

# RDF Schéma

## RDFS

M1 SIC

Université de Tlemcen

2019-2020

# Introduction

- RDFS (pour RDF Schéma) a pour but d'étendre RDF en décrivant les ressources utilisées pour étiqueter les graphes.
- Il fournit un mécanisme permettant: de spécifier les classes dont les ressources seront des instances, ainsi que les propriétés.
- RDFS s'écrit toujours à l'aide **de triplets RDF**, en définissant la sémantique de nouveaux mots-clés.
- Fournit aussi des éléments de base pour la définition d'ontologies ou vocabulaires destinés à structurer des ressources RDF.
- La recommandation finale publiée par le W3C en février 2004. La dernière version est a l'url : <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/> .

# Concepts de Bases

- RDFS offre les moyens de définir un modèle (ou bien encore un schéma) de méta données qui permet de :
  - Définir des classes et des sous classes des ressources;
  - Donner du sens aux propriétés associées à une ressource ;
  - Formuler des contraintes sur les valeurs associées à une propriété.
- La spécification RDFS définit :
  - **les classes (sous classes)** : se sont des regroupements d'un ensemble de ressources de même nature.
  - **Des propriétés** : la définition des propriétés et relation entre classes.
  - **Des déclarations (signature)** : pour décrire l'utilisation concrète des propriétés et classes dans les données RDF.

# Concepts de Bases

- Le langage utilisé par RDFS se compose d'une collection de **ressources RDF** que l'on peut utiliser pour décrire les propriétés d'autres ressources RDF.
- Le vocabulaire central est défini dans un espace de noms « [rdfs](http://www.w3.org/2000/01/rdf-schema#) :  
<http://www.w3.org/2000/01/rdf-schema#> »
- La spécification emploie aussi l'espace de noms RDF  
<http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

# Les classes

- Les ressources peuvent être divisées en groupes appelés des classes.
- Les membres d'une classe sont dits des **instances** (instances) de la classe.
- Les classes sont identifiées également par des **URI** et peuvent être décrites en utilisant des propriétés RDF.

# Les classes: `rdfs:Resource`

- **`rdfs:Resource`**
  - Toutes les choses décrites par RDF sont appelées des ressources (resources) et sont des **instances** de la classe **`rdfs:Resource`**.
  - C'est la classe de toutes les choses. Toutes les autres classes sont des **sous-classes** de cette classe.

# Les classes: `rdfs:Class`

- Elle définit la notion de "classe" qui est un ensemble de plusieurs objets.
- C'est la classe des ressources qui sont des classes RDF.
- **Exemple : la définition de la classe cours en RDFS.**

```
<rdfs:Class rdf:about='#Cours' />
```

