

Slide 1



Cours No 5 - Langages d'ontologies : RDF et OWL



RDF : Resource Description Framework

RDF est un pilier de l'activité Web Sémantique du W3C pour créer des métadonnées :

Slide 2

- RDF : description de ressources Web (métadonnées)
- RDF Schéma (RDFS) : définition du vocabulaire de description (ontologie)
- Syntaxe XML : échange des métadonnées et des schémas



Projets et Applications

Slide 3

- PICS : Platform for Internet Content Selection
- RSS : RDF Site Summary (Netscape)
- ODB : **Open Directory Project**
- DAML : DARPA Agent Markup Language
- OWL : Ontology Web Language (W3C)



Logiciels

Slide 4

- Editeurs: Protégé, OilEd
- Parseurs: Sirpac, PerlXmlParser, RDF parser en XSLT, Validating RDF Parser (VRP)
- Entrepôts: Redland, rdfDB, Sesame, RDFSuite,
- Langages d'Interrogation: SiLRI, **RQL**



Niveaux du modèle RDF

Slide 5

- Physique : triplets
- Types de Base : ressources, propriétés, *déclarations*
- Types Complexe: collections, listes
- Schéma : classes, types de propriétés



Niveau Physique : Déclarations

Slide 6

- Ensemble d'URLs (ressources): U
- Ensemble de littéraux (chaînes de caractères): V
- Ensemble de *déclarations*: $T \subseteq U \times U \times (U \cup V)$

Le triplet (a, b, c) déclare que “la *propriété* b du *sujet* a a comme valeur l'*objet* c ”.



Exemples de Déclarations

Slide 7

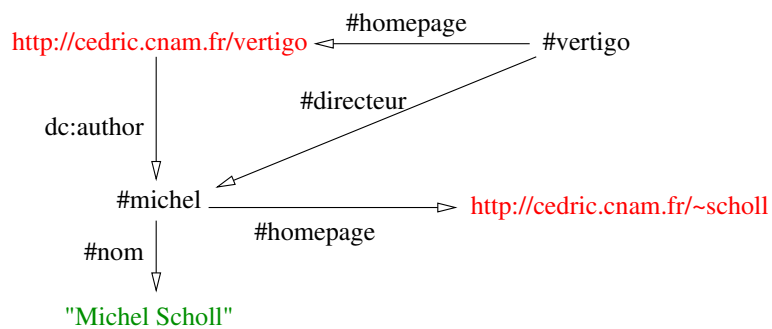
```
(http://cedric.cnam.fr/vertigo, dc:author, #michel)
(#vertigo, #homepage, http://cedric.cnam.fr/vertigo)
(#michel, #homepage, http://www-rocq.inria.fr/~scholl)
(#vertigo, #directeur, #michel)
(#michel, #nom, 'Michel Scholl')
```

- Ressources externes :
<http://cedric.cnam.fr/vertigo>, `dc:author`,
<http://www-rocq.inria.fr/~scholl>
- Ressources locales : `#michel`, `#vertigo`, `#homepage`,
`#directeur`, `#nom`
- Valeurs : `'Michel Scholl'`



Graphe RDF

Slide 8





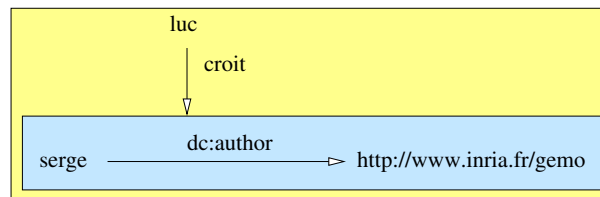
Réification

Une déclaration est une ressource : on peut créer déclarations sur des déclarations:

Luc croit que Serge est l'auteur de la page Gém0.

