

# Les Langages du Web Sémantique

Catherine Faron Zucker  
faron@polytech.unice.fr

# The Semantic Web in a Nutshell

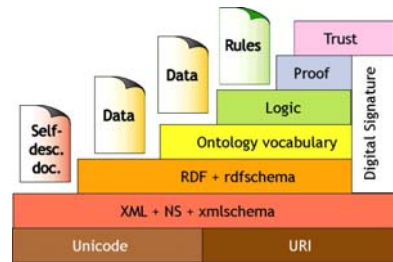
- World Wide Web Consortium : [www.w3.org](http://www.w3.org)
- « The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation »

[www.scientificamerican.com/2001/0501issue/0501berners-lee.html](http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html)

# The Semantic Web in a Nutshell

- Nomage : URI
- Syntaxe : XML
- Sémantique: RDF, RDFS, OWL, SPARQL, SWRL...

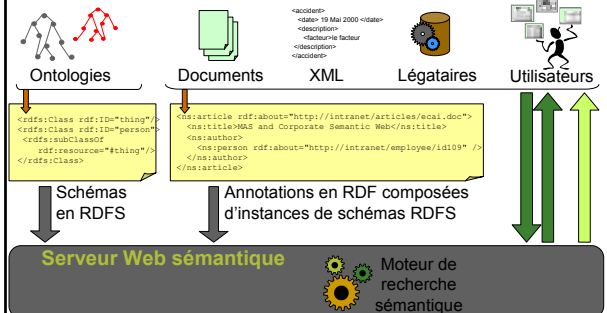
# W3C Semantic Web Stack



# Applications du Semantic Web

- Retrouver une séquence vidéo avec un facteur cycliste (annotation de documents multimedia)
  - Veille technologique: prévenir tels utilisateurs de telles conf, publi, etc. selon profils et thèmes
  - Mémoire d'entreprise structurée par un référentiel
  - Mémoire organisationnelle, plus largement
- > Recherche d'information "intelligente" sur le web

# Recherche sémantique sur le web (1/3)



## Recherche sémantique sur le web (2/3)

```
<author id='d:a'>
  <name> David </name>
</author>
d:a rdf:type c:Researcher

c:Researcher rdfs:subClassOf c:Person

?x rdf:type ?y
?y rdfs:subClassOf ?z

?x rdf:type ?z

d:a rdf:type c:Person

SELECT ?x WHERE { ?x rdf:type c:Person }
```

**RDF**

**RDFS**

**règles**

**Inférences élémentaires**

**SPARQL**

## Resource Description Framework : RDF

<http://www.w3.org/RDF>

## Resource Description Framework

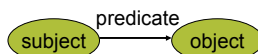
- Langage de représentation de connaissances
    - sur les ressources du Web
    - sur des entités identifiées sur le Web (par une uri)
  - Standard dédié
    - à l'échange des informations sur le web
    - au traitement des informations sur le web
- Modèle + Format d'échange RDF/XML

## RDF Design Goal

- Modèle simple
- Sémantique formelle
- Vocabulaire extensible basé sur les URI
- Syntaxe XML
- XML schema datatypes
- Autorise quiconque à faire des déclarations sur n'importe quelle ressource: modularité

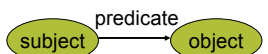
<http://www.w3.org/TR/rdf-concepts/>

## Modèle simple (1/4)



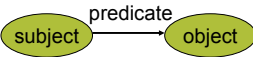
- Décrire les entités avec:
  - des propriétés
  - des valeurs de propriétés
- Modèle de triplets
  - ressource propriété valeur
- Modèle de graphe
  - Jointure (conjonction) de triplets

## Modèle simple (2/4)



- Web-based design: Propriétés vs Classes
- Pourquoi les propriétés?  
Car le web est un *réseau* où tout le monde peut participer
- Une classe n'est pas définie entièrement localement: elle peut être réutilisée, complétée ailleurs par de nouvelles propriétés

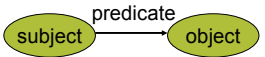
## Modèle simple (3/4)



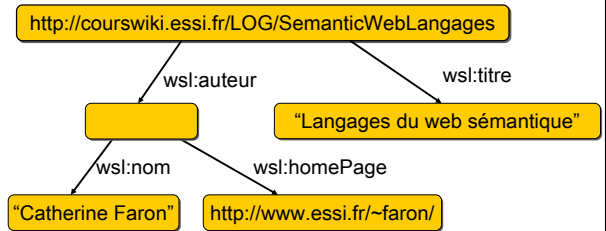
- Modèle de triplets
  - ressource propriété valeur
- Les valeurs sont :
  - des ressources
  - des littéraux (valeurs atomiques)

```
<http://courswiki.essi.fr/LOG/SemanticWebLangages> titre 'Langages ...'  
<http://courswiki.essi.fr/LOG/SemanticWebLangages> auteur _:x  
_:x nom 'Catherine Faron'  
_:x homePage <http://www.essi.fr/~faron/>
```

