



Cours Web Sémantique- 4IFPartie 4 : Ontologies/RDFS/OWL

Sylvie Calabretto et Mehdi Kaytoue





Plan du cours

- Introduction au web sémantique
- Décrire les données du Web : RDF
- Représenter les connaissances : ontologies, RDFS et OWL
- Interroger le Web : SPARQL
- Raisonner : inférences et RIF



Introduction

- RDF permet de tisser un graphe étiqueté entre les données
- Comment capturer formellement certains aspects du sens de ces étiquettes ?
 - •Il est nécessaire de se doter de vocabulaires plus ou moins formels en fonction des besoins
 - •Il est nécessaire de référencer ces vocabulaires dans les graphes RDF



Typologie des vocabulaires

- Trois grands types de vocabulaires plus ou moins contrôlés :
 - Les ontologies
 - •Les thésaurus
 - Les folksonomies
- Ces trois structures de données participent à l'indexation des contenus afin de :
 - Naviguer de façon pertinente
 - Rechercher de façon précise
 - Traiter des masses de ressources grandissantes



Les ontologies

Les ontologies permettent de :

se donner un vocabulaire partagé pour décrire un domaine

- le concept Amine représente la classes des amines
- le concept AmineSecondaire représente la classe des amines secondaires

se donner des primitives de typage des classes et de relations

- AmineSecondaire est une sous-classe de Amine
- Propanolol est une instance du concept AmineSecondaire

raisonner = déduire de nouveaux faits à partir de ceux qu'on a déjà

 on déduit de ce qui est dit plus haut que Propanolol est aussi une instance du concept Amine



Les ontologies : définitions

- Une ontologie est une «spécification formelle et explicite d'une conceptualisation partagée» (Gruber, 1993).
 - spécification formelle : compréhensible par une machine
 - spécification explicite : les concepts, relations, axiomes sont explicitement définis
 - conceptualisation : modèle abstrait d'une partie du monde que l'on veut représenter
 - partagée : les connaissances représentées sont partagées par une communauté



Définition formelle d'une ontologie

Une structure d'ontologie est un quintuplet $O := \{C, \mathcal{R}, \mathcal{H}^C, rel, \mathcal{A}^O\}$

 ${\it C}$ et ${\it R}$: ensembles disjoints des **concepts** et des **relations**

 \mathcal{H}^{C} hiérarchie (taxonomie) de concepts : $\mathcal{H}^{C} \subseteq C \times C$, $\mathcal{H}^{C}(c_1, c_2)$ signifie que c_1 est un sous-concept de c_2 (relation orientée)

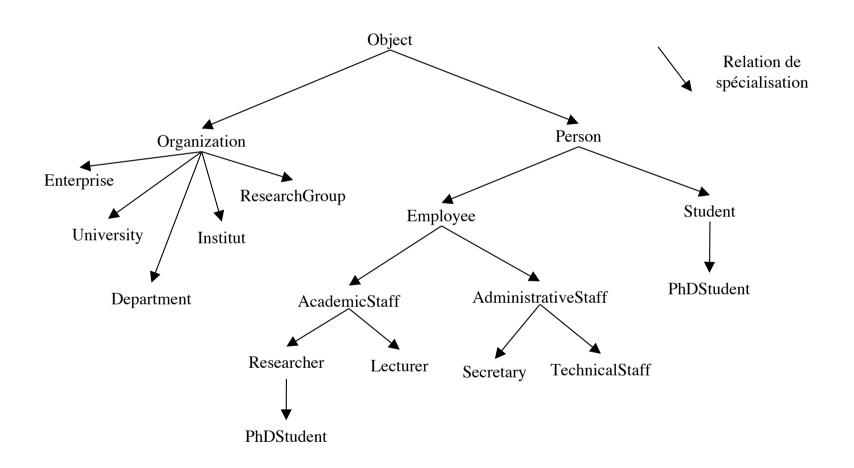
Rel: relation rel: $R \rightarrow C \times C$ (définit des relations sémantiques non taxonomiques)

$$rel(R) = (C_1, C_2)$$
 s'écrit aussi $R(C_1, C_2)$

 \mathcal{A}^O : ensemble **d'axiomes**, exprimés dans un langage logique adapté (logique de description, logique du 1er ordre)

