances" >
  <rdf:Description rdf:ID="Lucas">
    <h:hasMother rdf:resource="#Laura"/>
  </rdf:Description>
</rdf:RDF>
usage and references to schemas
in a resource description

```xml
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
       xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
       xmlns:h="http://inria.fr/2005/humans.rdfs#"
       xml:base="http://inria.fr/2005/humans.rdfs-instances" >
  <h:Man rdf:ID="Lucas">
    <h:hasMother rdf:resource="#Laura"/>
  </h:Man>
</rdf:RDF>
```
usage and references to schemas
in a resource description

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:h="http://inria.fr/2005/humans.rdfs#"
  xml:base="http://inria.fr/2005/humans.rdfs-instances" >
  <h:Man rdf:ID="Lucas">
    <h:hasMother rdf:resource="#Laura"/>
  </h:Man>
</rdf:RDF>
```

RDF/XML

```
@prefix h: <http://inria.fr/2005/humans.rdfs#> .
@base <http://inria.fr/2005/humans.rdfs-instances> .
<Lucas> a h:Man; h:hasMother <Laura> .
```
usage and references to schemas in a resource description

```xml
<RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:h="http://inria.fr/2005/humans.rdfs#"
     xml:base="http://inria.fr/2005/humans.rdfs-instances">
    <Man rdf:ID="Lucas">
        <hasMother rdf:resource="#Laura"/>
    </Man>
</RDF>
```

@prefix h: <http://inria.fr/2005/humans.rdfs#> .
@base <http://inria.fr/2005/humans.rdfs-instances> .
<Lucas> a h:Man; h:hasMother <Laura> .
usage and references to schemas
in a resource description

```xml
<rdf:RDF xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:h="http://inria.fr/2005/humans.rdfs#"
    xml:base=" http://inria.fr/2005/humans.rdfs-instances" >
    <h:Man rdf:ID="Lucas">
        <h:hasMother rdf:resource="#Laura"/>
    </h:Man>
</rdf:RDF>
```

RDF/XML

Turtle

```turtle
@prefix h: <http://inria.fr/2005/humans.rdfs#> .
@base <http://inria.fr/2005/humans.rdfs-instances> .
<Lucas> a h:Man; h:hasMother <Laura> .
```
usage and references to schemas
in a resource description

<rdf:RDF xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns="http://inria.fr/2005/humans.rdfs#"
  xml:base=" http://inria.fr/2005/humans.rdfs-instances" >
  <Man rdf:ID="Lucas">    
    <hasMother rdf:resource="#Laura"/>
  </Man>
</rdf:RDF>
usage and references to schemas in a resource description

```xml
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns="http://inria.fr/2005/humans.rdfs#"
  xml:base=" http://inria.fr/2005/humans.rdfs-instances" >
  <Man rdf:ID="Lucas">
    <hasMother rdf:resource="#Laura"/>
  </Man>
</rdf:RDF>
```
usage and references to schemas
in a resource description

```xml
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns="http://inria.fr/2005/humans.rdfs#"
  xml:base = "http://inria.fr/2005/humans.rdfs-instances" >
  <Man rdf:ID="Lucas">
    <hasMother rdf:resource="#Laura"/>
  </Man>
</rdf:RDF>
```

```turtle
@prefix : <http://inria.fr/2005/humans.rdfs#> .
@base <http://inria.fr/2005/humans.rdfs-instances> .
<Lucas> a :Man; :hasMother <Laura> .
```
SEMANTIC WEB
OWL provides additional primitives for heavyweight ontologies
Web Ontology Language (OWL)

- a W3C recommendation
- additional primitives for more complex ontologies.
- richer definitions of classes and properties.
- perform more inferences, draw more conclusions.
namespace and prefix for OWL

http://www.w3.org/2002/07/owl#

- namespace of the OWL primitives
- same principle as RDFS
- `owl:` prefix in the rest of the slides
OWL in one...

- Algebraic properties:
  - Union
  - Disjunction
  - Intersection
  - Complement
- Disjoint properties:
  - Restriction
- Qualified cardinality:
  - 1..1
- Individual property negation
- Chained property
- Disjoint union
- Value restrict.
- Equivalence
- Enumeration
- Cardinality

...
enumerated class \{a,b,c,d,e\}

define a class by providing all its members

```xml
<owl:Class rdf:id="EyeColor">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:ID="Blue"/>
    <owl:Thing rdf:ID="Green"/>
    <owl:Thing rdf:ID="Brown"/>
    <owl:Thing rdf:ID="Black"/>
  </owl:oneOf>
</owl:Class>
```

```prolog
<EyeColor> rdf:type owl:Class ;
  owl:oneOf
    ( <Blue> <Green> <Brown> <Black> ) .
```
classes defined by union of other classes

```xml
<owl:Class rdf:id="LegalAgent">
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Person"/>
    <owl:Class rdf:about="#Group"/>
  </owl:unionOf>
</owl:Class>