## Modèle simple (4/4)



- Modèle de graphe



## Vocabulaire basé sur les URI (1/2)

- Uniform Resource Identifier  
une URI identifie toute ressource physique (accessible sur le web ou non) ou abstraite
  - http://www.essi.fr
  - http://www.w3.org/TR/rdf-concepts/#section-design-goals
  - ftp://ftp-sop.inria.fr
  - mailto:faron@essi.fr
- Un noeud d'un graphe RDF est soit une URI, soit un littéral soit un blank node
- Une propriété est une URI

## Vocabulaire basé sur les URI (2/2)

- Resource:
  - http://courswiki.essi.fr/LOG/SemanticWebLangages
- Property:
  - http://courswiki.essi.fr/LOG/SemanticWebLangages/schema#titre
- Value:
  - 'Langages du web sémantique'

## Syntaxe N-Triples

```
<http://courswiki.essi.fr/LOG/SemanticWebLangages> titre 'Langages ...'  
<http://courswiki.essi.fr/LOG/SemanticWebLangages> auteur _:x  
_:x nom 'Catherine Faron'  
_:x homePage <http://www.essi.fr/~faron/>
```

## Syntaxe XML (1/3)

```
<rdf:Description  
  rdf:about="http://courswiki.essi.fr/LOG/SemanticWebLangages"  
  xmlns:swl="http://courswiki.essi.fr/LOG/SemanticWebLangages/schema#"  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">  
  <swl:titre>Langages du Web Sémantique</swl:titre>  
  <swl:auteur>  
    <rdf:Description>  
      <swl:nom>Catherine Faron</swl:nom>  
      <swl:homePage>'http://www.essi.fr/~faron/'</swl:homePage>  
    </rdf:Description>  
  </swl:auteur>  
</rdf:Description>
```

<http://www.w3.org/TR/rdf-syntax-grammar/>

## Syntaxe XML (2/3) : Namespaces

- A un schéma (vocabulaire) est associé un nom symbolique: un URI appelé namespace
- Les termes issus du schéma sont préfixés par cet URI
- <http://www.w3.org/TR/REC-xml-names/>

```
xmlns:swl="http://courswiki.essi.fr/PMLT/SemanticWebLangages/schema#"
```

swl:titre désigne

```
'http://courswiki.essi.fr/PMLT/SemanticWebLangages/schema#titre'
```

## Syntaxe(s) XML (3/3)

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <ex:editor>
    <rdf:Description>
      <ex:homePage>
        <rdf:Description rdf:about="http://purl.org/net/dajobe/">
          </rdf:Description>
        </ex:homePage>
        <ex:fullName>Dave Beckett</ex:fullName>
      </rdf:Description>
    </ex:editor>
    <dc:title>RDF/XML Syntax Specification (Revised)</dc:title>
  </rdf:Description>
```

## Syntaxe(s) XML (3/3)

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <ex:editor>
    <rdf:Description>
      <ex:homePage rdf:resource="http://purl.org/net/dajobe/">
      <ex:fullName>Dave Beckett</ex:fullName>
    </rdf:Description>
  </ex:editor>
  <dc:title>RDF/XML Syntax Specification (Revised)</dc:title>
</rdf:Description>
```

## Syntaxe(s) XML (3/3)

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <ex:editor>
    <rdf:Description>
      <ex:homePage rdf:resource="http://purl.org/net/dajobe/">
      <ex:fullName>Dave Beckett</ex:fullName>
    </rdf:Description>
  </ex:editor>
  <dc:title>RDF/XML Syntax Specification (Revised)</dc:title>
</rdf:Description>
```

## Syntaxe(s) XML (3/3)

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar"
  dc:title="RDF/XML Syntax Specification (Revised)">
  <ex:editor>
    <rdf:Description ex:fullName="Dave Beckett" >
      <ex:homePage rdf:resource="http://purl.org/net/dajobe/">
    </rdf:Description>
  </ex:editor>
</rdf:Description>
```

## Exercice

## Typage de ressource (1/3)

```
<http://courswiki/...Langages> rdf:type <http://www.essi.fr/schema#Cours>
```

```
_:x rdf:type <http://www.essi.fr/schema#Personne>
```

- La référence à **Cours** ou **Personne** en tant que types suffit à leur donner une existence
- Une ressource peut avoir plusieurs types