# Les classes: `rdf:Property`

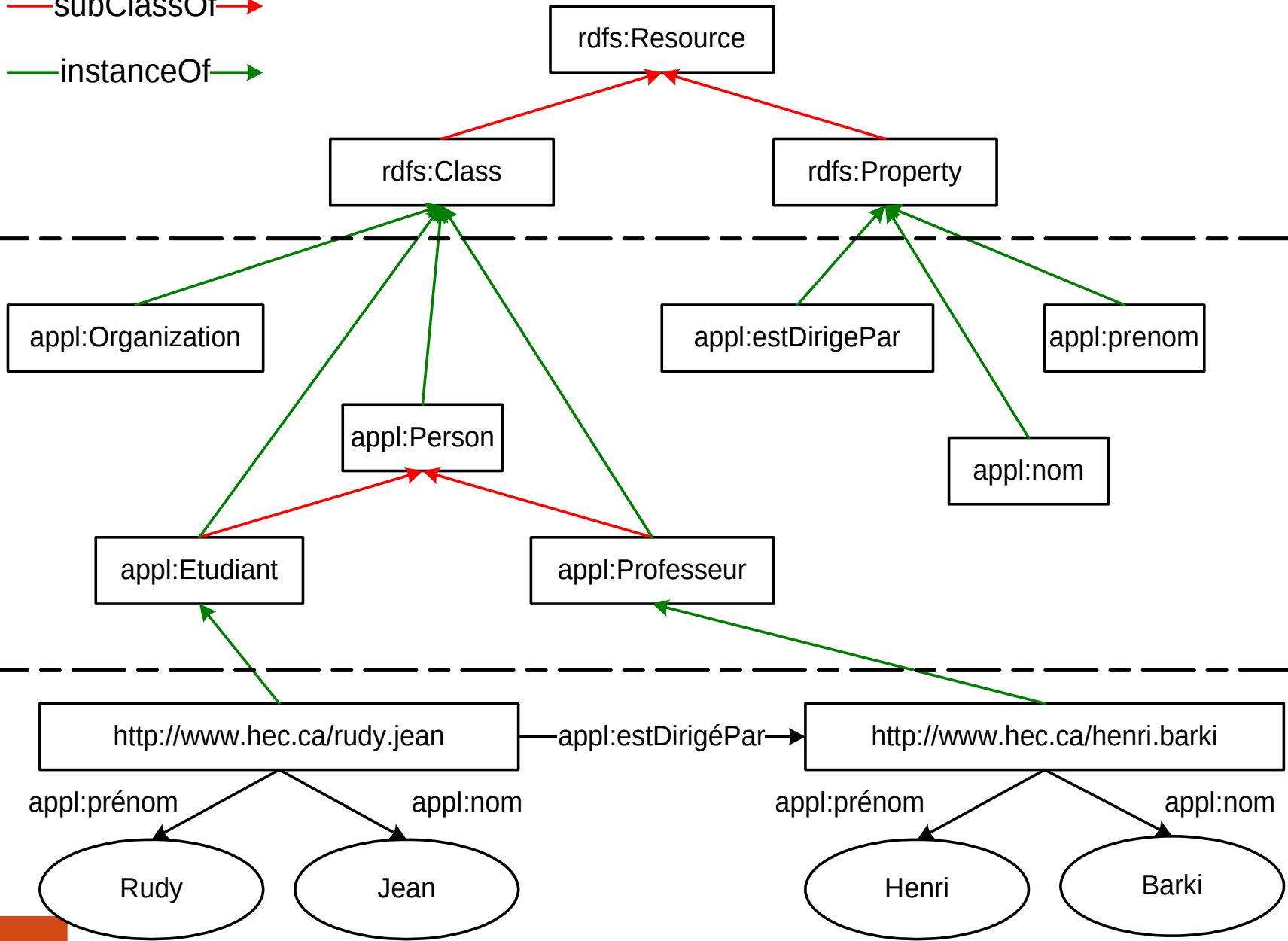
- La classe `rdf:Property` est la classe des propriétés RDF.
- La classe `rdf:Property` est une instance de `rdfs:Class`.
- Exemple : la définition de la propriété « titre » en RDFS.

```
<rdf:Property rdf:about='#titre'>  
... <!--définition-->  
</rdf:Property>
```



—subClassOf—→

—instanceOf—→



Niveau RDF/RDFS & namespace

Application schéma & namespace

Application données actuelles

# Les classes: `rdfs:Literal` , `rdfs:Datatype`

- La classe **`rdfs:Literal`** est la classe des valeurs littérales (literal values) telles que les chaînes et les nombres.
- La classe **`rdfs:Datatype`** est la classe des types de données, Chaque instance de `rdfs:Datatype` est une sous-classe de **`rdfs:Literal`**

## Les classes: **rdfs:Container**, **rdfs:ContainerMembershipProperty**

- La classe **rdfs:Container** est la super-classe des conteneurs RDF, comme : **rdf:Bag**, **rdf:Seq**, **rdf:Alt**.
- La classe **rdfs:ContainerMembershipProperty** a comme instances les propriétés **rdf:\_1**, **rdf:\_2**, **rdf:\_3** . Est une sous-classe de **rdf:Property**.

# Les propriétés

- **rdf:type**: La propriété **rdf:type** est une instance de la classe **rdf:Property**, utilisée pour déclarer qu'une ressource est une **instance** d'une classe.
- **Exemple** : la déclaration RDF qui indique que la ressource 'IA2' est de type 'cours'

```
<rdf:Description rdf:about='http://www.univ-  
tlemcen.dz/cours/SIC/IA2'>  
<rdf:type rdf:resource='#Cours' />  
<titre>le web sémantique </titre>  
</rdf:Description>
```

