Introduction à RDF-S
(RDF-Schema)

Bernard ESPINASSE

Aix-Marseille Université
LIS UMR CNRS 7020

Septembre 2019

- **Introduction à RDFS**
- **Méta-modèle RDFS**
- **Inférences dans RDFS**
- **Intérêts et limites de RDFS**

Références

- **Livres, articles et rapports :**
  - O. Corby and F. Gandon and C. Faron-Zucker, Le Web sémantique : comment lier les données et les schémas sur le web ? Dunod, 2012.
  - G. Antoniou, Van Harmelen F., A Semantic Web Primer, The MIT Press Cambridge, Massachusetts London, England, 1999.
  - John Hebeler and Matthew Fisher and Ryan Blace and Andrew Perez-Lopez and Mike Dean, Semantic Web Programming, Wiley, 2009.
- **Web W3C :**
  - Page du W3C : https://www.w3.org/TR/rdf-schema/
  - …
- **Cours/tutoriaux :**
  - Cours de M. Gagnon, Ecole Polytechnique de Montréal, 2007.
  - Cours de S. Staab, ISWeb – Lecture Semantic Web, Univ. Koblenz-Landau.
  - Cours de J.-B. Hook, Université Paris Sud, 2013.
  - Cours de C.A. Caron, Université de Lille 3, 2014.
  - Cours de O. Papini, Aix-Marseille Université, 2015.
  - Cours de I-M. Bilasco, Université de Lille, 2018.
  - …

Plan

1. Introduction à RDF-S

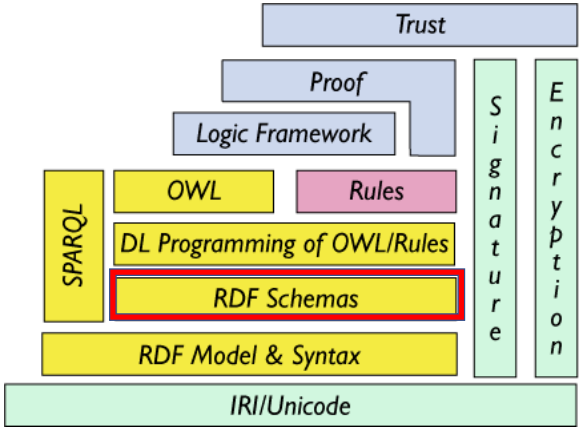
- **Sémantique d'un vocabulaire**
- **Ressources et littéraux en RDF**
- **Définitions de classes et propriétés en RDFS**

Sémantique d'un vocabulaire

- **RDF** permet de définir des graphes étiquetés - **Graphes RDF**, en utilisant des ressources du web **sans vraiment de sémantique**
- Pour donner de la sémantique des étiquettes du graphe RDF, il faut des **vocabulaires plus riches**
- Ces vocabulaires permettront :
  - d'améliorer l'indexation des contenus en ligne,
  - un traitement plus efficace des requêtes,
  - des réponses plus pertinentes, et une meilleure interopérabilité des systèmes, …
- Pour formaliser cette sémantique on utilisera des **ontologies**
- Les **ontologies** permettent aussi un enrichissement des données en utilisant de l'inférence (données intentionnelles vs extensionnelles)

⇨ **RDFS** (RDF-Schema), extension de RDF, **permet de construire des ontologies légères basées sur RDF**

Place de RDF Schema dans le gâteau du WS …



Source : W3C, T Berners-Lee, Ivan Herman

Introduction à RDFS (RDF-Schema)

- Recommandation du W3C depuis 2004
- Permet de définir des vocabulaires RDF, en nommant :
  - des classes
  - des relations de sous-classe
  - des relations de sous-propriété
  - le typage des prédicats : domaine, co-domaine, …
- Permet de définir une organisation hiérarchique des classes et des propriétés
- RDFS est donc un **premier langage de définition d'ontologie**
- RDFS a une expressivité réduite, permet des inférences simples (par rapport à d'autres langage de définition d'ontologie comme OWL), mais trop d'expressivité n'est pas forcément une qualité ...