## Typage de ressource (2/3)

### ■ Syntaxe XML

```
<rdf:Description  
  rdf:about="http://courswiki.essi.fr/LOG/SemanticWebLangages">  
  <rdf:type rdf:resource="http://www.essi.fr/schema#Cours" />  
  ...  
</rdf:Description>
```

```
<rdf:Description>  
  <rdf:type rdf:resource="http://www.essi.fr/schema#Personne" />  
  ...  
</rdf:Description>
```

## Typage de ressource (3/3)

### ■ Syntaxe(s) XML

```
<Cours  
  rdf:about="http://courswiki.essi.fr/LOG/SemanticWebLangages">  
  <swl:titre>Langages du Web sémantique</swl:titre>  
  <swl:auteur>  
    <swl:Personne>  
      <swl:nom>Catherine Faron</swl:nom>  
    </swl:Personne>  
  </swl:auteur>  
</swl:Cours>
```

## Ressources anonymes (1/5)

- Ressource intermédiaire non identifiée (sans URI)
- Sémantique existentielle : il existe une ressource

```
<swl:Cours>  
  <swl:titre>Langages du web sémantique</swl:titre>  
  <swl:auteur>  
    <swl:Personne>  
      <swl:nom>Catherine Faron</swl:nom>  
    </swl:Personne>  
  </swl:auteur>  
</swl:Cours>
```

## Ressources anonymes (2/5)

- Il faut pouvoir différencier les différents blank nodes d'un graphe dans une représentation par triplets ou dans la syntaxe XML
- Blank node identifiers
  - Modèle de triplets: `_:name`
  - Syntaxe XML: `rdf:nodeID = "name"`

## Ressources anonymes (3/5)

### ■ Syntaxe(s) XML

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar"  
  dc:title="RDF/XML Syntax Specification (Revised)">  
  <ex:editor>  
    <rdf:Description ex:fullName="Dave Beckett" >  
      <ex:homePage rdf:resource="http://purl.org/net/dajobe/">  
    </rdf:Description>  
  </ex:editor>  
</rdf:Description>
```

## Ressources anonymes (4/5)

- Syntaxe(s) XML : identification des blank nodes

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar"
  dc:title="RDF/XML Syntax Specification (Revised)">
  <ex:editor rdf:nodeID="abc"/>
</rdf:Description>

<rdf:Description rdf:nodeID="abc" ex:fullName="Dave Beckett">
  <ex:homePage rdf:resource="http://purl.org/net/dajobe/" />
</rdf:Description>
```

## Ressources anonymes (5/5)

- Syntaxe(s) XML : omission des blank nodes

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar"
  dc:title="RDF/XML Syntax Specification (Revised)">
  <ex:editor rdf:parseType="Resource" >
    <ex:fullName>"Dave Beckett" </ex:fullName>
    <ex:homePage rdf:resource="http://purl.org/net/dajobe/" />
  </ex:editor>
</rdf:Description>
```

## SPARQL Protocol And RDF Query Language

<http://www.w3.org/2001/sw/DataAccess/>

## SPARQL

- SPARQL Query Language for RDF  
<http://www.w3.org/TR/rdf-sparql-query/>
- SPARQL Query Results XML Format  
<http://www.w3.org/TR/rdf-sparql-XMLres/>
- SPARQL Protocol for RDF  
<http://www.w3.org/TR/rdf-sparql-protocol/>

## SPARQL Query Language for RDF

<http://www.w3.org/2001/sw/DataAccess/>

## Introduction

- Langage de requête sur des données RDF
- Appariements de patterns de graphes
- Langage « à la SQL »

## Basic Graph Patterns (1/3)

### ■ Triple pattern

```
select ?title
where
{ <http:...book1> <http:...title> ?title }
```

### ■ Ensemble de triple patterns

```
select ?mbox
where
{
  ?x <http:...name> "Johnny Lee Outlaw" .
  ?x <http:...mbox> ?mbox
}
```

## Basic Graph Patterns (2/3)

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
select ?mbox
where
{
  ?x foaf:name "Johnny Lee Outlaw" .
  ?x foaf:mbox ?mbox
}
```

## Basic Graph Patterns (3/3)

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
select ?mbox
where
{
  ?x foaf:name "Johnny Lee Outlaw" .
  ?x foaf:mbox ?mbox
}
```

## Basic Graph Pattern Matching (1/3)

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
select ?mbox
where { ?x foaf:name "Johnny Lee Outlaw" .
        ?x foaf:mbox ?mbox }

_:a foaf:name "Johnny Lee OutLaw"
_:a foaf:mbox <mailto:jlow@example.com>
_:b foaf:name "Peter GoodGuy"
_:b foaf:mbox <mailto:peter@example.org>
```