Slide 9

```
(#statement1, rdf:subject, #serge)
(#statement1, rdf:predicate, dc:author)
(#statement1, rdf:object, http://www.inria.fr/gemo)
(#luc, #croit, #statement1)
```



Types de Base

Espace de noms RDF rdf:

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

Les types de base permettent de distinguer les URLS qui sont des ressources, de propriétés et/ou des déclarations :

Slide 10

- La propriété rdf:type peut avoir comme valeur rdf:Resource, rdf:Property et/ou rdf:Statement.

Une URL peut avoir plus qu'un type!



Types Complexes: Containers et Listes

Un container est une ressource de type `rdfs:Container`. Cette classe a trois sous-classes :

- `rdf:Bag`: multi-ensemble de ressources
- `rdf:Sequence`: séquence de ressources
- `rdf:Alt`: énumération de ressource

Slide 11

L'appartenance à une collection est encodée par des propriétés `rdf:_1`, `rdf:_2`, `rdf:_3`, ...

Une collection est une ressource : on peut avoir des collections de collections.

Une liste est une ressource de type `rdf:List` :

- constructeurs: `rdf:first`, `rdf:rest`, `rdf:nil`



Collection: Exemple

Membres Verso :

```
(#students, rdf:type, rdf:Bag)
(#students, rdf:_1, #greg)
(#students, rdf:_2, #omar)
(#versomembers, rdf:type, rdf:Bag)
(#versomembers, rdf:_1, #serge)
(#versomembers, rdf:_2, #luc)
(#versomembers, rdf:_3, #michel)
(#versomembers, rdf:_4, #tova)
(#versomembers, rdf:_5, #claudel)
(#versomembers, rdf:_6, #students)
```

Slide 12



Schémas de Métadonnées

Slide 13

- *RDF* : ensemble de classes et de propriétés prédéfini.
- *RDF Schema* permet d'étendre ce vocabulaire avec des classes et types de propriétés spécifiques à une application ou un domaine.
- Les métadonnées RDF peuvent être validées par un schéma RDF.



Classes et Propriétés

`xmlns:rdfs=http://www.w3.org/2000/01/rdf-schema#`

- `rdfs:Class`
- `rdfs:subclassOf` : sous-classes

Slide 14

Un type de propriété est une ressource de type `rdf:Property`.
RDFS permet de restreindre le domaine et le co-domaine d'un type de propriété :

- `rdfs:subpropertyOf` : sous-propriétés
- `rdfs:domain` : domaine d'une propriété
- `rdfs:range` : co-domaine

Un type de propriétés peut avoir plusieurs domaines mais un seul co-domaine.



Exemple

Une classe est une ressource de type `rdfs:Class` :

Slide 15

```
(#Group, rdf:type, rdfs:Class)
(#ResearchGroup, rdf:type, rdfs:Class)
(#ResearchGroup, rdfs:subclassOf, #Group)
(#memberOf, rdf:type, rdf:Property)
(#memberOf, rdfs:domain, #Student)
(#memberOf, rdfs:domain, #Researcher)
(#memberOf, rdfs:range, #Group)
(#vertigo, rdf:type, #ResearchGroup)
```



Syntaxe XML

RDF propose une syntaxe XML pour échanger des métadonnées RDF et schémas RDFS :

- Document RDF/RDFS = séquence d'éléments de type `Description`

Slide 16

Avantages :

- Outils XML : parseurs, stockage
- Utilisation des espaces de noms pour l'intégration

Mais attention :

- Modèle XML : modèle d'arbres d'éléments et de sous-éléments
- Modèle RDF : modèle de graphe de ressources (noeuds et arcs sont étiquetés avec des URLs)



Exemple de document RDF

Slide 17

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:ex="http://mon_url.org/version/1.1/">
  <rdf:Description rdf:ID="michel"/>
  <rdf:Description rdf:ID="vertigo">
    <ex:homepage>
      <rdf:Description rdf:about='http://cedric.cnam.fr/vertigo">
        <dc:author>
          <rdf:Description rdf:about="#michel"/>
        </dc:author>
      </ex:homepage>
    <ex:directeur>
      <rdf:Description rdf:about="#michel" />
    </ex:directeur>
  </rdf:Description>
</rdf:RDF>
```



Création de propriétés

Définition des propriétés d'une ressource existante :

Slide 18