⇨ **Ainsi RDFS étend RDF à la description d'ontologies (légères)**

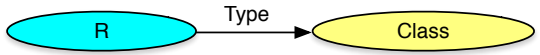
Définitions de classes et propriétés en RDFS

Par rapport à la définition de classes et propriétés d'un langage de POO comme Java :

- **RDFS opte pour une approche centrée sur les propriétés** :
  - au lieu de **définir des classes** en donnant leurs propriétés, on définit des propriétés en donnant leur **domaine** et **co-domaine**
- **Exemple** :
  - en Java : « la *classe* eg:Document a un *attribut* eg:author de *type* eg:Person »
  - en RDFS : « la *propriété* eg:author a pour *domaine* la classe eg:Document, et pour *co-domaine* la classe eg:Person »
- Une **ressource peut avoir plusieurs types**,
- Une **ressource peut être instance de plusieurs classes** (plusieurs rdf :type pour une même ressources)
- Lorsqu'on écrit un document RDF, la **définition du vocabulaire** à l'aide de RDFS n'est **pas obligatoire** (préfixe **rdfs**)

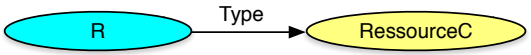
Classes en RDFS

- **Ressources et classes :**
  - Les ressources peuvent être « rangées » dans des **classes**
  - Une **classe** est une **ressource**
- **Déclarations :**
  - d'une *ressource* comme *classe* : propriété **rdfs:Class**




```
RessourceC rdf:type rdfs:Class
```

  - d'une *ressource* comme une *instance de classe* : propriété **rdfs:Class**

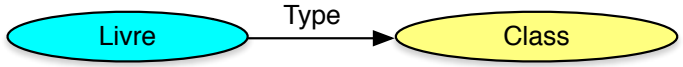



```
ressource rdf:type RessourceC
```

- **Extensions :**
  - L'ensemble des instances d'une classe est appelé son extension
  - 2 classes différentes peuvent avoir la même extension

Classes en RDFS : exemple

- **Ressources et classes :**



(Livre, Type, Class)

- **XML :**

```
<rdf:Description rdf:ID="Livre">
    <rdf:type rdf:resource=
      "http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
```

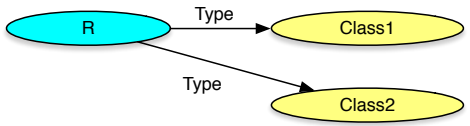
- **Version simplifiée XML :**

```
<rdfs:Class rdf:ID="Livre"/>
```

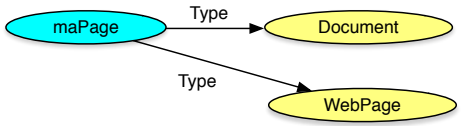

Multi-instanciation

- Possibilité pour une ressource d'avoir plusieurs types de classe
- Permet de ne pas avoir à déclarer une nouvelle classe

```
ressource rdf:type Class1
ressource rdf:type Class2
```



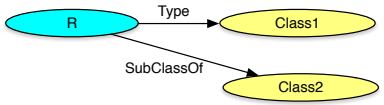
- Exemple :



(maPage, type, Document)
(maPage, type, WebPage)

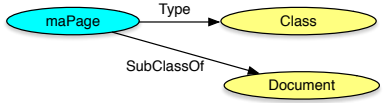
Sous-classes en RDFS

- **Une classe peut être sous-classe d'une ou plusieurs classes :**
  - On utilise la propriété **rdfs:subClassOf** pour exprimer **qu'une classe est sous-classe d'une autre classe**
  - Si C sous-classe de C' alors toutes les instances de C sont instances de C'
    - par inférence sur le Schéma
    - par transitivité




```
ressource rdf:type Class1
ressource rdfs:subClassOf Class2
```

  - Exemple :



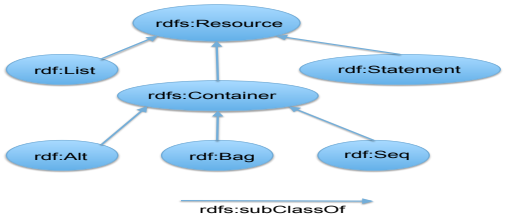
(maPage, type, Class)
(maPage, subClassOf, Document)

Exemple de classes et sous classes

Source : A.-C. Caron

Exemple : la classe rdf:Bag, sous-classe de rdfs:Container :

```
<rdfs:Class rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag">
<rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
   <rdfs:label>Bag</rdfs:label>
<rdfs:comment>The class of unordered containers.</rdfs:comment>
<rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-
schema#Container"/>
</rdfs:Class>
```



Exemple de vocabulaire et d'instance

Source : A.-C. Caron

- 1. Vocabulaire pour définir des monuments à visiter :

```
espace de nom tour: <http://www.fil.univ-lille1.fr/WS/schema>
tour:Monument a rdfs:Class .
tour:Eglise a rdfs:Class ;
    rdfs:subClassOf tour:Monument .
tour:Cathedrale a rdfs:Class ;
    rdfs:subClassOf tour:Eglise .
```