<LegalAgent> rdf:type owl:Class ;
  owl:unionOf ( <Person> <Group> ) .
```
classes defined by intersection of other classes

```xml
<owl:Class rdf:id="Man">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Person"/>
    <owl:Class rdf:about="#Male"/>
  </owl:unionOf>
</owl:Class>

<Man> rdf:type owl:Class ;
  owl:intersectionOf ( <Person> <Male> ) .
```
complement of a class

<owl:Class rdf:ID="Inedible">
  <owl:complementOf rdf:resource="#Edible"/>
</owl:Class>

<Inedible> rdf:type owl:Class ;
  owl:complementOf <Edible> .
disjunction of classes

<owl:Class rdf:ID="Square">
  <owl:disjointWith rdf:resource="#Circle"/>
</owl:Class>

<Square> rdf:type owl:Class ;
  owl:disjointWith <Circle> .
disjunction of several classes

```xml
<owl:AllDisjointClasses>
  <owl:members rdf:parseType="Collection">
    <owl:Class rdf:about="#Square"/>
    <owl:Class rdf:about="#Circle"/>
    <owl:Class rdf:about="#Triangle"/>
  </owl:members>
</owl:AllDisjointClasses>

[] rdf:type owl:AllDisjointClasses ;
owl:members
  ( <Square> <Circle> <Triangle> ) .
```
disjoint union of several classes

```
<owl:Class rdf:about="Passenger">
  <owl:disjointUnionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Adult"/>
    <owl:Class rdf:about="#Child"/>
    <owl:Class rdf:about="#Pet"/>
  </owl:disjointUnionOf>
</owl:Class>

<Passenger rdf:type owl:Class ;
  owl:disjointUnionOf ( <Adult> <Child> <Pet> ) .
```
What can we deduce?

ex:Man owl:intersectionOf (ex:Male ex:Human) .
ex:Woman owl:intersectionOf (ex:Female ex:Human) .
ex:Human owl:unionOf (ex:Man ex:Woman) .
ex:Jane a ex:Human .
ex:John a ex:Man .
ex:James a ex:Male .
ex:Jane a ex:Female .
What are we defining?

ex:p a rdf:Property ;
  rdfs:domain [ 
    a owl:Class ;
    owl:unionOf (ex:Human ex:Software) 
  ] .
quizz

What are we defining and inferring?

@prefix ex: <http://example.org/>

ex:GrandFather rdfs:subClassOf [  
a owl:Class ;  
  owl:intersectionOf ( ex:Parent ex:Man )  
] .

ex:Jim a ex:Man, ex:Parent .
ex:Jack a ex:GrandFather .
types of properties

• `owl:ObjectProperty` are relations between resources only e.g. `hasParent(#thomas,#stephan)`
• `owl:DatatypeProperty` have a literal value possibly typed `ex:hasAge(#thomas,16^^xsd:int)`
• `owl:AnnotationProperty` are ignored in inferences and used for documentation and extensions
**symmetric property**
a relation that, as soon as it exists, exists in both directions (e.g. to be married)
\[ x \mathrel{R} y \implies y \mathrel{R} x \]

```xml
<owl:SymmetricProperty rdf:ID="hasSpouse" />

<hasSpouse> a owl:SymmetricProperty .
```
asymmetric property

A relation that, as soon as it exists, exists in only one direction (e.g. parent)

\[ x R y \implies \neg y R x \]

```xml
<owl:AsymmetricProperty rdf:ID="hasChild" />

<hasChild> a owl:AsymmetricProperty .
```
inverse property

Two relations that exist simultaneously and inversely (ex. parent_of / child_of)

\[ x \ R_1 \ y \iff y \ R_2 \ x \]

\[
<\text{rdf:Property rdf:ID="hasChild"}>
   <\text{owl:inverseOf rdf:resource="#hasParent"} />
<\text{/rdf:Property}>
\]

\[
<\text{hasChild} \ owl:inverseOf <\text{hasParent}> .
\]
transitive property
a property propagated from peers to peers (e.g. ancestors)

\[ x R y \& y R z \Rightarrow x R z \]

<owl:TransitiveProperty rdf:ID="hasAncestor" />

<hasAncestor> a owl:TransitiveProperty .
disjoint properties
relations that cannot exist together on the same subject and the same object

<owl:ObjectProperty rdf:about="hasSon">  
  <owl:propertyDisjointWith rdf:resource="hasDaughter"/>  
</owl:ObjectProperty>

<hasSon> owl:propertyDisjointWith <hasDaughter> .
reflexive property
a relation that links all individuals to themselves

<owl:ReflexiveProperty rdf:about="hasRelative"/>

<hasRelative> a owl:ReflexiveProperty .
irreflexive properties
relations never link resources to themselves

<owl:IrreflexiveProperty rdf:about="hasParent"/>

<hasParent> a owl:IrreflexiveProperty .
property chain
relations which combine as a path/chain imply another relation
(e.g. parent + brother = uncle)

\( x \ P \ y \ & \ y \ Q \ z \Rightarrow x \ R \ z \)

<owl:ObjectProperty rdf:ID="uncle">
  <owl:propertyChainAxiom rdf:parseType="Collection">
    <owl:ObjectProperty rdf:about="#parent"/>
    <owl:ObjectProperty rdf:about="#brother"/>
  </owl:propertyChainAxiom>
</owl:ObjectProperty>

<uncle> rdf:type owl:ObjectProperty ;
  owl:propertyChainAxiom
    ( <parent> <brother> ) .
**functional property**

a relation for which a resource can have only one value (e.g. birth date)

\[ x R y \& x R z \Rightarrow y = z \]