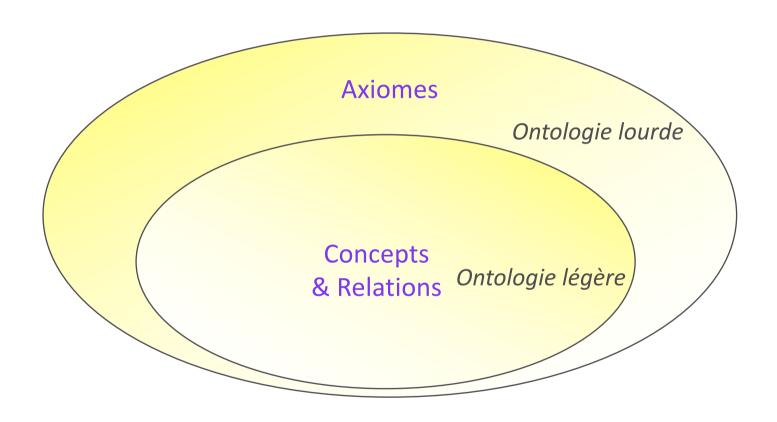


Une partie de l'ontologie (KA) initiative





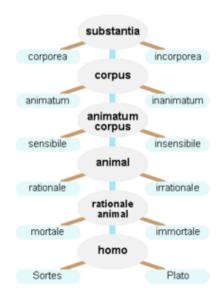
Ontologie légère / Ontologie lourde





Exemples d'ontologies

- UMLS (Unified Medical Language System) : http://www.nlm.nih.gov/research/umls/
- OpenCyc: http://www.opencyc.org/
- Wordnet: http://wordnet.princeton.edu/
- Ontologie de Porphyre





Wordnet

Dans Wordnet, un nœud est appelé Synset. Il contient

3 parties:

- Le terme représentant du Synset
- Les termes synonymes (séparés par des virgules)
- •Le glossaire : contient une définition du concept
- Relations sémantiques entre synsets :

hyperonymie- hyponymie (is-a), antonymie (relation entre ensembles de mots qui, par leur sens, s'opposent), etc.

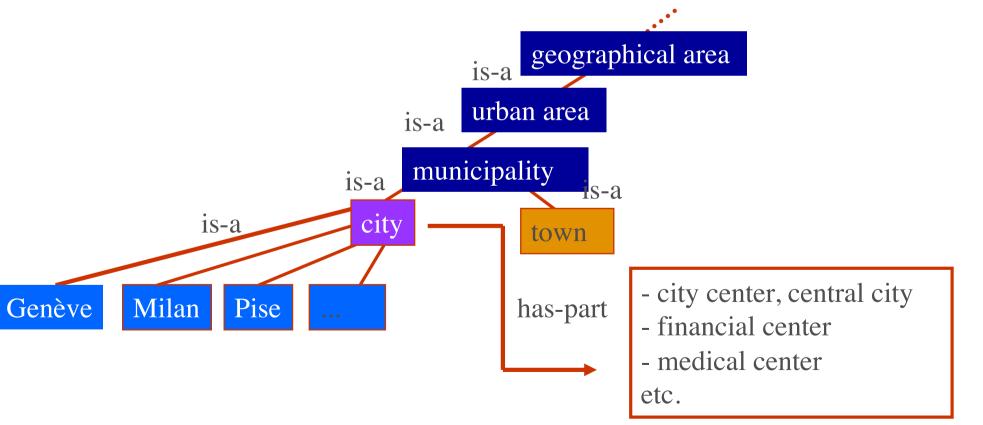
Exemple : « mouse »

mouse – (any of numerous small rodents ...)

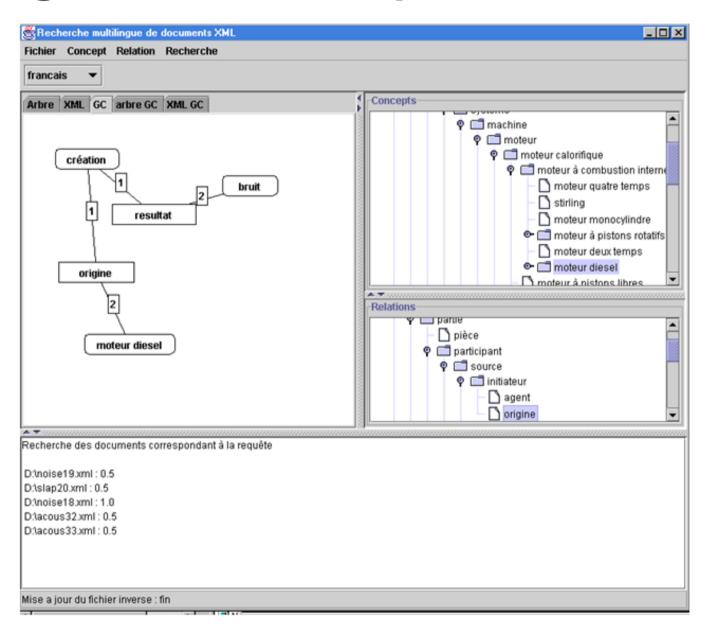
mouse, computer mouse -- (a hand-operated electronic device that controls the coordinates of a cursor on your computer screen as you move it around on a pad)