- 2. Instance : Notre Dame de Paris :

```
espace de nom nddp: <http://www.fil.univ-lille1.fr/WS/data>
nddp:NDP a tour:Cathedrale ;
    tour:altitude "126.7" ;
    rdfs:label "Notre Dame de Paris" ;
    tour:oeuvres [ a rdf:Bag ;
        rdf:_1 <http://fr.wikipedia.org/wiki/Le_Sacre_de_Napoléon> ;
        rdf:_2 nddp:NDP_VH ] .
nddp:NDP_VH dc:creator "Victor Hugo" ;
    dc:title "Notre Dame de Paris" ;
    dc:subject nddp:NDP
```


Ressources et littéraux (1)

- **RDFS décrit des ressources**, toutes instances de la classe rdfs:Resource, y compris rdfs:Resource et rdfs:Class
- Toutes les primitives du langage sont des instances soit de la classe **rdfs:Class** soit de la classe **rdf:Property**
- Les littéraux sont instances de la classe **rdfs:Literal**
- **rdfs:Datatype** est la classe de tous les types de données
- Toute **instance** de rdfs:Datatype est sous-classe de rdfs:Literal

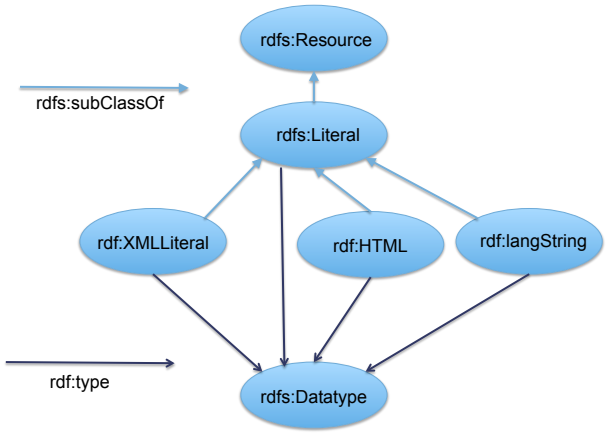
- **Exemple** : le type de données **rdf:XMLLiteral**

```
<rdfs:Datatype rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#XMLLiteral">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
    <rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
    <rdfs:label>XMLLiteral</rdfs:label>
    <rdfs:comment>The class of XML literal values.</rdfs:comment>
</rdfs:Datatype>
```

permet d'écrire un contenu XML, interprété en tant que littéral (pas en tant que RDF/XML).

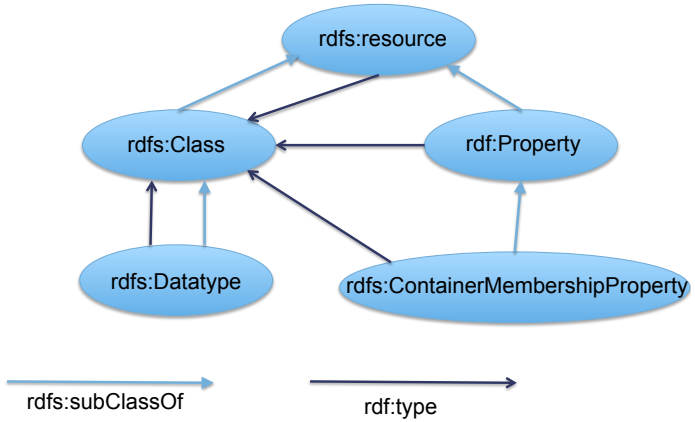
Ressources et littéraux (2)

Source : A.-C. Caron



Ressources et littéraux (3)

Source : A.-C. Caron



Propriété

- Une **propriété** est de type rdf:Property
- Dans le schéma précédent, figure la **classe** rdf:Property, instance de rdfs:Class
- rdfs:subPropertyOf définit la **relation de sous-propriété** entre 2 propriétés.

  Si P(s, o) et P sous-propriété de P0 alors P0(s, o)
- On peut définir le type du sujet (**domaine**) et/ou de l'objet (**co-domaine**) d'une propriété :
  - rdfs:domain (le domaine) : définit la classe (rdf :Class) des sujets liés à une propriété P :
  - rdfs:range : (le co-domaine) : définit la classe ou le type de données des valeurs de la propriété P :
- Une **propriété** peut avoir **plusieurs domaines et plusieurs co-domaines** (voir plus loin la partie « inférence »).