```
<owl:FunctionalProperty rdf:ID="birthDate" />

<birthDate> a owl:FunctionalProperty .
```
inverse functional property
a relation for which identical values imply the same subject (e.g. SSN)
\[ x \ R \ y \ & \ z \ R \ y \ \Rightarrow \ x = z \]

<owl:InverseFunctionalProperty
    rdf:ID="socialSecurityNumber" />

socialSecurityNumber a owl:InverseFunctionalProperty .
identification by keys

Two resources with the same key values are the same.

\[ x c_1 v_1 ; c_2 v_2 \land y c_1 v_1 ; c_2 v_2 \implies x = y \]

```xml
<owl:Class rdf:ID="Person">
  <owl:hasKey rdf:parseType="Collection">
    <owl:ObjectProperty rdf:about="#name"/>
    <owl:ObjectProperty rdf:about="#firstname"/>
    <owl:ObjectProperty rdf:about="#birthdate"/>
    <owl:ObjectProperty rdf:about="#birthplace"/>
  </owl:hasKey>
</owl:Class>

<Person> owl:hasKey ( <name> <firstname> <birthdate> <birthplace> ) .
```
What can we deduce?

ex:hasSpouse a owl:SymmetricProperty .
ex:hasChild owl:inverseOf ex:hasParent .
ex:hasParent rdfs:subPropertyOf ex:hasAncestor .
ex:hasAncestor a owl:TransitiveProperty .
ex:Jim ex:hasChild ex:Jane .
ex:Jane ex:hasSpouse ex:John .
ex:Jim ex:hasParent ex:James .
equivalent classes
two classes containing exactly the same resources.

ex:Human owl:equivalentClass foaf:Person .

mit:Student owl:equivalentClass keio:Gakusei .
equivalent properties
two properties representing exactly the same relation.

ex:name owl:equivalentProperty my:label
identical resources
two URIs identifying exactly the same thing.

ex:Bill owl:sameAs ex:William
transitivity and symmetry of equivalences

ex:Bill owl:sameAs ex:William
ex:Bill owl:sameAs ex:Willy
ex:Willy owl:sameAs ex:Will

⇒ ????

⇒ ????
different resources

two URI for which we know they represent different things.

ex:Good  owl:differentFrom  ex:Evil
What can we deduce?

ex:Human owl:equivalentClass foaf:Person .
foaf:name owl:equivalentProperty ex:name .
ex:JimmyPage a ex:Human ;
    owl:sameAs ex:JamesPatrickPage .
ex:JimmyHendrix owl:differentFrom ex:JimmyPage .

quizz
What are we defining?

ex:UnluckyPerson owl:equivalentClass [ a owl:Class ; owl:intersectionOf ( ex:Person [ a owl:Class ; owl:complementOf ex:Lucky ] ) ] .
restriction on all values

<owl:Class rdf:ID="Herbivore">
  <subClassOf rdf:resource="#Animal"/>
  <subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#eats" />
      <owl:allValuesFrom rdf:resource="#Plant" />
    </owl:Restriction>
  </subClassOf>
</owl:Class>
restriction on some values

<owl:Class rdf:ID="Sportive">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hobby" />
      <owl:someValuesFrom rdf:resource="#Sport" />
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
restriction to an exact value

<owl:Class rdf:ID="Bike">
  <subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#nbWheels" />
      <owl:hasValue>2</owl:hasValue>
    </owl:Restriction>
  </subClassOf>
</owl:Class>
**self restriction**

Classes where instances have themselves as value of a property

```plaintext
ex:NarcisticPerson rdfs:subClassOf
[ a owl:Restriction ;
  owl:onProperty ex:love ;
  owl:hasSelf true ]
```
restriction on cardinality

how many times a property is used for a same subject but with different values

• Constraints: minimum, maximum, exact number

• Exemple

```
<owl:Class rdf:ID="Person">
  <subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#name" />
      <owl:maxCardinality>1</owl:maxCardinality>
    </owl:Restriction>
  </subClassOf>
</owl:Class>
```
**qualified cardinality restriction**

constraint on the number of time a property may be used with values of a given type with the same subject: minimum, maximum, nombre exact

```xml
<owl:Class rdf:ID="Human">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasParent" />
      <owl:onClass rdf:resource="#Male" />
      <owl:qualifiedCardinality>1</owl:qualifiedCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```
What can we deduce?

ex:Human rdfs:subClassOf
    [ a owl:Restriction ;
      owl:onProperty ex:hasParent ;
      owl:allValuesFrom ex:Human ] .

ex:John a ex:Human .
ex:John ex:hasParent ex:James, ex:Jane .
What are we defining and inferring?