Ensemble de synonymes (Synset) 1.city, metropolis, urban center -- (a large and densely populated urban area; may include several independant administrative districts; etc) 2. city -- (an incorporated administrative district established by state charter) 3. city, metropolis -- (people living in a large densely populated municipality)



Ontologie de la mécanique automobile





Les thésaurus

- Objectif : constituer un vocabulaire normalisé pour indexer un corpus documentaire et faciliter les recherches dans ce corpus
- Le thésaurus permet à la fois de guider l'indexation et d'enrichir les requêtes
- Contient le lexique de tous les termes normalisés (masculin singulier) du langage documentaire, reliés par des relations sémantiques :
 - •relation d'équivalence
 - •relation hiérarchique
 - relation d 'association



Les thésaurus

Relation d'équivalence :

```
voiture == automobile
```

Relation hiérarchique (entre terme générique et terme spécifique) :
 véhicule > voiture

Relation d'association :

voiture ⇔ conducteur

 L'indexation impose d'utiliser les termes descripteurs les plus spécifiques pour ensuite utiliser la hiérarchie et les équivalences du thésaurus

Les documents seront indexés par cabriolet et identifiables lors d'une recherche par des termes plus génériques comme véhicule ou voiture ou leurs synonymes



Les thésaurus : exemples

- Rameau (Répertoire d'autorité-matière encyclopédique et alphabétique unifié) : thésaurus de la BNF (http://www.bnf.fr/web-bnf/infopro/rameau)
- Agrovoc : domaine de l'agriculture (http://www.fao.org/agrovoc)
- MeSH (Medical Subject Headings): domaine biomédical (http://www.ncbi.nlm.nih.gov/mesh)



Agrovoc





Thesaurus vs. Ontologie

Thesaurus

```
Contenu : Termes de la langue naturelle, 3 relations : «is_a», « synonyme », «voir_aussi»
```

Utilisé par un agent humain (documentaliste, spécialiste) pour indexer des documents

Ontologie

Contenu : une taxinomie des concepts, une taxinomie de relation

Décrite dans un langage de représentation des connaissances et exploitée par un système informatique

Possibilité de comparer et de classer des concepts / Inférences



Les folksonomies

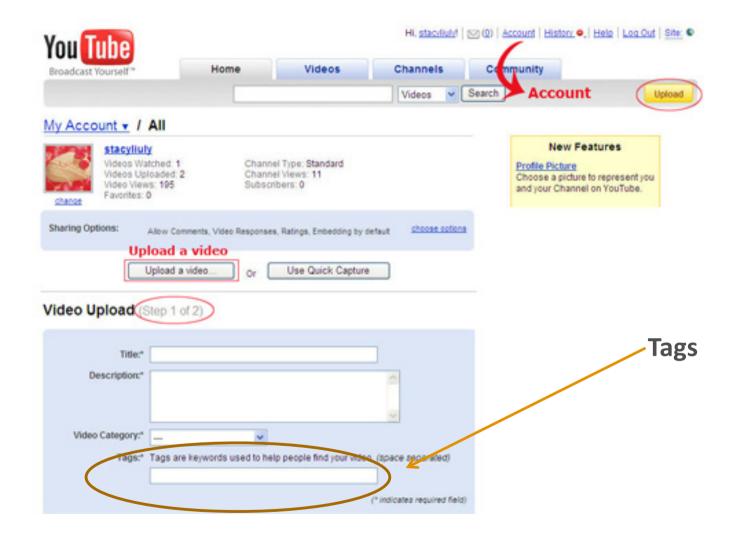
- Terme issu du croisement entre folks et taxonomy
- Contexte : web 2.0 ou web social



- •Cette pratique repose sur l'association libre de mots-clés aux ressources du web sans contrainte sur les termes utilisés
- •Un tel terme associé à une ressource pour le catégoriser librement est appelé un *tag* (étiquette)
- L'ensemble des tags utilisés, des ressources taguées, des utilisateurs auteurs de ces tags et des liens qui unissent les tags, les ressources et les tagueurs forment un graphe tripartite que l'on appelle une folksonomie.



Les folksonomies





Les folksonomies



Les avantages :

- •Simplicité de l'approche de classification de contenus : pas de vocabulaire prédéfini, évolution constante du vocabulaire, etc.
- •La structure de la folksonomie propose de facto un graphe de navigation entre les ressources en suivant les tags communs à plusieurs ressources
- •La popularité des tags, dans une folksonomie, est rendue visible par l'utilisation de nuages de tags (tag clouds)

Les limites :

- •Problèmes d'ambiguïté et de synonymies de tags pour la recherche d'information : un même tag peut faire référence à plusieurs notions distinctes et différents tags peuvent faire référence à la même notion
- Pas de liens possibles entre les tags