```
<rdf:Description rdf:about='http://www.inria.fr/gemo' >
  <rdf:type>
    <rdf:Description rdf:about='#Homepage' />
  </rdf:type>
  <dc:author resource='#serge' />
</rdf:Description>
```



Création de ressources locales

Définition d'une nouvelle ressource locale (avec ses propriétés) :

Slide 19

```
<rdf:Description rdf:ID='gemo' >
  <chief resource='#serge' />
</rdf:Description>
```

Remarque : `<chief resource='#serge' />` est une version compacte de

```
<chief>
  <rdf:Description rdf:about="#serge"/>
</chief>
```



Descriptions Distributives

Description de tous les éléments d'une collection

Slide 20

```
<rdf:Description
  aboutEach='#versomembers'>
  <likes>coffee</likes>
  <hates>bad research</hates>
</rdf:Description>

<rdf:Description
  aboutEachPrefix='http://www.rocq-inria*'>
  <copyright>INRIA</copyright>
</rdf:Description>
```



Syntaxe abrégé

Description :

```
<rdf:Description rdf:ID='serge' >
  <rdf:type resource='#Researcher' />
  <vehicle>
    <rdf:Description>
      <rdf:type resource='#Car' />
      <brand>Peugeot</car>
    </rdf:Description>
  </vehicle>
</rdf:Description>
```

Slide 21

Syntaxe abrégé :

```
<Researcher rdf:ID='serge'>
  <vehicle>
    <car brand='Peugeot' />
  </vehicle>
</Researcher>
```

Attention : Ce document représente un graphe.



Pour être complet...

Il est possible d'ajouter des informations supplémentaires pour faciliter l'interprétation d'une description :

- `xml:lang` : langue utilisée dans les descriptions
- `rdf:ParseType="literal"` : propriété de type XML
- `rdf:datatype="datatypeURI"` : la valeur de la propriété est une référence vers un type de données (par exemple, XML Schema)

Slide 22



Conclusion sur RDF

Slide 23

- RDF est un modèle puissant pour le partage de métadonnées.
- RDF est largement accepté et utilisé.
- RDF est limité :
 - pour décrire des contraintes sémantiques
 - la définition intensionnelle de concepts
 - le raisonnement

Slide 24



Web Ontology Language : OWL



OWL

“Extension” de RDFS pour faciliter

Slide 25

- le partage d'ontologies : RDFS considère la définition d'une ressource comme l'union de ces descriptions.
- l'évolution et intéropérabilité d'ontologies : *rdfs:subClassOf* et *rdfs:subPropertyOf* sont insuffisant pour l'intégration.
- la détection d'inconsistances



OWL : Définition de classes

Une classe peut être définie

Slide 26

- par l'énumération de ses instances
- par ses propriétés (définition intensionnelle)
- comme union, intersection, complément d'autres classes



Définition par énumération

Slide 27

```
<owl:Class rdf:ID="mycontinents">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Eurasia"/>
    <owl:Thing rdf:about="#Africa"/>
    <owl:Thing rdf:about="#NorthAmerica"/>
    <owl:Thing rdf:about="#SouthAmerica"/>
    <owl:Thing rdf:about="#Australia"/>
    <owl:Thing rdf:about="#Antarctica"/>
  </owl:oneOf>
</owl:Class>
```



Définition intensionnelle

Slide 28

La classe est définie par ses propriétés :

- Types de propriétés: *owl:allValuesFrom*,
owl:someValuesFrom
- Valeurs: *owl:hasValue*
- Cardinalité: *owl:maxCardinality*,
owl:minCardinality, *owl:Cardinality*

Exemple : Classe des ressources qui ont au moins une propriété
#membre de type #Student