@prefix ex: <http://example.org/>
ex:PersonList rdfs:subClassOf
[  a owl:Restriction ;
    owl:onProperty rdf:first ;
    owl:allValuesFrom ex:Person
] ,
  [  a owl:Restriction ;
    owl:onProperty rdf:rest ;
    owl:allValuesFrom ex:PersonList
] .

ex:value rdfs:range ex:PersonList .
ex:abc ex:value (ex:a ex:b ex:c) .
quizz

What are we defining and inferring?

@prefix ex: <http://example.org/>

ex:Human rdfs:subClassOf [ 
    owl:intersectionOf ( 
        [ 
            a owl:Restriction ; 
            owl:onProperty ex:hasFather ; 
            owl:maxCardinality 1
        ] , [ 
            a owl:Restriction ; 
            owl:onProperty ex:hasMother ; 
            owl:maxCardinality 1
        ] )
    ] .

ex:John a ex:Human ; ex:hasFather ex:James , ex:Jimmy .
quizz

What are we defining and inferring?

@prefix ex: <http://example.org/>

ex:Wealthy a owl:Class ;
    owl:equivalentClass [ a owl:Class ; owl:intersectionOf ( [ a owl:Restriction ;
        owl:onProperty ex:hasChild ;
        owl:allValuesFrom ex:Wealthy ] ,
        [ a owl:Restriction ;
        owl:onProperty ex:hasChild ;

ex:John a ex:Wealthy ; ex:hasChild ex:Jim .
document the schemas

• an ontology is a Resource
• an ontology has a URI
• OWL provides primitives to describe that ontology resource

description of the ontology

versions of classes and properties
owl:DeprecatedClass, owl:DeprecatedProperty
describe an ontology
one class (owl:Ontology) and several properties (owl:imports, owl:versionInfo, owl:priorVersion, owl:backwardCompatibleWith, owl:incompatibleWith)

```xml
xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:owl ="http://www.w3.org/2002/07/owl1#">
  <owl:Ontology rdf:about="http://inria.fr/2005/humans/">
    <rdfs:comment>An example OWL ontology</rdfs:comment>
    <owl:imports rdf:resource="http://cnrs.fr/animals/>
    <rdfs:label>Bio Ontology</rdfs:label>
  </owl:Ontology>
</rdf:RDF>
```
changes in classes or properties indicate a class or property is obsolete

```xml
    xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl = "http://www.w3.org/2002/07/owl#">

    <owl:DeprecatedClass rdf:ID="mammals"/>
    <owl:DeprecatedProperty rdf:ID="age"/>

</rdf:RDF>
```
OWL profiles

• Each profile is a sub-set of the OWL primitives.
• Choosing a profile is choosing a level of expressivity.
• The higher the expressivity the more complex the inferences.
• The more complex the expressivity, the longer it takes to compute the results.
OWL 1 profiles

**Lite**: essentially for lightweight hierarchies.

**DL**: more complex ontologies but complete reasoning.

**Full**: maximum expressivity but incomplete reasoning.
OWL 2 profiles

**EL**: large numbers of properties and/or classes and polynomial time.

**QL**: large volumes of instance data, and conjunctive query answering using conventional relational database in LOGSPACE

**RL**: scalable reasoning without sacrificing too much expressive power using rule-based reasoning in polynomial time

**DL**: the most expressive with complete reasoning
Protégé
Web Protégé
FAMOUS SCHEMAS
- Dublin core
- Creative Commons
- FOAF

...
SKOS knowledge

thesauri, classifications, subjects, taxonomies, folksonomies, ...
... controlled vocabulary
**SKOS schema**: 4 classes and 28 properties in OWL

<table>
<thead>
<tr>
<th>SKOS:Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URI</strong>: <a href="http://www.w3.org/2004/02/skos/core#Concept">http://www.w3.org/2004/02/skos/core#Concept</a></td>
</tr>
<tr>
<td><strong>Definition</strong>: <a href="http://www.w3.org/2004/02/skos/core#Concept">Section 3. The skos:Concept Class</a></td>
</tr>
<tr>
<td><strong>Label</strong>: Concept</td>
</tr>
<tr>
<td><strong>Disjoint classes</strong>: skos:Collection, skos:ConceptScheme</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SKOS:ConceptScheme</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>URI</strong>: <a href="http://www.w3.org/2004/02/skos/core#ConceptScheme">http://www.w3.org/2004/02/skos/core#ConceptScheme</a></td>
</tr>
<tr>
<td><strong>Definition</strong>: <a href="http://www.w3.org/2004/02/skos/core#ConceptScheme">Section 4. Concept Schemes</a></td>
</tr>
<tr>
<td><strong>Label</strong>: Concept Scheme</td>
</tr>
<tr>
<td><strong>Disjoint classes</strong>: skos:Collection, skos:Concept</td>
</tr>
</tbody>
</table>