Conclusion sur les vocabulaires

- Un certain nombre de systèmes s'intéressent maintenant à faire cohabiter plusieurs structures/vocabulaires : ontologies, thésaurus et folksonomies
- Des schémas existent pour représenter et mélanger ces structures sur le web







RDF Schema: ontologies légères

 Nommer et définir un vocabulaire conceptuel consensuel et faire des inférences élémentaires

Nommer les classes de ressources existantes

Nommer les relations/propriétés qui existent entre ces classes et donner leur signature

Liens hiérarchiques entre classes et entre propriétés





OWL: ontologies lourdes

OWL plus expressif que RDFS

OWL Lite / DL / Full

Fondé sur les logiques de description

- Définition de classes par énumération, union, intersection, complément, disjonction, restriction valeur et cardinalité des propriétés
- Caractérisation des propriétés (symétrique, transitive, fonctionnelle, inversement fonctionnelle, inverse)
- Gestion des équivalences, versions, documenter





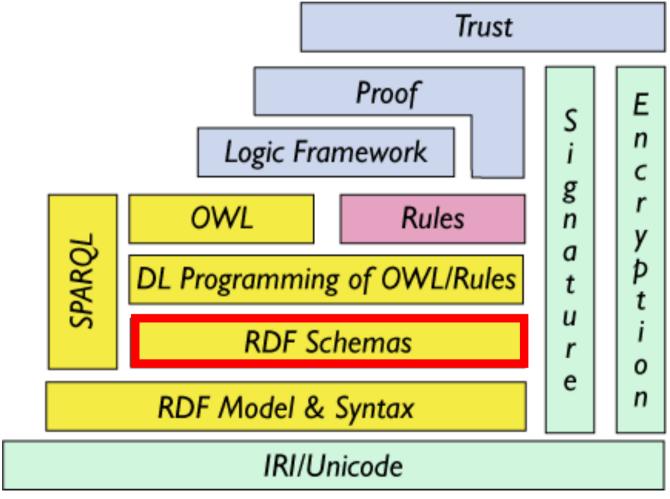
RDFS Schema pour les ontologies légères

Recommandation du W3C depuis 2004



Le gâteau du web sémantique...

- RDF : modèle de triplets pour annoter des ressources
- RDFS: décrit le vocabulaire (ontologies) utilisé pour ces annotations





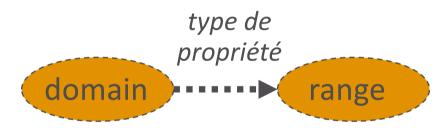
W3C, T Berners-Lee, Ivan Herman

RDFS : Echanger des ontologies légères

- Nommer et définir un vocabulaire conceptuel consensuel et faire des inférences élémentaires
- Nommer les classes de ressources existantes
- Nommer les relations qui existent entre ces classes
- Donner la signature de ces relations:

Le domaine (d'où la relation part)

Le range (où la relation arrive)



- Documenter ces notions en langue naturelle
- Squelette taxonomique d'une ontologie

Liens hiérarchiques des classes

Liens hiérarchiques des propriétés

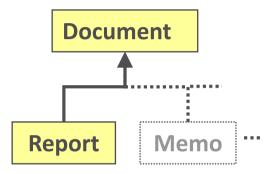


Ontologie (concepts/classes)

Class Document

Class Report

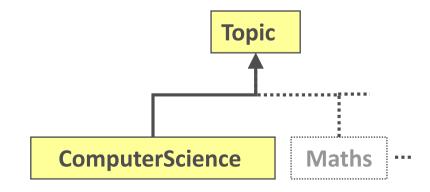
subClassOf Document



Class Topic

Class ComputerScience

subClassOf Topic





Ontologie (relations/propriétés)