### ■ Binding de ?mbox avec <mailto:jlow@example.com>

## Basic Graph Pattern Matching (2/3)

- Le résultat d'une requête est l'ensemble de tous les patterns solutions sur lesquels le query pattern peut être projeté
  - Une variable peut avoir plusieurs bindings
- Plusieurs variables peuvent être sélectionnées dans la requête

## Basic Graph Pattern Matching (3/3)

- Noms et adresses mail des personnes présentes dans la base d'annotation

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
select ?name ?mbox
where { ?x foaf:name ?name . ?x foaf:mbox ?mbox }

_:a foaf:name "Johnny Lee OutLaw"
_:a foaf:mbox <mailto:jlow@example.com>
_:b foaf:name "Peter GoodGuy"
_:b foaf:mbox <mailto:peter@example.org>

{{(name="Johnny Lee OutLaw" , mbox=<mailto:jlow@example.com>),
 (name="Peter GoodGuy", mbox=<mailto:peter@example.org>)}}
```

## Exercice

## SPARQL Results Document (1/3)

- Variable binding results

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head> ... </head>
  <results> ... </results>
</sparql>
```

- Boolean results

```
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head> ... </head>
  <boolean> ... </boolean>
</sparql>
```

## SPARQL Results Document (2/3)

```
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="x">
      ...
    </variable>
  </head>
  <results ordered="false" distinct="false">
    <result>
      <binding name="x">...</binding>
      ...
    </result>
    ...
  </results>
</sparql>
```

## Exercice

- SPARQL results document pour la requête précédente?

## SPARQL Results Document (3/3)

```
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="name">
    <variable name="mbox">
  </head>
  <results ordered="false" distinct="false">
    <result>
      <binding name="name">Johnny Lee OutLaw</binding>
      <binding name="mbox"><mailto:jlow@example.com></binding>
    </result>
    <result>
      <binding name="name">Peter GoodGuy</binding>
      <binding name="mbox"><mailto:peter@example.org></binding>
    </result>
  </results>
</sparql>
```

## Exercice

- Ecrire le « SPARQL Result Document » résultat de la requête suivante

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
select ?x ?name
where { ?x foaf:name ?name }
```

```
_:a foaf:name "Johnny Lee OutLaw"
_:a foaf:mbox <mailto:jlow@example.com>
_:b foaf:name "Peter GoodGuy"
_:b foaf:mbox <mailto:peter@example.org>
```



## Solution

```
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="name">
      <variable name="x">
    </head>
    <results ordered="false" distinct="false">
      <result>
        <binding name="name">Johnny Lee OutLaw</binding>
        <binding name="x">_c</binding>
      </result>
      <result>
        <binding name="name">Peter GoodGuy</binding>
        <binding name="x">_d</binding>
      </result>
    </results>
  </sparql>
```

## Raccourcis syntaxiques (1/2)

- `?x foaf:name ?name ; foaf:nick "Alice", "Alice_" .`  
`?x foaf:name ?name .`  
`?x foaf:nick "Alice" .`  
`?x foaf:nick "Alice_" .`
- `[ foaf:name ?name ] .` ou bien `[ ] foaf:name ?name .`  
`_b foaf:name ?name .`
- `[ foaf:name ?name ] foaf:nick "Alice" .`  
`_b foaf:name ?name .`  
`_b foaf:name "Alice" .`

## Raccourcis syntaxiques (1/2)

- `:x :q [ :p "v" ] .`  
`:x :q _c .`  
`_c :p "v" .`
- `[ foaf:name ?name ;`  
`foaf:mbox <mailto:peter@example.org> ] .`  
`_b foaf:name ?name .`  
`_b foaf:mbox <mailto:peter@example.org>`

## Matching RDF Literals

- RDF data  
`@prefix xsd: <http://www.w3.org/2001/XMLSchema#>`  
`x ns:p "42"^^xsd:integer .`  
`y ns:p "cat"@en .`
- SPARQL queries
  1. `select ?v where { ?v ?p 42 }`
  2. `select ?v where { ?v ?p "cat" }`
- Résultats
  1. `?v=x`
  2. pas de solution

## Value Constraints

- 1) PREFIX dc: <http://purl.org/dc/elements/1.1/>  
PREFIX ns: <http://example.org/ns#>  
SELECT ?title ?price  
WHERE { ?x ns:price ?price .  
**FILTER** (?price < 30.5) .  
?x dc:title ?title . }
- 2) PREFIX dc: <http://purl.org/dc/elements/1.1/>  
SELECT ?title  
WHERE { ?x dc:title ?title  
**FILTER** regex(?title, "SPARQL") }

## Exercice

- RDF data  
`@prefix dc: <http://purl.org/dc/elements/1.1/> .`  
`@prefix : <http://example.org/book/> .`  
`@prefix ns: <http://example.org/ns#> .`  
`:book1 dc:title "SPARQL Tutorial" .`  
`:book1 ns:price 42 .`  
`:book2 dc:title "The Semantic Web" .`  
`:book2 ns:price 23 .`
- Réponses aux requêtes précédentes?  
`?title="The SemanticWeb" ?price=23`  
`?title="SPARQL Tutorial"`

## Opérateurs de filtre

- Opérateurs unaires de SPARQL
  - Opérateurs unaires de XQuery: ! + -
  - Tests: **bound**, isURI, isBLANK, isLITERAL
  - Accesseurs: str, lang, datatype
- Opérateurs binaires de SPARQL
  - Connecteurs logiques: && ||
  - Tests XPath: != < <= > >=
  - Tests SPARQL: != entre termes RDF, regexp
  - Opérateurs arithmétiques XPath: + - \* /

## L'opérateur bound(var) (1/4)