http://www.w3.org/2004/02/skos/core
natural language expressions to refer to concepts

inria:CorporateSemanticWeb
  skos:prefLabel "corporate semantic web"@en;
  skos:prefLabel "web sémantique d'entreprise"@fr;
  skos:altLabel "corporate SW"@en;
  skos:altLabel "CSW"@en;
  skos:hiddenLabel "web semantique d'entreprise"@fr.
between concepts

inria:CorporateSemanticWeb
  skos:broader w3c:SemanticWeb;
  skos:narrower inria:CorporateSemanticWiki;
  skos:related inria:KnowledgeManagement.
inria:CorporateSemanticWeb

skos:scopeNote "only within KM community";

skos:definition "a semantic web on an intranet";

skos:example "Nokia's internal use of RDF gateway";

skos:historyNote "semantic intranet until 2006";

skos:editorialNote "keep wikipedia def. uptodate";

skos:changeNote "acronym added by fabien"."
CC (Creative Commons)
a very popular schema to describe rights associated to a resource
6+12 classes and 11 properties to:
• describe the rights associated with a resources
• describe a license and associated to a resource
• state the rights, conditions and prohibitions to use a resource

@prefix cc: <http://creativecommons.org/ns#>

CC REL : Creative Commons Rights Expression Language

http://creativecommons.org/ns
FOAF: Friend of a Friend

a very popular schema to describe persons and social networks

13 classes and 62 properties to describe:
• user profiles (your RDF homepage)
• social networks (persons you know)
• social activities (accounts, actions)

@prefix foaf: <http://xmlns.com/foaf/0.1/>

FOAF = Core + Social Web + Linked Data Utilities

http://xmlns.com/foaf/spec/
VoID: describing RDF datasets/linksets
DCAT: describe any dataset (not just RDF)
Data Cube: publish multi-dimensional data (statistics)
Provenance: PROV-DM & PROV-O
describe entities and activities
involved in providing a resource
e.g. a chart produced from two sources of data

ex:illustrate prov:used ex:composition .
ex:chart1 prov:wasGeneratedBy ex:illustrate
e.g. a chart produced from two sources of data

```
ex:compose prov:used ex:dataSet1 ;
    prov:used ex:regionList .
ex:composition prov:wasGeneratedBy ex:compose .
ex:illustrate prov:used ex:composition .
ex:chart1 prov:wasGeneratedBy ex:illustrate
```
PROV primer full example: more data
semantic waste separation
the web is a garbage can,
the semantic web will be a semantic garbage can.
EXTENDING TO OTHER SOURCES
toward all forms of data on the web

W3C DATA ACTIVITY Building the Web of Data

More and more Web applications provide a means of accessing data. From simple visualizations to sophisticated interactive tools, there is a growing reliance on the availability of data which can be “big” or “small”, of diverse origin, and in different formats; it is usually published without prior coordination with other publishers — let alone with precise modeling or common vocabularies. The Data Activity recognizes and works to overcome this diversity to facilitate potentially Web-scale data integration and processing. It does this by providing standard data exchange formats, models, tools, and guidance.

The overall vision of the Data Activity is that people and organizations should be able to share data as far as possible using their existing tools and working practices but in a way that enables others to derive and add value, and to utilize it in ways that suit them. Achieving that requires a focus not just on the interoperability of data but of communities.

Questions? Contact Phil Archer <phila@w3.org>, W3C Data Activity Lead.

Context & Vision

The Data Activity merges and builds upon the eGovernment and Semantic Web Activities. The eGovernment Activity comprised an interest group that offered members a series of interesting talks from well placed speakers in governments around the world, including from countries that are often under-represented at W3C such as Jordan and Uganda. Primary topics have been the use of social media for citizen engagement and open data. The Semantic Web Activity was launched in 2001 to lead the use of the Web as an exchange medium for data as well as documents. That overall aim, along with a series of associated activities by W3C and others, has been highly successful — although not necessarily in the way originally envisioned. For example, the vision was that organizations and individuals would publish data in vocabularies, which the user community sees as critical companions to Web standards such as XML, RDF and HTML.

The use of the Web as a platform for delivering data has been driven by policy as much as by technology. The Open Data Charter being a prime example. Other examples include President Obama’s Executive Order and the European Union’s revised PSI Directive. These policies apply equally to the areas of government information, scientific research, and cultural heritage and that creates a further source of diversity of workflow, people and the technologies they use.