```
<owl:Restriction>
  <owl:onProperty rdf:resource="#membre" />
  <owl:allValuesFrom rdf:resource="#Student" />
</owl:Restriction>
```



Intersection, union, complément

Définition d'une class à partir d'autres classes :

- owl:intersectionOf, owl:unionOf, owl:complementOf

Slide 29

```
<owl:Class>
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="mycontinents">
      <owl:Class>
        <owl:oneOf rdf:parseType="Collection">
          <owl:Thing rdf:about="#Europe" />
          <owl:Thing rdf:about="#Asia" />
          <owl:Thing rdf:about="#America" />
          <owl:Thing rdf:about="#Africa" />
          <owl:Thing rdf:about="#Australia" />
        </owl:oneOf>
      </owl:Class>
    </owl:intersectionOf>
  </owl:Class>
```



Relations entre classes

Slide 30

- *rdfs:subClassOf* : l'extension d'une classe est incluse dans l'extension de l'autre
- *owl:equivalentClass* : les deux classes ont la même extension, mais ne sont pas considérés comme identiques

```
<footballTeam owl:equivalentClass us:soccerTeam />
```
- *owl:disjointWith* : les deux class sont disjointes



Définition de propriétés

Slide 31

- RDF Schema: *rdfs:subPropertyOf*, *rdfs:domain* et *rdfs:range*
- Relations entre propriétés :
 - *owl:equivalentProperty*: les deux propriétés ont la même extension, mais ne sont pas identiques :
 - *owl:inverseOf*: une propriété est l'inverse de l'autre.


```
<owl:ObjectProperty rdf:ID="enfant">
  <owl:inverseOf rdf:resource="#parent"/>
</owl:ObjectProperty>
```

Slide 32

- Contraintes de cardinalité :
 - *owl:FunctionalProperty*: propriétés mono-valuées
 - *owl:InverseFunctionalProperty*:


```
<owl:InverseFunctionalProperty rdf:ID="mèreBiologique">
  <rdfs:domain rdf:resource="#femme"/>
  <rdfs:range rdf:resource="#personne"/>
</owl:InverseFunctionalProperty>
```
- Contraintes logiques :
 - *owl:SymmetricProperty* (époux)
 - *owl:TransitiveProperty* (ancêtre)



Commentaires et compatibilité

Commentaires :

- *rdfs:label*, *rdfs:comment*, *rdfs:seeAlso*,
rdfs:isDefinedBy

Slide 33

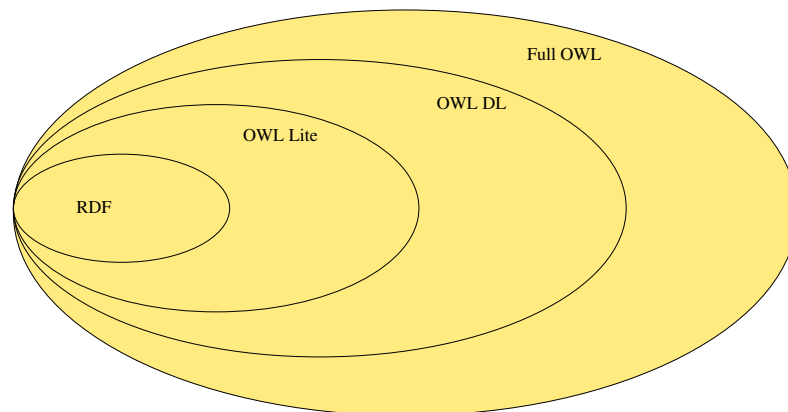
Compatibilité :

- *owl:versionInfo*, *owl:priorVersion*,
owl:backwardCompatibleWith,
owl:incompatibleWith
- *owl:DeprecatedClass*, *owl:DeprecatedProperty*



Hiérarchies de langages OWL

Slide 34





Prochain Cours

- Communautés Web et Portails Sémantiques

Slide 35