Property concern

domain Document

range Topic

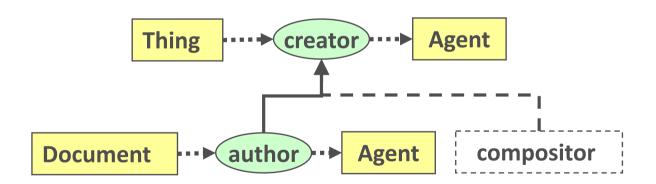


Property author

domain Document

range Person

subPropertyOf creator



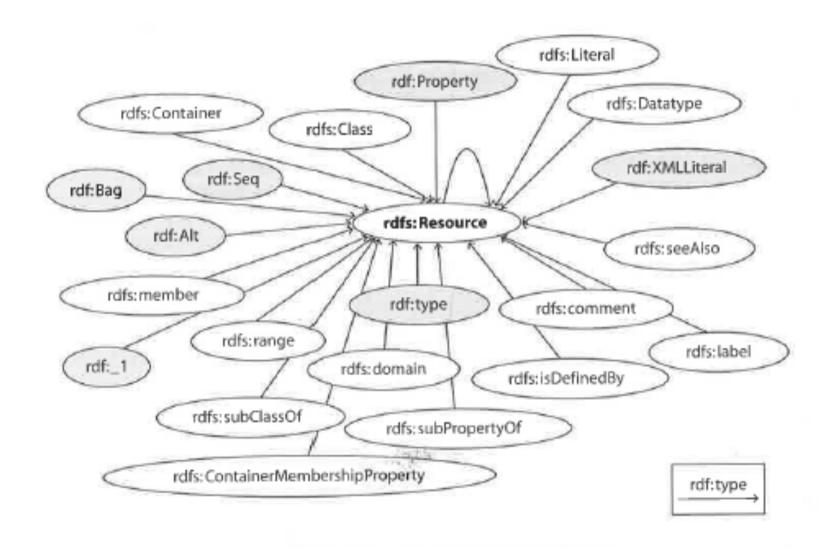


Le vocabulaire RDF Schema

- RDF Schema : vocabulaire standard pour écrire des ontologies légères
- Le vocabulaire RDFS se compose de 15 primitives décrites dans le langage RDF
- RDFS s'interroge en SPARQL



Le vocabulaire RDF Schema





Nommage de classe et nommage de propriété

- Une classe est désignée par un URI décrit comme étant de type rdfs:Class (en utilisant la propriété rdf:type)
- Une propriété est désignée par un URI décrit comme étant de type rdf:Property.
- Par convention d'usage, les noms de classes commencent par une majuscule et ceux des propriétés par une minuscule.



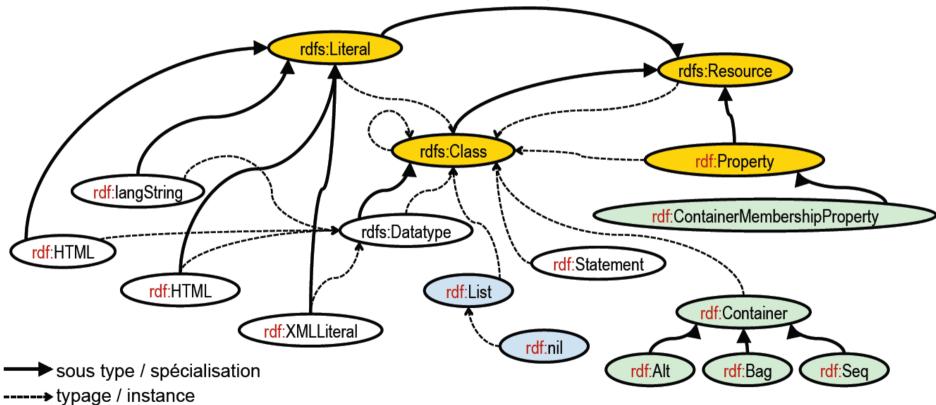
Le méta-modèle de RDFS

- Les primitives du langage RDFS sont définies dans les langages RDFS et RDF.
- Les classes rdfs:Resource et rdfs:Class sont les points d'amorce des graphes de déclaration des classes et des relations.
- Toutes les ressources sont des instances de la classe rdfs:Resource
- Toutes les primitives du langage sont des instances soit de la classe rdfs:Class, soit de la classe rdf:Property



Le méta-modèle RDFS : déclaration des classes

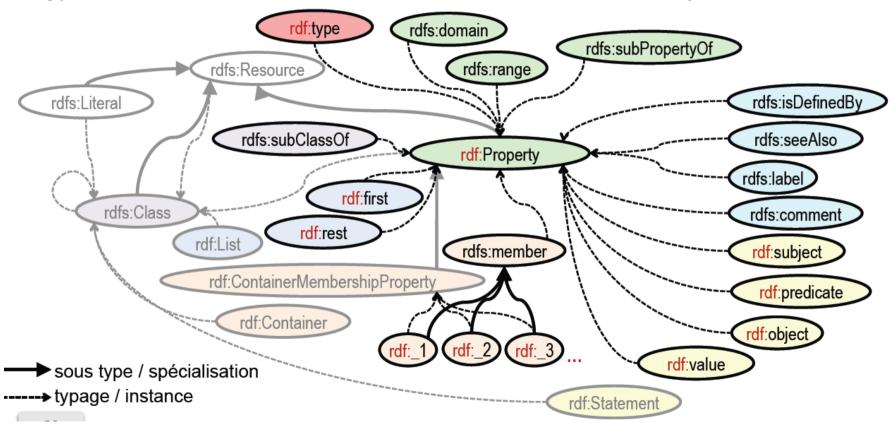
classes du vocabulaire RDFS et certains des liens qu'elles entretiennent





Le méta-modèle RDFS : déclaration des relations

les types de relations du vocabulaire RDFS et certains des liens qu'elles entretiennent