Quelques propriétés particulières

- **Propriétés complémentaires :**
  - **rdfs:seeAlso** permet d'associer 2 classes ou 2 propriétés.

    Cette relation permet d'une part d'associer des informations complémentaires à une ressource, mais aussi d'augmenter les liens entre les données du web.
  - **rdfs:isDefinedBy** permet d'indiquer une ressource définissant la ressource sujet
- **Propriétés de documentation :**
  - **rdfs:label** permet d'associer aux classes et propriétés que l'on définit des noms (labels) compréhensibles par des humains.
  - **rdfs:comment** permet d'associer un commentaire à une classe ou une propriété, pour en donner une description ou une définition.

Modélisation RDFS : exemple 1 (1)

Source : Cours de Bilasco inspiré de RDF Primer W3C

- Exemple de schéma RDFS (Turtle) :

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

<http://www.labd.org/2015/voitures/schema#Personne>
   a rdfs:Class ;
   rdfs:comment "La classe personne" .

<http://www.labd.org/2015/voitures/schema#Vehicule>
   a rdfs:Class ;
   rdfs:comment "La classe vehicule" .

<http://www.labd.org/2015/voitures/schema#Voiture>
   a rdfs:Class ;
   rdfs:comment "La classe voiture" ;

   rdfs:subClassOf <http://www.labd.org/2015/voitures/schema#Vehicule> .
<http://www.labd.org/2015/voitures/schema#conducteur>
   a rdf:Property ;
   rdfs:range <http://www.labd.org/2015/voitures/schema#Personne> ;
   rdfs:domain <http://www.labd.org/2015/voitures/schema#Vehicule> .
```

Modélisation RDFS : exemple 1 (2)

Source : Cours de Bilasco inspiré de RDF Primer W3C

- Exemple de schéma (XML) :

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF 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="http://www.labd.org/2015/voitures/schema#Personne">
    <rdfs:comment>La classe personne</rdfs:comment>
  </rdfs:Class>

  <rdfs:Class rdf:about="http://www.labd.org/2015/voitures/schema#Vehicule">
    <rdfs:comment>La classe vehicule</rdfs:comment>
  </rdfs:Class>

  <rdfs:Class rdf:about="http://www.labd.org/2015/voitures/schema#Voiture">
    <rdfs:comment>La classe voiture</rdfs:comment>
    <rdfs:subClassOf rdf:resource="http://www.labd.org/2015/voitures/schema#Vehicule"/>
  </rdfs:Class>

  <rdf:Property rdf:about="http://www.labd.org/2015/voitures/schema#conducteur">
    <rdfs:range rdf:resource="http://www.labd.org/2015/voitures/schema#Personne"/>
    <rdfs:domain rdf:resource="http://www.labd.org/2015/voitures/schema#Vehicule"/>
  </rdf:Property>

</rdf:RDF>
```


Modélisation RDFS : exemple 1 (3)

Source : Cours de Bilasco inspiré de RDF Primer W3C

- Exemple d'instance du schéma précédent :

```
@prefix ns0: <http://www.labd.org/2015/voitures/schema#> .

<http://www.labd.org/2015/voitures/data#vo001>
   a ns0:Voiture ;
   ns0:conducteur <http://www.labd.org/2015/voitures/data#p101> .

<http://www.labd.org/2015/voitures/data#vo002>
   a ns0:Voiture  ;
   ns0:conducteur <http://www.labd.org/2015/voitures/data#p102> .

<http://www.labd.org/2015/voitures/data#p102> a ns0:Personne .
<http://www.labd.org/2015/voitures/data#p101> a ns0:Personne .
```


Modélisation RDFS : exemple 1 (4)

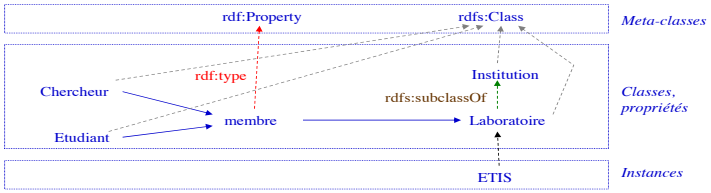
Source : Cours de Bilasco inspiré de RDF Primer W3C

- Graphe RDF associé :

Modélisation RDFS : exemple 2

Source : Dan Vodislav (Univ. Cergy Pontoise)