```
_:a foaf:givenName "Alice".
_:b foaf:givenName "Bob" .
_:b dc:date "2005-04-04T04:04:04Z"^^xsd:dateTime .
```

Quels sont les noms ayant une date associée?

```
SELECT ?name
WHERE { ?x foaf:givenName ?name .
        OPTIONAL { ?x dc:date ?date } .
        FILTER ( bound(?date) ) }
```

Résultat: ?name="Bob"

## L'opérateur bound(var) (2/4)

### ■ Négation as failure (NAF)

Quels sont les noms de personnes *n'ayant pas* de date associée?

```
_:a foaf:givenName "Alice".
_:b foaf:givenName "Bob" .
_:b dc:date "2005-04-04T04:04:04Z"^^xsd:dateTime .
```

```
SELECT ?name
WHERE { ?x foaf:givenName ?name .
        OPTIONAL { ?x dc:date ?date } .
        FILTER ( ! bound(?date) ) }
```

Résultat: ?name="Alice"

## L'opérateur bound(var) (3/4)

### ■ Négation as failure (NAF)

Rechercher les noms des personnes non mariées

```
SELECT ?name
WHERE {
  ?x ns:name ?name .
  OPTIONAL { ?x ns:marriedWith ?y . }
  FILTER ( ! bound(?y) )
}
```

## L'opérateur bound(var) (4/4)

### ■ Négation as failure (NAF)

Recherche des personnes ne s'appelant pas "Alice"

```
_:a foaf:givenName "Alice".
_:a foaf:givenName "Marie".
_:b foaf:givenName "Bob" .
_:b dc:date "2005-04-04T04:04:04Z"^^xsd:dateTime .
```

```
SELECT ?x
WHERE { ?x foaf:givenName ?name .
        OPTIONAL { ?x foaf:givenName ?n
                  FILTER ( ?n = "Alice" ) } .
        FILTER ( ! bound(?n) ) }
```

## L'opérateur lang

```
_:a foaf:name "Robert"@EN.
_:a foaf:name "Roberto"@ES.
_:a foaf:mbox <mailto:bob@work.example> .
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE { ?x foaf:name ?name ; foaf:mbox ?mbox .
        FILTER ( lang(?name) = "ES" ) }
```

## Graph Patterns

- Basic Graph Pattern
  - Ensemble de triple patterns
- Group Graph Pattern
  - Ensemble de graph patterns
- Optional Graph Pattern
- Union/Alternative Graph Pattern
- RDF Dataset Graph Pattern

## Optional Graph Pattern

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
select ?name ?mbox
where { ?x foaf:name ?name .
       optional { ?x foaf:mbox ?mbox }
}
```

```
_:a foaf:name "Johnny"
_:a foaf:mbox <mailto:jlow@example.com>
_:a foaf:mbox <mailto:jlow@work.example>
_:b foaf:name "Peter"
```

```
?name="Johnny", ?mbox= <mailto:jlow@example.com>
?name="Johnny", ?mbox= <mailto:jlow@work.example>
?name="Peter"
```

## Multiple Optional Graph Patterns

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
select ?name ?mbox ?hpage
where { ?x foaf:name ?name .
       optional { ?x foaf:mbox ?mbox } .
       optional { ?x foaf:homepage ?hpage } .
}
```

```
_:a foaf:name "Johnny" .
_:a foaf:homepage <http://work.example.org/johnny/> .
_:b foaf:mbox <mailto:peter@work.example> .
_:b foaf:name "Peter"
```