Sémantique

Tout est ressource

```
Sl x p y ALORS
x rdf:type rdfs:Resource
```

Slx p y ALORS
y rdf:type rdfs:Resource



Nommage de classes et propriétés

- Description de classe et de propriété
- Signature d'une propriété

rdfs:domain

rdfs:range

Labels:

rdfs:label

Commentaires

rdfs:comment

Réseaux sémantiques

rdfs:seeAlso



Description de classe : des instances de rdfs:class

la classe des classes est dans l'espace de nommage de RDFS.

```
<rdf:RDF xml:base="http://inria.fr/2005/humans.rdfs"
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> .
```



Description de propriété : des instances de rdf:property

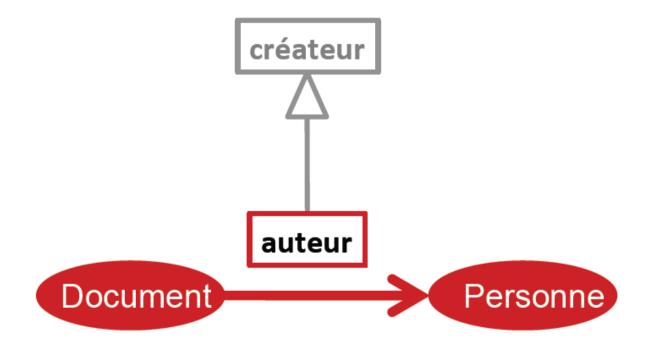
la classe des propriété est placée dans l'espace de nom RDF car les triplets sont une construction de RDF.

```
<rdf:RDF xml:base="http://inria.fr/2005/humans.rdfs"</pre>
 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> .
```



Signature d'une propriété

- classe de départ de la relation
- classe d'arrivée de la relation





Domaine (domain) et co-domaine/portée (range)

Classe de départ ou domaine: rdfs:domain Classe d'arrivée, co-domaine ou portée: rdfs:range

```
<rdf:RDF xml:base="http://inria.fr/2005/humans.rdfs"
   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">
    <rdf:Property rdf:ID="hasMother">
    <rdfs:subPropertyOf rdf:resource="#hasParent"/>
    <rdfs:domain rdf:resource="#Human"/>
    <rdfs:range rdf:resource="#Woman"/>
    </rdf:Property>
</rdf:RDF>
```



Labels textuels attachés aux ressources

toute ressource peut avoir un ou plusieurs labels dans une ou plusieurs langues



rdfs:comment & rdfs:seeAlso

les commentaires fournissent des définitions et explications en langue naturelle

```
@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 .
```

invitation à aller voir une autre ressource

```
<rdf:RDF xml:base="http://inria.fr/2005/humans.rdfs"
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>
```



Sémantique de RDFS

- Si (c₂, subClassOf, c₁) et (x, type, c₂)
 alors (x, type, c₁)
 Exemple: (Mehdi, type, Homme) ⇒ (Mehdi, type, Humain)
- Si (p₂, subPropertyOf, p₁) et (x, p₂, y)
 alors (x, p₁, y)
 Exemple: (Mehdi, auteur, Note) ⇒ (Mehdi, créateur, Note)
- Si (c₃, subClassOf, c₂) et (c₂, subClassOf, c₁) alors (c₃, subClassOf, c₁) *Transitivité*
- Si (p₃, subPropertyOf, p₂) et (p₂, subPropertyOf, p₁) alors (p₃, subPropertyOf, p₁) Transitivité
- subClassOf et subPropertyOf sont réflexives



Sémantique de RDFS

```
    Si (p, range, c) et (x, p, y)
    alors (y, type, c)
    Exemple : (aPourMere, range, Femme)
    (Mehdi, aPourMere, Cécile)
    ⇒ (Cécile, type, Femme)
```

Règles de la sémantique de RDFS :

http://www.w3.org/TR/rdf-mt/#rules



RDFS: Réutiliser ou lier des vocabulaires

- Dublin Core : Schéma de métadonnées le plus connu, mondialement utilisé pour décrire des ressources numériques (15 descriptions traduits en RDF)
- FOAF: vocabulaire permettant de décrire des personnes, les relations entre elles, leurs activités (photos, calendriers, etc.).
 Vocabulaire RDF composé de 8 classes et 36 propriétés.
- FOAF utilise le Dublin Core.