# Les propriétés: rdfs:subClassOf

- Permet de définir qu'une classe est un sous-ensemble d'une autre classe.
- Toutes les instances d'une classe (fille) sont des instances d'une autre (mère).
- **Exemple :**

```
<rdfs:Class rdf:about='#Cours' />
<rdfs:Class rdf:about='#CoursSIC'>
  <rdfs:subClassOf rdf:resource='#Cours' />
</rdfs:Class>
<rdfs:Class rdf:about='#CoursMID'>
  <rdfs:subClassOf rdf:resource='#Cours' />
</rdfs:Class>
<rdfs:Class rdf:about='#CoursCommun'>
  <rdfs:subClassOf rdf:resource='#CoursSIC' />
<rdfs:subClassOf rdf:resource='#CoursMID' />
</rdfs:Class>
```

## Les propriétés: rdfs:subPropertyOf (1/2)

- C'est pour définir une sous propriété d'une propriété.
- Toutes les ressources reliées par une propriété (fille) sont également reliées par une autre (mère).
- **Exemple** : la propriété «Acoordonateur » est une sous propriété de la propriété « Aenseignant »

```
<rdf:Property    rdf:about='#Acoordonateur'>  
<rdfs:subPropertyOf  rdf:resource='#Aenseignant' />  
</rdf:Property>
```

## Les propriétés: rdfs:subPropertyOf (2/2)

- La déclaration RDF suivante implique la dernière:

```
<ns:Cours rdf:about='http://www.univ-tlemcen.dz/cours/IA2'>  
  <ns:coordinateur rdf:resource='http://www.univ-  
tlemcen.dz/ProfID01' />  
</ns:Cours>
```

```
<ns:Cours  rdf:about='http://www.univ-tlemcen.dz/cours/IA2'>  
  <ns:enseignant rdf:resource='http://www.univ-  
tlemcen.dz/ProfID01' />  
</ns:Cours>
```

# Les propriétés: `rdfs:label`

- La propriété **`rdfs:label`** est une instance de la classe **`rdf:Property`**, utilisée pour fournir une version lisible par un humain du nom d'une ressource.
- **Exemple :**

```
<rdf:Property rdf:about='#name'>  
<rdfs:label xml:lang='fr'>nom</rdfs:label>  
<rdfs:label xml:lang='en'>name</rdfs:label>  
....  
</rdf:Property>
```



# Les propriétés: rdfs: comment

- Utilisée pour fournir une description lisible par un humain d'une ressource.
- **Exemple :**

```
<rdfs:Class rdf:about='#Woman'>  
  <rdfs:subClassOf rdf:resource="#Person" />  
  <rdfs:comment xml:lang='fr'>  
    une personne adulte du sexe féminin  
  </rdfs:comment>  
  <rdfs:comment xml:lang='en'>  
    a female adult Person  
  </rdfs:comment>  
</rdfs:Class>
```

## les propriétés :les constraints sur les propriétés (Signature)

- Une signature d'une propriété est définie à travers :
  - La spécification des classes auxquelles on peut affecter la propriété (Sujet de la propriété). En RDFs ce la est fait par la propriété « **rdfs:domain** ».
  - La spécification de champ d'application de la propriété (s'applique à la classe). En RDFs ce la est fait par la propriété : « **rdfs:range** ».

# les propriétés :les constraints sur les propriétés (Signature)

## ● Exemple :

```
<rdf:Property rdf:about='#titre'>  
  <rdfs:domain rdf:resource='#Cours' />  
  <rdfs:range rdf:resource='&rdfs;Literal' />  
</rdf:Property>
```

```
<rdf:Property rdf:about='#enseigne '>  
  <rdfs:domain rdf:resource='#Cours' />  
  <rdfs:range rdf:resource='#Personne' />  
</rdf:Property>
```

```
<rdf:Property rdf:about='#nom'>  
  <rdfs:domain rdf:resource='#Personne' />  
  <rdfs:range rdf:resource='&rdfs;Literal' />  
</rdf:Property>
```

# Propriétés auxiliaires

- **rdfs:seeAlso** :
  - Cette propriété est utilisée pour indiquer une ressource susceptible de fournir des informations supplémentaires à propos de la ressource « sujet ».
  - **Exemple** :

```
<rdfs:Class rdf:about="#Man">  
<rdfs:seeAlso rdf:resource='#Person' />  
</rdfs:Class>
```

# Propriétés auxiliaires

- **rdfs:isDefinedBy**

- Cette propriété est utilisable pour indiquer un vocabulaire RDF dans lequel une ressource est décrite.