The W3C Data Activity will support technologists tasked with responding to this political pressure. It will do so in a way that works for those individuals and at the same time delivers maximum return on the political and financial
R2RML
a standard transformation of a relational database in RDF
two types of transformations

• Default transformation
  [A Direct Mapping of Relational Data to RDF]

• Customized transformation
  [R2RML: RDB to RDF Mapping Language]
many data

buried and dormant in web pages
**RDFa means RDF in HTML attributes**

```html
<body vocab="http://purl.org/dc/terms/>

<div resource="http://lib.com/books/0684853949">

<h2 property="title">The Man Who Mistook His Wife For a Hat</h2>
<h3 property="creator">Oliver Sacks</h3>

...
Le Web Sémantique

Date: 2012-05-01

Fabien

Fabien Gandon, phone: +33492965170, mail: fgandon@inria.fr
HTML+RDFa content (in browser)

(...)  
<body vocab="http://purl.org/dc/terms/">
  <div resource="/books/web_semantique">
    <h2 property="title">Le Web Sémantique</h2>
    <p>Date: <span property="created">2012-05-01</span></p>
    <h3 property="creator" resource="#fg">Fabien</h3>
  </div>
  <div vocab="http://xmlns.com/foaf/0.1/" resource="#fg" typeof="Person">
    <p>
      Fabien Gandon, phone: <a property="phone" href="tel:+33492965170">+33492965170</a>
      mail: <a property="mbox" href="mailto:fgandon@inria.fr">fgandon@inria.fr</a>
    </p>
  </div>

Le Web Sémantique

Date: 2012-05-01

Fabien

Fabien Gandon, phone: +33492965170 mail: fgandon@inria.fr
HTML+RDFa content (read by RDFa parser)

(...) 
<body vocab="http://purl.org/dc/terms/">
  <div resource="/books/web_semantique">
    <h2 property="title">Le Web Sémantique</h2>
    <p>Date: <span property="created">2012-05-01</span></p>
    <h3 property="creator" resource="#fg">Fabien</h3>
  </div>
  <div vocab="http://xmlns.com/foaf/0.1/" resource="#fg" typeof="Person">
    <p>
      <span property="name">Fabien Gandon</span>,
      phone: <a property="phone" href="tel:+33492965170">+33492965170</a>,
      mail: <a property="mbox" href="mailto:fgandon@inria.fr">fgandon@inria.fr</a>
    </p>
  </div>
</body>
Do it...

• Look at the Web Page
  http://schema.openspring.net/person/dries-buytaert

• Call the translator on this Web page to get Turtle:
  http://rdf-translator.appspot.com/

• What are the types of the main resource extracted?

• Do the same with:
  http://schema.openspring.net/event/2014-winter-olympics
  http://schema.openspring.net/recipe/apple-pie
  http://schema.openspring.net/events/drupalcamps
Do it...

Use the online tool to play with RDFa adding for instance a “creator” property

https://rdfa.info/play/
<div about="" typeof="cc:Work"
   xmlns:cc="http://creativecommons.org/ns#"
   xmlns:dc="http://purl.org/dc/elements/1.1/" align="center">
   <img alt="Creative Commons License"
        src="http://i.creativecommons.org/l/by/3.0/us/88x31.png" />
   <br />
   <span property="dc:title">The Lessig Blog</span>, a
   collection of texts
   <a property="cc:attributionName" rel="cc:attributionURL"
      href="http://lessig.org/">Lawrence Lessig</a>,
   is licensed under a
   <a rel="license"
      href="http://creativecommons.org/licenses/by/3.0/">
   Creative Commons Attribution License</a>.<br />
   There are <a rel="cc:morePermissions"
      href="http://lessig.org/blog/other-license">
   alternative licensing options</a>. </div>

CC REL in RDFa
schema.org = Bing + Google + Yahoo! + Yandex

schemas to improve index, search and display e.g:

• Creative works, Book, Movie, MusicRecording, Recipe, TVSeries ...
• Embedded non-text objects, AudioObject, ImageObject, VideoObject
• Event
• Organization
• Person
• Place, LocalBusiness, Restaurant ...
• Product, Offer, AggregateOffer
• Review, AggregateRating
OGP code

<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="en-US"
xmlns:fb="https://www.facebook.com/2008/fbml">
<head prefix="og: http://ogp.me/ns# fb: http://ogp.me/ns# YOUR_NAMESPACE:
http://ogp.me/ns/apps/YOUR_NAMESPACE#">
<meta property="fb:app_id" content="YOUR_APP_ID" />
<meta property="og:type" content="YOUR_NAMESPACE:recipe" />
<meta property="og:title" content="Stuffed Cookies" />
<meta property="og:image" content="http://example.com/cookie.jpg" />
<meta property="og:description" content="The Turducken of Cookies" />
<meta property="og:url" content="http://example.com/cookie.html">
</head>
<body>
<script type="text/javascript">
function postCook()
{
   FB.api('/me/YOUR_NAMESPACE:cook' + '?recipe=http://example.com/cookie.html','post', (...))
}
</script>
(…)
<form>
   <input type="button" value="Cook" onclick="postCook()" />
</form>
</body>
</html>
These data are accessible to everyone!
If you apply a parser to these pages, you will get their data...
Test online

- IMDB uses RDFa – OGP for the I like button
- Choose a movie on IMDB [http://www.imdb.com](http://www.imdb.com)
- Copy the URL of the page of the movie
- Go to the RDFa 1.0 RDFa Distiller and Parser: [https://www.w3.org/2007/08/pyRdfa/](https://www.w3.org/2007/08/pyRdfa/)
- Open the URI option, past the URL of the movie page and configure and perform the extraction to get Turtle
- Try also the transformation on the translator: [http://rdf-translator.appspot.com/](http://rdf-translator.appspot.com/)
Call the translator on...

- A product on eBay
- A movie in Dailymotion
- An article on LeMonde.fr
- A recipe on Marmiton.org
- A hotel on Booking.com

...
page && data
Open as possible means public domain plus a strong community

My journey implementing and/or iterating/improving/creating "open" standards began almost 10 years at Microsoft when I was assigned the area of CSS support in Internet Explorer for Macintosh. Along the way I've learned a lot about the longterm value of open standards, open source, and open content, and as a result the plethora of "open" licenses out there. Having seen real difficulties that different "open" projects have had working together due to license (or even philosophical definition of "freedom") incompatibilities, limitations, friction, barriers to developing derivative materials to help "open" projects, and even FUD used inside many corporations to limit use of "open" resources, it led me inexorably to one conclusion.
Linked Data in JSON

- JSON (JavaScript Object Notation)
  - hierarchy of name-value pairs
- JSON-LD (JSON for Linked Data)
  - designed around the notion of "context" to provide additional mappings from JSON to an RDF model.
  - a context can be embedded directly in a JSON-LD document or put into a separate file and referenced.
  - specific reserved names prefixed by @
    e.g. @context, @type
{
"@context": {
"@vocab": "http://schema.org/",
"@base" : "http://data.org/",
"id" : "@id",
"firstName": "givenName",
"lastName": "familyName",
"headline": { "@id": "jobTitle", "@language": "en" },
"siteStandardProfileRequest" : null },
"firstName": "Fabien",
"headline": "Research Director at Inria",
"id": "Fg-fjekzI",
"lastName": "Gandon",
"siteStandardProfileRequest": {
"url": "https://www.linkedin.com/profile/view?id=AAAAAA"
}
}

e.g. LinkedIn JSON


Mapping with @vocab e.g. LinkedIn JSON
```json
{
  "@context": {
    "@vocab": "http://schema.org/",
    "@base": "http://data.org/",
    "id": "@id",
    "firstName": "givenName",
    "lastName": "familyName",
    "headline": { "@id": "jobTitle", "@language": "en" },
    "siteStandardProfileRequest": null
  },
  "firstName": "Fabien",
  "headline": "Research Director at Inria",
  "id": "Fg-fjekzI",
  "lastName": "Gandon",
  "siteStandardProfileRequest": {
    "url": "https://www.linkedin.com/profile/view?id=AAAAAA"
  }
}