```
?name="Johnny", ?hpage= <http://work.example.org/johnny/>
?name="Peter", ?mbox= <mailto:peter@work.example>
```

## Nested Optional Graph Pattern

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
select ?foafname ?mbox ?gname ?fname
where { ?x foaf:name ?foafname .
       optional { ?x foaf:mbox ?mbox } .
       optional { ?x vcard:N ?vc .
                 ?vc vcard:Given ?gname .
                 optional { ?vc vcard:Family ?fname }
               }
}
```

## Nested Optional Graph Pattern

- RDF data

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@work.example> .
_:a vcard:N _:x .
_:x vcard:Family "Hacker" .
_:x vcard:Given "Alice" .
_:b foaf:name "Bob" .
_:b foaf:mbox <mailto:bob@work.example> .
_:b vcard:N _:z .
_:z vcard:Family "Hacker" .
_:e foaf:name "Ella" .
_:e vcard:N _:y .
_:y vcard:Given "Eleanor" .
```

## Nested Optional Graph Pattern

```
<results>
<result>
  <binding name="foafName">Alice</binding>
  <binding name="mbox"><mailto:alice@work.example></binding>
  <binding name="gname">Alice</binding>
  <binding name="fname">Hacker</binding>
</result>
<result>
  <binding name="foafName">Bob</binding>
  <binding name="mbox"><mailto:bob@work.example></binding>
</result>
<result>
  <binding name="foafName">Ella</binding>
  <binding name="gname">Eleanor</binding>
</result>
</results>
```

## Exercice

### ■ RDF data

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/book/> .
@prefix ns: <http://example.org/ns#> .
:book1 dc:title "SPARQL Tutorial" .
:book1 ns:price 42 .
:book2 dc:title "The Semantic Web" .
:book2 ns:price 23 .
```

- Lister les titres des livres de la base et préciser leurs prix s'ils sont inférieurs à 30

## Constraints in Optional Graph Pattern

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
SELECT ?title ?price
WHERE { ?x dc:title ?title .
        OPTIONAL { ?x ns:price ?price . FILTER (?price < 30) } }
```

title	price
"SPARQL Tutorial"	
"The Semantic Web"	23

## Union/Alternative Graph Pattern

```
PREFIX dc10: <http://purl.org/dc/elements/1.0/>
PREFIX dc11: <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE { { ?book dc10:title ?title } UNION { ?book dc11:title ?title } }
```

```
@prefix dc10: <http://purl.org/dc/elements/1.0/> .
@prefix dc11: <http://purl.org/dc/elements/1.1/> .
_:a dc10:title "SPARQL Query Language Tutorial" .
_:a dc10:creator "Alice" .
_:b dc11:title "SPARQL Protocol Tutorial" .
_:b dc11:creator "Bob" .
_:c dc10:title "SPARQL" .
_:c dc11:title "SPARQL (updated)" .
```

## Union/Alternative Graph Pattern

```
PREFIX dc10: <http://purl.org/dc/elements/1.0/>
PREFIX dc11: <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE { { ?book dc10:title ?title } UNION { ?book dc11:title ?title } }
```

```
@prefix dc10: <http://purl.org/dc/elements/1.0/> .
@prefix dc11: <http://purl.org/dc/elements/1.1/> .
_:a dc10:title "SPARQL Query Language Tutorial" .
_:a dc10:creator "Alice" .
_:b dc11:title "SPARQL Protocol Tutorial" .
_:b dc11:creator "Bob" .
_:c dc10:title "SPARQL" .
_:c dc11:title "SPARQL (updated)" .
```

## Union/Alternative Graph Pattern

- Comment distinguer les titres enregistrés avec la version 1.0 du DC et ceux avec la version 1.1?

```
SELECT ?x ?y
WHERE { { ?book dc10:title ?x } UNION { ?book dc11:title ?y } }
```

x	y
"SPARQL"	"SPARQL (updated)"
"SPARQL Query Language Tutorial"	"SPARQL Protocol Tutorial"

## RDF Dataset Graph Pattern

- # Default graph (stored at <http://example.org/dft.ttl>)  
@prefix dc: <http://purl.org/dc/elements/1.1/> .  
<http://example.org/bob> dc:publisher "Bob Hacker"  
<http://example.org/alice> dc:publisher "Alice Hacker" .

- # Named graph: <http://example.org/alice>  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
\_:a foaf:name "Alice" .  
\_:a foaf:mbox <mailto:alice@work.example.org> .

- # Named graph: <http://example.org/alice>  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
\_:a foaf:name "Alice" .  
\_:a foaf:mbox <mailto:alice@work.example.org> .

## RDF Dataset Graph Pattern

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?who ?g ?mbox
FROM <http://example.org/dft.ttl>
FROM NAMED <http://example.org/alice>
FROM NAMED <http://example.org/bob>
WHERE { ?g dc:publisher ?who .
        GRAPH ?g { ?x foaf:mbox ?mbox } }

Who      g      mbox
"Bob Hacker" <http://example.org/bob> <mailto:bob@oldcorp.example.org>
"Alice Hacker" <http://example.org/alice> <mailto:alice@work.example.org>
```