- Description de *classes* et de *types de propriétés :*
  - Classes: *rdfs:Class*, *rdfs:subclassOf*
  - Propriétés: *rdfs:subpropertyOf*, *rdfs:domain, rdfs:range*



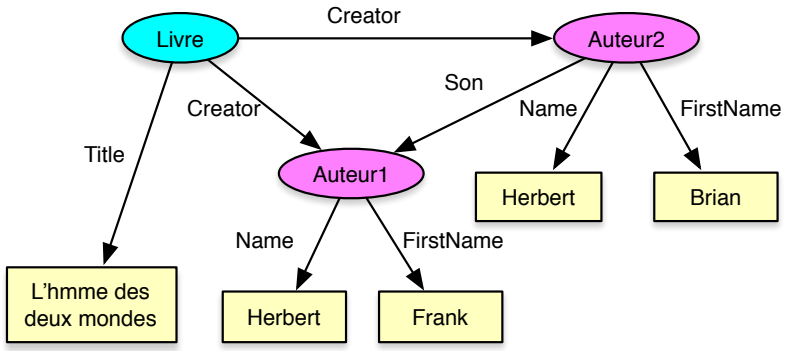
- Triplets RDFS :

(#Institution, rdf:type, rdfs:Class)
(#Laboratoire, rdf:type, rdfs:Class)
(#Laboratoire, rdfs:subclassOf, #Institution)
(#membre, rdf:type, rdf:Property)
(#membre, rdfs:domain, #Etudiant)
(#membre, rdfs:domain, #Chercheur)
(#membre, rdfs:range, #Institution)
(#ETIS, rdf:type, #Laboratoire)

Modélisation RDFS : exemple 3 (1)

Source : Fournier

Considérons le livre « L'Homme de deux mondes », premier roman écrit par Frank Herbert (disparu en 1986), en collaboration avec son fils Brian :



Modélisation RDFS : exemple 3 (2)

- Schéma RDFS associé à l'exemple (en XML):

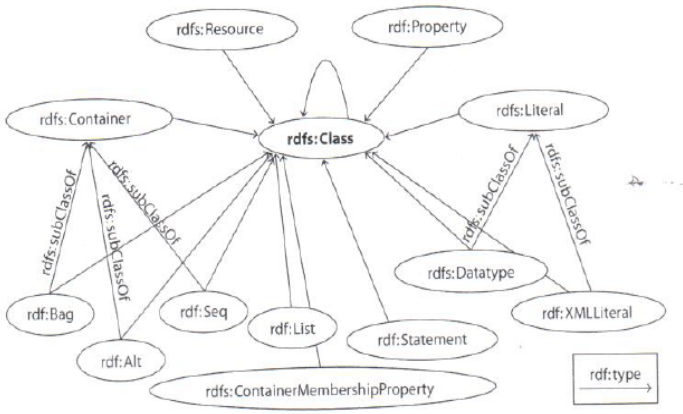
```
<rdfs:Class rdf:ID="Livre"/>
<rdfs:Class rdf:ID="Personne"/>

<rdf:Property rdf:ID="Title">
      <rdfs:domain rdf:resource="#Livre"/>
      <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/> </rdf:Property>
<rdf:Property rdf:ID="Creator">
      <rdfs:domain rdf:resource="#Livre"/>
      <rdfs:range rdf:resource="#Personne"/> </rdf:Property>
<rdf:Property rdf:ID="Name">
      <rdfs:domain rdf:resource="#Livre"/>
      <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/> </rdf:Property>
<rdf:Property rdf:ID="FirstName">
      <rdfs:domain rdf:resource="#Livre"/>
      <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/> </rdf:Property>
<rdf:Property rdf:ID="Son">
      <rdfs:domain rdf:resource="#Personne"/>
      <rdfs:range rdf:resource="#Personne"/>
</rdf:Property>
```


2. Méta-modèle RDF-S

- **Méta-modèle de déclaration des classes**
- **Méta-modèle de déclaration des propriétés**
- **Méta-modèle général**