@prefix : <http://schema.org/> .

<http://data.org/Fg-fjekzI> :familyName "Gandon" ;
  :givenName "Fabien" ;
  :jobTitle "Research Director at Inria"@en .

Mapping with @vocab e.g. LinkedIn JSON
Test online

• Transform your FOAF profile in JSON-LD with the translator: http://rdf-translator.appspot.com/

• Use the following online tool to generate different variations of JSON-LD of your profile (expanded, collapsed, flattened, etc.) http://json-ld.org/playground/
**CSV-LD & Linked CSV**

- contexts to interpret and generate CSV
- conventions for CSV to be linked in RDF
<table>
<thead>
<tr>
<th>country</th>
<th>country group</th>
<th>name (en)</th>
<th>name (fr)</th>
<th>name (de)</th>
<th>latitude</th>
<th>longitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>at</td>
<td>eu</td>
<td>Austria</td>
<td>Autriche</td>
<td>Österreich</td>
<td>47.6965545</td>
<td>13.34598005</td>
</tr>
<tr>
<td>be</td>
<td>eu</td>
<td>Belgium</td>
<td>Belgique</td>
<td>Belgien</td>
<td>50.501045</td>
<td>4.47667405</td>
</tr>
<tr>
<td>bg</td>
<td>eu</td>
<td>Bulgaria</td>
<td>Bulgarie</td>
<td>Bulgrien</td>
<td>42.72567375</td>
<td>25.4823218</td>
</tr>
<tr>
<td>Country</td>
<td>Country Group</td>
<td>Name (en)</td>
<td>Name (fr)</td>
<td>Name (de)</td>
<td>Latitude</td>
<td>Longitude</td>
</tr>
<tr>
<td>---------</td>
<td>---------------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>Austria</td>
<td>eu</td>
<td>Austria</td>
<td>Autriche</td>
<td>Österreich</td>
<td>47.6965545</td>
<td>13.34598005</td>
</tr>
<tr>
<td>Belgium</td>
<td>eu</td>
<td>Belgium</td>
<td>Belgique</td>
<td>Belgien</td>
<td>50.501045</td>
<td>4.47667405</td>
</tr>
<tr>
<td>Bulgaria</td>
<td>eu</td>
<td>Bulgaria</td>
<td>Bulgarie</td>
<td>Bulgarien</td>
<td>42.72567375</td>
<td>25.4823218</td>
</tr>
</tbody>
</table>

Based on JSON contexts
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix schema: <http://schema.org/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://example.org/country/at> a schema:Country;
    schema:geo <http://example.org/country/at#geo>;
    schema:name "Austria"@en, "Autriche"@fr, "Österreich"@de .