## Exercise

```
# Named graph: http://example.org/foaf/aliceFoaf
_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@work.example> .
_:a foaf:knows _:b .
_:b foaf:name "Bob" .
_:b foaf:mbox <mailto:bob@work.example> .
_:b foaf:nick "Bobby" .
_:b rdfs:seeAlso <http://example.org/foaf/bobFoaf> .

# Named graph: http://example.org/foaf/bobFoaf
_:z foaf:mbox <mailto:bob@work.example> .
_:z rdfs:seeAlso <http://example.org/foaf/bobFoaf> .
_:z foaf:nick "Robert" .
```

## Exercise

- Quels surnoms sont donnés à celui ayant pour adresse mail <mailto:bob@work.example> , et dans quels graphes?

```
SELECT ?src ?bobNick
WHERE { GRAPH ?src
        { ?x foaf:mbox <mailto:bob@work.example> .
          ?x foaf:nick ?bobNick } }
```

src	bobNick
<http://example.org/foaf/aliceFoaf>	"Bobby"
<http://example.org/foaf/bobFoaf>	"Robert"

## Solution Sequence Modifiers

- Quelle est la réponse à cette requête?

```
SELECT ?x
WHERE { ?x ns:p ?y1 .
        ?x ns:p ?y2 }
FILTER ( ?y1 != ?y2 )
```

- RDF data
  - \_:a ns:p Johnny
  - \_:a ns:p Peter

## Solution Sequence Modifiers

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT DISTINCT ?name
WHERE { ?x foaf:name ?name }
ORDER BY ?name
OFFSET 10
LIMIT 5
```

## Query Result Forms

- SELECT
- ASK
- CONSTRUCT

## Asking yes or no Questions

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Alice" .
_:a foaf:homepage <http://work.example.org/alice/> .
_:b foaf:name "Bob" .
_:b foaf:mbox <mailto:bob@work.example> .
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
ASK { ?x foaf:name "Alice" }
```

```
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head></head>
  <results> <boolean>true</boolean> </results>
</sparql>
```

## Constructing an Output Graph

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
CONSTRUCT { <http://example.org/person#Alice> vcard:FN ?name }
WHERE { ?x foaf:name ?name }
```

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@example.org> .
```

```
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
<http://example.org/person#Alice> vcard:FN "Alice" .
```

## RDF Vocabulary Description Langage: RDF Schema

<http://www.w3.org/TR/rdf-schema/>

## RDF et RDF Schema

- RDF sert à annoter des ressources
- Ces annotations reposent sur des vocabulaires partagés : des ontologies
- RDF Schema permet de définir *en RDF* des vocabulaires RDF
- RDF Schema est un ensemble de classes et de propriétés RDF qui peuvent être utilisées pour définir des classes et des propriétés (en RDF)

## Une approche centrée sur les propriétés

- RDF diffère des systèmes de typage des langages à objets en ce sens que : plutôt que de définir une classe en terme de propriétés que possèdent ses instances, **RDFS décrit une propriété en terme de classes de ressources auxquelles elle s'applique**

```
public class Book {
  private String author;
  ... }
ns:author rdfs:type rdf:Property
ns:author rdfs:domain ns:Document
ns:author rdfs:range ns:Person
```

## Signature d'une propriété (1/2)

- Domain (ou conjonction de domaines) + Range
  - Une ressource peut être instance de plusieurs classes

```
ns:vitesse rdfs:type rdf:Property
ns:vitesse rdfs:domain ns:Objet
ns:vitesse rdfs:domain ns:Mobile
ns:vitesse rdfs:range rdfs:Literal
```

## Signature d'une propriété (2/2)

### ■ Typage

```
swl:enseignant rdf:type rdf:Property
swl:enseignant rdfs:domain swl:Cours
swl:enseignant rdfs:range swl:Person
_:x swl:enseignant swl:catherine
-----
_:x rdf:type swl:Cours
swl:catherine rdf:type swl:Person
```

## Hiérarchies de classes et de propriétés

■ ns:Book rdfs:subClassOf ns:Document

■ ns:author rdfs:subPropertyOf ns:actor

### ■ Héritage multiple

```
ns:EnseignantChercheur rdfs:subClassOf ns:Enseignant
ns:EnseignantChercheur rdfs:subClassOf ns:Chercheur
```