Méta-modèle RDFS : déclaration de classes

Source : Le Web sémantique, F.Gandon, C.Faron-Zucker, O.Corby



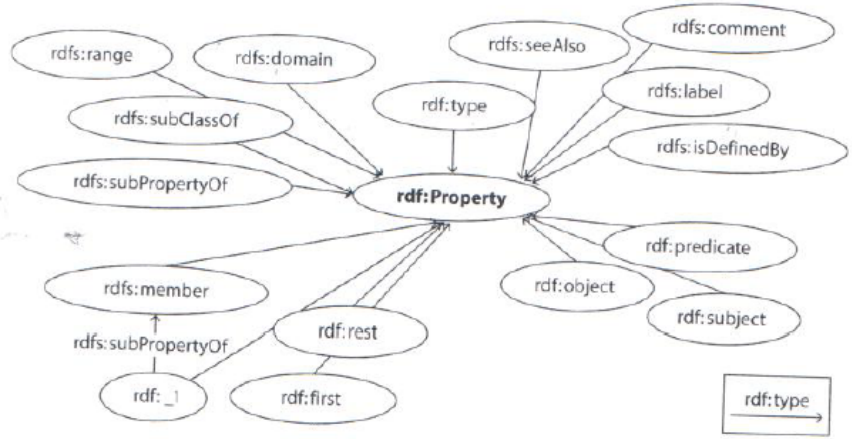
Les classes RDF/RDFS

Source : A.-C. Caron

Classe	Commentaire
rdfs:Resource	Tout est ressource
rdfs:Literal	Un littéral donc du texte
rdf:langString	Un littéral avec une indication de langue (par exemple @fr)
rdf:HTML	Littéral HTML
rdf:XMLLiteral	Littéral XML
rdfs:Class	Une classe
rdf:Property	Une propriété
rdfs:Datatype	Un type de données
rdf:Statement	Un énoncé RDF (s,p,o)
rdf:Bag	Un container non ordonné
rdf:Seq	Un container ordonné
rdf:Alt	Un container d'alternatives
rdfs:Container	Un container RDF
rdfs:ContainerMembershipProperty	Une propriété d'appartenance à un container (rdf:_1, rdf:_2, … ss-prop de rdfs:member)
rdf:List	Une liste RDF (rdf:nil est instance de rdf:List)

Méta-modèle RDFS : déclaration de propriétés

Source : Le Web sémantique, F.Gandon, C.Faron-Zucker, O.Corby



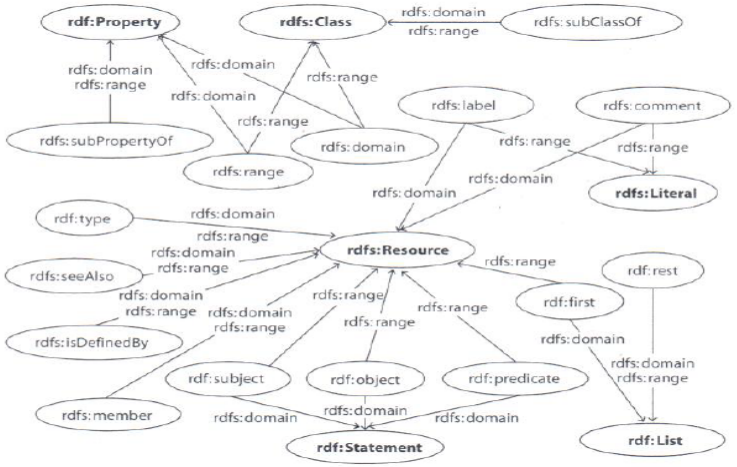
Les propriétés RDF/RDFS

Source : A.-C. Caron

Propriété	Domaine	Co-domaine
rdf:type	rdfs:Resource	rdfs:Class
rdfs:subclassOf	rdfs:Class	rdfs:Class
rdfs:subPropertyOf	rdf:Property	rdf:Property
rdfs:domain	rdf:Property	rdfs:Class
rdfs:range	rdf:Property	rdfs:Class
rdfs:label	rdfs:Resource	rdfs:Literal
rdfs:comment	rdfs:Resource	rdfs:Literal
rdfs:member	rdfs:Resource	rdfs:Resource
rdf:first	rdf:List	rdfs:Resource
rdf:rest	rdf:List	rdf:List
rdfs:seeAlso	rdfs:Resource	rdfs:Resource
rdfs:isDefinedBy	rdfs:Resource	rdfs:Resource
rdf:value	rdfs:Resource	rdfs:Resource
rdf:subject	rdfs:Statement	rdfs:Resource
rdf:predicate	rdfs:Statement	rdfs:Resource
Rdf:object	rdfs:Statement	rdfs:Resource

Méta-modèle RDFS général

Source : Le Web sémantique, F.Gandon, C.Faron-Zucker, O.Corby



3. Inférences en RDF-S

- **Saturation d'un modèle RDFS**
- **Règles d'inférence / de saturation**

Saturation d'un modèle RDFS

- **But : permet de rendre explicite tous les triplets implicites d'une base de données RDFS**
- Ce qui peut être saturé :
  - le **schéma** : ne fait pas grand sens, excepté pour des requêtes sur le schéma.
  - les **données basées sur le schéma associé** : pour garanti une exhausivité de réponses aux requêtes sur la base de données.