<http://example.org/country/be> a schema:Country;
    schema:geo <http://example.org/country/be#geo>;
    schema:name "Belgium"@en, "Belgique"@fr, "Belgien"@de .

<http://example.org/country/bg> a schema:Country;
    schema:geo <http://example.org/country/bg#geo>;
    schema:name "Bulgaria"@en, "Bulgarie"@fr, "Bulgarien"@de .

<http://example.org/country/at#geo> a schema:GeoCoordinates;
    schema:latitude 4.76965545e1;
    schema:longitude 1.334598005e1 .

<http://example.org/country/be#geo> a schema:GeoCoordinates;
    schema:latitude 5.0501045e1;
    schema:longitude 4.47667405e0 .

<http://example.org/country/bg#geo> a schema:GeoCoordinates;
    schema:latitude 4.272567375e1;
    schema:longitude 2.54823218e1 .
doggy-bag
impossible to predict every usage
avoid building
black boxes
make conceptualizations explicit
open your data to those who could use them
66 FOAF primitives

3,475,908,348 references

\[ \times 52 \text{ millions} \]

“a small tree ruling a big graph”

(1) Franck Van Harmelen, ISWC 2011
(2) Libby Miller, 2009
“semantic web” and not “semantic web”

“a lightweight ontology allows us to do lightweight reasoning”
identify

describe & link

query

reasoning

trace

GOALS AND MEANS

URI

RDF

HTTP, SPARQL, LDP

RDFS & OWL

PROV-O
identify

describe & link

query

reasoning

trace

GOALS AND MEANS

http://fabien.fr#me

#me type man

select * {?r type ?t}

man subClassOf male

wasAttributedTo #me
linked data

linked enterprise data

enterprise data

personal data

open data

linked open data

linked data

linked healthcare data

open data

linked open data

linked data

web of data

linked data schemas

web of sensors, things, ...

big data streams
data bases
data mining
data structures
data models
data type

data schemas
data schemas

data schemas

big linked data

big data

data streams

data streams

data streams

data streams

data

data

data

closed data

closed data

closed data

Vicinity

Visibility

Velocity

Volume

Variety

Web of sensors, things, ...

Linked data streams

Linked enterprise data

Enterprise data

Personal data

Open data

Linked open data

Linked data

Linked healthcare data

Open data

Linked open data

Linked data

Web of data

Linked data schemas

Semantic web of data
The Quark and the Jaguar: Adventures in the Simple and the Complex [Paperback]
Murray Gell-Mann (Author)

Price: $13.59
In Stock.
Ships from and sold by Amazon.com.

<table>
<thead>
<tr>
<th>Format</th>
<th>Amazon Price</th>
<th>New from</th>
<th>Used from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardcover</td>
<td>--</td>
<td>$10.97</td>
<td>$0.65</td>
</tr>
<tr>
<td>Paperback</td>
<td>$13.59</td>
<td>$10.01</td>
<td>$7.84</td>
</tr>
</tbody>
</table>

Show 4 more formats

Product Details
- Paperback: 392 pages
- Publisher: St. Martin's Griffin (September 15, 1995)
- Language: English
- ISBN-10: 0385072535
- Product Dimensions: 9.1 x 6.1 x 1.2 inches
- Shipping Weight: 1.2 pounds (view shipping rates and policies)
The Quark and the Jaguar: Adventures in the Simple and the Complex [Paperback]

Murray Gell-Mann (Author)

Price: $13.59

In Stock.
Ships from and sold by Amazon.com.

<table>
<thead>
<tr>
<th>Format</th>
<th>Amazon Price</th>
<th>New from</th>
<th>Used from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardcover</td>
<td>--</td>
<td>$13.67</td>
<td>$0.65</td>
</tr>
<tr>
<td>Paperback</td>
<td>$13.59</td>
<td>$10.01</td>
<td>$7.84</td>
</tr>
</tbody>
</table>

Show 4 more formats

Product Details

Paperback: 392 pages
Publisher: St. Martin's Griffin (September 15, 1995)
Language: English
ISBN-10: 0385072535
Product Dimensions: 9.1 x 6.1 x 1.2 inches
Shipping Weight: 1.2 pounds (View shipping rates and policies)
Average Customer Review: ★★★★★ (30 customer reviews)
Amazon Bestsellers Rank: #373,365 in Books

Customer Reviews

30 Reviews

2 star: (3)
4 star: (8)
3 star: (7)
2 star: (4)
1 star: (2)

Average Customer Review: ★★★★★ (30 customer reviews)
one web...
he who controls metadata, controls the web and through the world-wide web many things in our world.

fabien, gandon, @fabien_gandon, http://fabien.info