- **Exemple :**

```
<rdfs:Class rdf:about="http://exemple.net/elements#Livre">  
<rdfs:isDefinedBy  
rdf:resource="http://exemple.net/elements#" />  
<rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-  
schema#Resource" />  
<rdfs:label>Livre</rdfs:label>  
<rdfs:comment>Un livre écrit par un auteur</rdfs:comment>  
</rdfs:Class>
```

# Les inférences (1/3)

- Si  $(c_2, \text{subClassOf}, c_1)$  et  $(x, \text{type}, c_2)$   
alors  $(x, \text{type}, c_1)$ 
  - *Exemple* :  $(\text{Lo}, \text{type}, \text{Homme}) \Rightarrow (\text{Lo}, \text{type}, \text{Humain})$
- Si  $(p_2, \text{subPropertyOf}, p_1)$  et  $(x, p_2, y)$   
alors  $(x, p_1, y)$ 
  - *Exemple* :  $(\text{Lo}, \text{auteur}, \text{Note}) \Rightarrow (\text{Lo}, \text{créateur}, \text{Note})$
- Si  $(c_3, \text{subClassOf}, c_2)$  et  $(c_2, \text{subClassOf}, c_1)$   
alors  $(c_3, \text{subClassOf}, c_1)$  (transitivité)
- Si  $(p_3, \text{subPropertyOf}, p_2)$  et  $(p_2, \text{subPropertyOf}, p_1)$  alors  $(p_3, \text{subPropertyOf}, p_1)$  (transitivité)
- Idem réflexivité subClassOf et subPropertyOf

# Les inférences (2/3)

- Si (p, **range**, c) et (x, p, y) alors (y, type, c)
- Si (p, **domain**, c) et (x, p, y) alors (x, type, c)
  - *Exemple* : (aPourMere, range, Femme)  
(Fabien, aPourMere, Josette)  
⇒ (Josette, type, Femme)
- La **signature est héritée**
- Signature effective = **conjonction** des signatures héritées et spécifiées

# Les inférences (3/3)

## ● Résumé

<i>Rule name</i>	<i>If</i>	<i>Then add</i>
rdf1	(x p y)	(p rdf:type rdf:Property)
rdfs2	(p rdfs:domain c),(x p y)	(x rdf:type c)
rdfs3	(p rdfs:range c),(x p y)	(y rdf:type c)
rdfs4a	(x p y)	(x rdf:type rdfs:Resource)
rdfs4b	(x p y)	(y rdf:type rdfs:Resource)
rdfs5	(p rdfs:subPropertyOf q),(q rdfs:subPropertyOf r)	(p rdfs:subPropertyOf r)
rdfs6	(p rdf:type rdf:Property)	(p rdfs:subPropertyOf p)
rdfs7	(p rdfs:subPropertyOf q),(x p y)	(x q y)
rdfs8	(c rdf:type rdfs:Class)	(c rdfs:subClassOf rdfs:Resource)
rdfs9	(c rdfs:subClassOf d),(x rdf:type c)	(x rdf:type d)
rdfs10	(c rdf:type rdfs:Class)	(c rdfs:subClassOf c)
rdfs11	(c rdfs:subClassOf d),(d rdfs:subClassOf e)	(c rdfs:subClassOf e)
rdfs12	(p rdf:type rdfs:ContainerMembershipProperty)	(p rdfs:subPropertyOf rdfs:member)
rdfs13	(x rdf:type rdfs:Datatype)	(x rdfs:subClassOf rdfs:Literal)



# Insuffisances du RDFS

- **Contraintes de cardinalité** sur les propriétés:
  - Exemple: une personne (Person) a exactement un seul père biologique ;
  - Par exemple pouvoir dire que, pour une équipe de football, la propriété ex:hasPlayers a 11 valeurs, alors que, pour une équipe de basket ball, la même propriété ne devrait avoir que 5 valeurs ;
- **Équivalence de classes**: particulièrement utile pour “fusionner” des Métadonnées provenant de deux vocabulaires différents ;
- Décrire de nouvelles classes en fonction de combinaisons (par exemple, unions et intersections) d'autres classes, ou de dire que deux classes sont disjointes.

# En savoir plus

- <http://www.w3.org/TR/rdf-schema/>
- <http://www.w3.org/2001/sw/wiki/RDFS>.
- <http://xml.mfd-consult.dk/ws/2003/01/rdfs/>