Foaf:Agent de FOAF est équivalent à dc:Agent du Dublin Core

Foaf:maker de FOAF est équivalent à dc:creator de Dublin Core



Utilisation et référence aux schémas

```
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
<rdf:RDF
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:foaf="http://xmlns.com/foaf/0.1/"
      xmins: ac="nttp://puri.org/ac/elements/i.i/
   foaf:Prson rdf:nodeID="fabien gandon">
    <foaf:title>Dr</foaf:title>
    <foaf: firstName>Fabien</foaf:firstName>
    <foaf:surname>Gandon</foaf:surname>
    <foaf: hick>Fab</foaf:nick>
    <foaf:gender>male</foaf:gender>
    <foaf: box rdf:resource="mailto:Fabien.Gandon@sophia.inria.fr" />
    <foaf:homepage
         rdf:resource="http://www-
  sop.inria.fr/acacia/personnel/Fabien.Gandon/"/>
    <foaf:img>
      <foaf:Image rdf:about="http://www-sop.inria.fr/acacia/personnel/</pre>
                          Fabien.Gandon/resources/images/me small.gif">
        <dc title>Fabien Gandon</dc:title>
        <dc description>Picture of Fabien in 2004</dc:description>
        <dc format>image/gif</dc:format>
      </ri></ri></ri></ri>
    </foaf:img>
```



Exemple: conception d'un nouveau vocabulaire

- Le vocabulaire Relationship, qui permet de décrire la nature des relations entre personnes, réutilise le vocabulaire FOAF
- Ex : la propriété http://purl.org/vocab/relationship/friendOf spécialise la propriété foaf:knows et a pour domaine et range la classe foaf:Person

```
http://vocab.org/relationship/.rdf

1. <rdf:Description
2. rdf:about="http://purl.org/vocab/relationship/friendOf">
3. <rdfs:subPropertyOf
4. rdf:resource="http://xmlns.com/foaf/0.1/knows"/>
5. <rdfs:label xml:lang="en">Friend Of</rdfs:label>
6. <rdfs:domain rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
7. <rdfs:range rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
8. <rdfs:isDefinedBy</pre>
```

rdf:resource-"http://purl.org/vocab/relationship/"/>

Listing 3.2 - Extrait du vocabulaire Relationship



10. </rdf:Description>

OWL pour les ontologies lourdes

Recommandation du W3C depuis 2004



Le gâteau du web sémantique ...

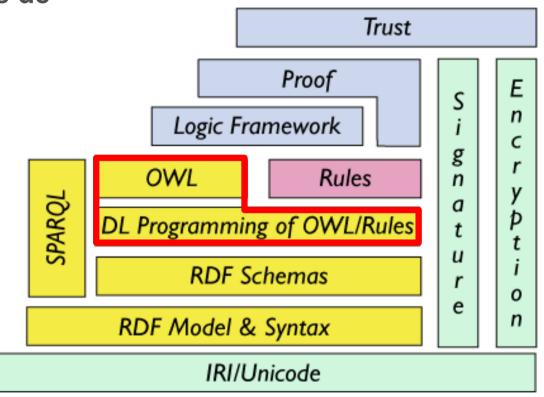
OWL représenté au dessus de

RDFS

OWL Lite

OWL DL

OWL Full



Ontologies

W3C,

T Berners-Lee, Ivan Herman

OWL: Ontologies lourdes

OWL plus expressif que RDFS

OWL Lite / DL / Full

Fondé sur les logiques de description

- Définition de classes par énumération, union, intersection, complément, disjonction, restriction valeur et cardinalité des propriétés
- Caractérisation des propriétés (symétrique, transitive, fonctionnelle, inversement fonctionnelle, inverse)
- Gestion des équivalences, versions, documenter





OWL: Ontologies lourdes

- Fournit des primitives supplémentaires pour des ontologies plus complexes
- Permet des définitions plus riches des classes et des propriétés
- Permet de tirer plus de conclusions, de faire plus d'inférences





OWL

- Basé sur les Logiques de Description (LD)
- Issu des projets de recherche américain DAML (Darpa Agent Markup Language) et européen OIL (Ontology Inference Layer)
- 3 couches selon les constructeurs considérés :

OWL Lite : facile à implémenter, plus expressif que RDFS

OWL DL : basé sur une logique de description expressive et décidable

OWL Full : complètement compatible avec la sémantique de RDFS



De RDFS à OWL : l'influence DL

- OWL DL signifie OWL Description Logic
- Logiques de description : Concept / Rôle / Individu

Deux niveaux distincts:

- *niveau terminologique* : représentation et manipulation des concepts et des rôles (TBox) : subsomption, hiérarchies de concepts et de rôles
- *niveau factuel / assertionnel* : description et manipulation des individus (ABox)
- Parallèle : Concept ↔ Classe & Rôle ↔ Propriété



Opérations dans les Logiques de Description

- Test de subsomption : vérifier qu'un concept en subsume un autre (utile pour valider une classification)
- Classification : placer un concept ou un rôle dans la hiérarchie.
 (assistance à la construction et l'évolution des ontologies)
- Test de satisfiabilité : vérifier qu'un concept admet des instances (utile pour vérifier la cohérence)
- Identification : retrouver les concepts les plus spécifiques dont un individu est susceptible d'être une instance.