- **Avantage :**
  - facile à faire
- **Inconvénients :**
  - requiert de l'espace mémoire
  - non robuste aux mises à jours (induit des coûts de maintenance)

⇨ **usage de règles d'inférence (ou de saturation)**

Règles d'inférences/de saturation (1)

- RDFS permet **d'inférer de nouveaux triplets**, à partir de :
  - triplets existants,
  - relations de sous-classe,
  - relations sous-propriété,
  - domaines et co-domaines.
- **Différents types de règles :**
  - Règles basées sur les relations de sous-classes et de sous-propriétés
  - Règles basées sur la transitivité
  - Règles basées sur les domaines et co-domaines

Règles d'inférences/de saturation (1)

Règles basées sur les relations de sous-classes et de sous-propriétés :

- **R1 : Si** x de type C **ET** C $\subseteq$ C' **ALORS** x de type C'

$$\frac{(x\ ,\ \texttt{rdf:type}\ ,\ C) \wedge (C\ ,\ \texttt{rdfs:subclassOf}\ ,\ C')}{(x\ ,\ \texttt{rdf:type}\ ,\ C')}$$

Ex :

`(Man, rdfs:subClassOf, Person)` ET `(Tom, rdf:type, Man)`
ALORS `(Tom, rdf:type, Person)`

- **R2 : Si** p(x,y) **ET** p sous-propriétés de p' **ALORS** p'(x,y)

$$\frac{(x\ ,\ p\ ,\ y) \wedge (p\ ,\ \texttt{rdfs:subPropertyOf}\ ,\ p')}{(x\ ,\ p'\ ,\ y)}$$

Ex :

`(author, rdfs:subPropertyOf, creator)`
ET `(Tom, author, Report12)`
ALORS `(Tom, creator, Report12)`

Règles d'inférences/de saturation (2)

Règles basées sur la transitivité :

- **Transitivité des relations de sous-classe :**
  - **R3 : Si** C sous-classe de C' **ET** C' sous-classe de C'' **ALORS** C sous-classe de C''

$$\frac{(C\ ,\ \texttt{rdfs:subclassOf}\ ,\ C') \wedge (C'\ ,\ \texttt{rdfs:subclassOf}\ ,\ C'')}{(C\ ,\ \texttt{rdfs:subclassOf}\ ,\ C'')}$$

- **Transitivité des relations de sous-propriété :**
  - **R4 : Si** p sous-propriété de p' **ET** p' sous-propriété de p'' **ALORS** p sous-propriété de p''

$$\frac{(p\ ,\ \texttt{rdfs:subPropertyOf}\ ,\ p') \wedge (p'\ ,\ \texttt{rdfs:subPropertyOf}\ ,\ p'')}{(p\ ,\ \texttt{rdfs:subPropertyOf}\ ,\ p'')}$$

Règles d'inférences/de saturation (3)

Règles basées sur les domaines et co-domaines

- **R5 : Si** p propriété de domaine C **ET** p(x,y) **ALORS** x de type C

$$\frac{(x\ ,\ p\ ,\ y) \wedge (p\ ,\ \texttt{rdfs:domain}\ ,\ C)}{(x\ ,\ \texttt{rdf:type}\ ,\ C)}$$

Ex :

(author, rdfs:domain, Human) ET (Tom, author, Report12)
ALORS (Tom, rdf:type, Human)

- **R6 : Si** p propriété de co-domaine C **ET** p(x,y) **ALORS** y de type C

$$\frac{(x\ ,\ p\ ,\ y) \wedge (p\ ,\ \texttt{rdfs:range}\ ,\ C)}{(y\ ,\ \texttt{rdf:type}\ ,\ C)}$$

Ex :

(author,rdfs:range, Work) ET (Tom,author, Report12)
ALORS (Report12, rdf:type, Work)

Exemple d'inférences (1)