## Hiérarchies de classes et de propriétés

### ■ Typage

```
rdf:type rdfs:range rdfs:Class
ex:olivier rdf:type ex:Person
  => ex:Person rdf:type rdfs:Class

ex:olivier rdf:type ex:Researcher
ex:Researcher rdfs:subClassOf ex:Person
  => ex:olivier rdf:type ex:Person

ex:olivier ex:responsableDeCours ex:log3
ex:responsableDeCours rdfs:subPropertyOf ex:enseignant
  => ex:olivier ns:enseignant ex:log3
```

## Signatures de propriétés et héritage

```
Man subClassOf Primate
Chimpanzee subClassOf Primate
```

```
<rdf:Property rdf:ID='child'>
  <rdfs:domain rdf:resource='#Primate'/>
  <rdfs:range rdf:resource='#Primate'/>
</rdf:Property>

<rdf:Property rdf:ID='manChild'>
  <rdfs:subPropertyOf rdf:resource='#child'/>
  <rdfs:domain rdf:resource='#Man'/>
  <rdfs:range rdf:resource='#Man'/>
</rdf:Property>
```

## Syntaxe RDF/XML (1/2)

```
<rdf:Property rdf:ID='vitesse'>
  <rdfs:domain rdf:resource='#Objet'/>
  <rdfs:domain rdf:resource='#Mobile'/>
  <rdfs:range rdf:resource='&rdfs:Literal'/>
</rdf:Property>

<rdf:Description rdf:ID='vitesse'>
  <rdf:type>
    <rdf:Description rdf:about='http://...Property'/>
  </rdf:type>
  <rdf:domain>
    <rdf:Description rdf:about='#Objet'/>
  </rdf:domain>
  ...
</rdf:Description>
```

## Syntaxe RDF/XML (2/2)

```
<rdf:Class rdf:ID='Researcher'>
  <rdfs:subClassOf rdf:resource='#Person'/>
</rdf:Class>

<rdf:Property rdf:ID='responsableDeCours'>
  <rdfs:subPropertyOf rdf:resource='#enseignant'/>
</rdf:Property>
```

## Une annotation basée sur une ontologie

```
<rdf:Description rdf:about='olivier'>
  <rdf:type rdf:resource='#Chercheur'/>
  <ns:institut>INRIA</ns:institut>
  <swl:responsableDeCours>
    <Cours rdf:resource='http://courswiki.essi.fr...'>
  </swl:responsableDeCours>
  <rdf:type rdf:resource='#Enseignant'/>
  <ns:etablissement>EPU</ns:etablissement>
</rdf:Description>
```

## Cohabitation de plusieurs schémas

- Grâce aux namespaces XML

```
<rdf:Description
  xmlns:rdf='&rdf;'
  xmlns:ns='http://www.inria.fr/acacia/cours#'
  xmlns:swl='http://courswiki.essi.fr/LOG/SemanticWebLangages#'>

  <ns:institut>INRIA</ns:institut>
  <swl:etablissement>EPU</swl:etablissement>
</rdf:Description>
```

## Retour sur SPARQL

- RDF data

```
_:x rdf:type swl:Homme
_:y rdf:type swl:Femme
```

- RDFS data

```
swl:Homme rdfs:subClassOf swl:HumanBeing
swl:Femme rdfs:subClassOf swl:HumanBeing
```

- SPARQL

```
SELECT _:h WHERE { _:h rdf:type swl:HumanBeing }
```

## Retour sur SPARQL

- RDF(S) data

```
swl:Homme rdfs:subClassOf swl:HumanBeing
swl:Femme rdfs:subClassOf swl:HumanBeing
swl:HumanBeing rdf:type rdfs:Class
```

- SPARQL

```
SELECT _:c WHERE { _:c rdf:type swl:Class }
```

## Règles ontologiques

## If ... Then ...

- Compléter une base RDF à l'aide de règles définies dans l'ontologie
- Application des règles sur la base RDF
  - en chaînage avant
  - à saturation
- Condition d'application
  - Il existe une projection du graphe en prémisses de la règle sur la base RDF



## Règles usuelles

- Transitivité
  - subClassOf, subPropertyOf, ancetreDe
- Symétrie
  - marriedWith
- Propriétés inverses
  - parentDe inverse de enfantDe

## Règles liées au domaine

- Si un membre d'une équipe de recherche a un centre d'intérêt, alors cette équipe a aussi ce centre d'intérêt

```
?person interestedBy ?topic  
?person member ?researchTeam
```

```
-----  
?team interestedBy ?topic
```

## Syntaxe CORESE

```
<cos:rule>  
  <cos:if>  
    ?x rdf:type ex:Francais  
    ?x ex:age ?a  
    ?a > 17  
  </cos:if>  
  <cos:then>  
    ?x rdf:type ex:Adulte  
  </cos:then>  
</cos:rule>
```