Description de OWL

- Définition de classe
- Définition de propriété
- La famille OWL



Définition de classe

- Une définition de classe repose sur une description de classe
- Il existe 6 types de description de classe :
 - Nommage d'une classe
 - Enumération des instances d'une classe
 - Intersection de descriptions de classe
 - Union de descriptions de classe
 - Complémentaire d'une description de classe
 - Restriction de propriétés pour la classe décrite



Nommage d'une classe

- Le nommage d'une classe est son identification par un URI typé comme étant une instance owl:Class qui désigne la classe des classes
- Deux classes sont prédéfinies en OWL :

owl:Thing et owl:Nothing



Enumération des instances d'une classe

 Définition en extension d'une classe i.e. en énumérant tous ses membres



Union et intersection de classes

Définition d'une classe par union de classes

 Définition complète d'une classe par intersection d'autres classes



Complément et disjonction de classes

Définition d'une classe complémentaire

```
<owl:Class rdf:ID="Male">
        <owl:complementOf rdf:resource="#Female"/>
</owl:Class>
```

Imposer une disjonction

```
<owl:Class rdf:ID="Carre">
     <owl:disjointWith rdf:resource="#Rond"/>
</owl:Class>
```

Contraindre toutes les valeurs



Restriction sur valeur des propriétés

Contraindre au moins une valeur

Imposer une valeur exacte

Restriction sur la cardinalité

- Cardinalité d'une propriété : nombres d'instances différentes d'une propriété i.e. nombres de fois où une même ressource est utilisée comme point de départ (domain) d'une propriété
- Contraintes: nb minimum, nb maximum, nb exact



Définition de propriétés

- Une définition de propriété, appelé aussi axiome de propriété, repose sur une description de propriétés
- Il existe trois types de propriété :
 - •Les ObjectProperty sont des relations entre les ressources uniquement.
 - ex: aPourParent(#thomas,#stéphane)
 - Les DatatypeProperty ont pour valeur un littéral possiblement typé
 ex: aPourNom(#thomas,"Thomas")
 - •Les AnnotationProperty sont ignorées dans les inférences, uniquement utilisées pour documenter ou pour des extensions hors des inférences DL



Caractérisation des propriétés

Propriété symétrique, xRy ⇒ yRx

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

Propriété transitive, xRy & yRz ⇒ xRz:

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

Propriété fonctionnelle, xRy & xRz ⇒ y=z

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

Propriété inversement fonctionnelle, xRy & zRy ⇒ x=z

```
<owl:InverseFunctionalProperty rdf:ID="NumSSociale" />
```

Deux propriétés inverses, xR1y <=> yR2x

```
<rdf:Property rdf:ID="hasChild">
  <owl:inverseOf rdf:resource="#hasParent"/>
  </rdf:Property>
```



Relations d'équivalence entre propriétés & Gestion de l'ontologie

- Classes équivalentes : owl:equivalentClass
- Propriétés équivalentes : owl:equivalentProperty
- Instances identiques ou différentes: owl:sameAs, owl:differentFrom
- Utilité dans la mise en correspondance d'ontologies :

```
<owl:Class rdf:about="&o1;Person">
  <owl:equivalentClass rdf:resource="&o2;Hito"/>
  </owl:Class>
```

- **Description de l'ontologie :** owl:Ontology, owl:imports, owl:versionInfo, owl:priorVersion, owl:backwardCompatibleWith, owl:incompatibleWith
- Versions des classes et des propriétés: owl:DeprecatedClass, owl:DeprecatedProperty



Description de l'ontologie

```
une classe (owl:Ontology) et plusieurs propriétés (owl:imports,
owl:versionInfo, owl:priorVersion, owl:backwardCompatibleWith,
owl:incompatibleWith)
<rdf:RDF xml:base="http://inria.fr/2005/humans/"
 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/owl#">
<owl:Ontology rdf:about="http://inria.fr/2005/humans/">
 <rdfs:comment>An example OWL ontology</rdfs:comment>
 <owl:priorVersion</pre>
   rdf:resource="http://inria.fr/2004/humans/"/>
 <owl:imports rdf:resource="http://cnrs.fr/animals/"/>
 <rdfs:label>Bio Ontology</rdfs:label>
</owl:Ontology>
</rdf:RDF>
```



OWL : une famille de langages avec des niveaux d'expressivité

- OWL Full contient tout ce que l'on a mentionné précédemment mais OWL Full n'est pas décidable
- OWL DL (Description Logic) est une première restriction qui permet un maximum d'expressivité tout en assurant la décidabilité des algorithmes d'inférence
- OWL Lite permet l'expression de contraintes simples pour lesquelles les algorithmes d'inférence sont décidables