Source : Cours de Bilasco inspiré de RDF Primer W3C

- Donner les inférences faites par RDFS :

c:creator rdfs:domain c:Person
i:Man241 c:creator i:Image262
i:Man241 rdf:type c:Person
c:author rdfs:subPropertyOf c:creator
c:author rdfs:range c:Document
i:Woman297 c:author i:Book812
i:Book812 rdf:type c:Document
i:Woman297 c:creator i:Book812
i:Woman297 rdf:type c:Person
c:aSoutenu rdfs:domain c:Docteur
c:aSoutenu rdfs:range c:These
i:Woman297 c:aSoutenu i:t127
i:Woman297 rdf:type c:Docteur
i:t127 rdf:type c:These
c:nbDeRoues rdfs:domain c:Vehicule
i:Car207 c:nbDeRoues "4"^^xsd:integer
i:Car207 rdf:type c:Vehicule

Exemple d'inférences (2)

Source : Cours de Bilasco inspiré de RDF Primer W3C

- Donner les inférences faites par RDFS :

c:creator rdfs:domain c:Person
i:Man241 c:creator i:Image262
i:Man241 rdf:type c:Person
c:author rdfs:subPropertyOf c:creator
c:author rdfs:range c:Document
i:Woman297 c:author i:Book812
i:Book812 rdf:type c:Document
i:Woman297 c:creator i:Book812
i:Woman297 rdf:type c:Person
c:aSoutenu rdfs:domain c:Docteur
c:aSoutenu rdfs:range c:These
i:Woman297 c:aSoutenu i:t127
i:Woman297 rdf:type c:Docteur
i:t127 rdf:type c:These
c:nbDeRoues rdfs:domain c:Vehicule
i:Car207 c:nbDeRoues "4"^^xsd:integer
i:Car207 rdf:type c:Vehicule

Exemple d'inférences (3)

Source : Cours de Bilasco inspiré de RDF Primer W3C

- Donner les inférences faites par RDFS :

c:creator rdfs:domain c:Person
i:Man241 c:creator i:Image262
i:Man241 rdf:type c:Person
c:author rdfs:subPropertyOf c:creator
c:author rdfs:range c:Document
i:Woman297 c:author i:Book812
i:Book812 rdf:type c:Document
i:Woman297 c:creator i:Book812
i:Woman297 rdf:type c:Person
c:aSoutenu rdfs:domain c:Docteur
c:aSoutenu rdfs:range c:These
i:Woman297 c:aSoutenu i:t127
i:Woman297 rdf:type c:Docteur
i:t127 rdf:type c:These
c:nbDeRoues rdfs:domain c:Vehicule
i:Car207 c:nbDeRoues "4"^^xsd:integer
i:Car207 rdf:type c:Vehicule

Exemple d'inférences (4)

Source : Cours de Bilasco inspiré de RDF Primer W3C

- Donner les inférences faites par RDFS :

c:creator rdfs:domain c:Person
i:Man241 c:creator i:Image262
i:Man241 rdf:type c:Person
c:author rdfs:subPropertyOf c:creator
c:author rdfs:range c:Document
i:Woman297 c:author i:Book812
i:Book812 rdf:type c:Document
i:Woman297 c:creator i:Book812
i:Woman297 rdf:type c:Person
c:aSoutenu rdfs:domain c:Docteur
c:aSoutenu rdfs:range c:These
i:Woman297 c:aSoutenu i:t127
i:Woman297 rdf:type c:Docteur
i:t127 rdf:type c:These
c:nbDeRoues rdfs:domain c:Vehicule
i:Car207 c:nbDeRoues "4"^^xsd:integer
i:Car207 rdf:type c:Vehicule

3. Conclusion

- **Intérêts de RDFS**
- **Limites de RDFS**

Intérêts de RDFS

- RDFS étend RDF à la **définition d'ontologie légères**
- En définissant des **vocabulaires** RDF avec :
  - des classes
  - des relations de sous-classe
  - des relations de sous-propriété
  - le typage des prédicats : domaine, co-domaine, …
- En définissant une organisation hiérarchique des classes et des propriétés

Limites de RDFS

- **Puissance expressive insuffisante, il manque :**
  - Cardinalités (min et max)
  - Décomposition (disjoint, exhaustivité)
  - Axiomes
  - Négation
- **Problèmes dans RDF/RDFS :**
  - Pas de distinction entre classes et instances

    <Espece, type, Class>

    <Lion, type, Espece>

    <Simba, type, Lion>
  - Les propriétés peuvent avoir des propriétés …
  - Pas de distinction entre constructeurs du langage et les termes de l'ontologie.

⇨Pour dépasser ces limites, passage à **OWL (Ontology Web Language)**