



Du Web aux ontologies

Introduction aux langages du Web sémantique

École d'été Interdisciplinaire en
Numérique de la Santé

28 mai 2025 - Paul Fabry, Adrien Barton

Objectifs

- Connaître l'histoire et le contexte du Web sémantique et des ontologies
- Comment représenter des (méta)données en RDF/RDFS
- Comment créer des requêtes en SPARQL pour interroger ces données

Plan

- Historique et contexte
- Décrire et structurer des données: RDF, RDFS
- Interroger ces données: SPARQL
- Application pratique

Historique et contexte

Il était une fois le World Wide Web...

- Le Web traditionnel relie des documents par des hyperliens
 - 1945: MEMory Extender (MEMEX) de V. Bush
 - 1965: Definition de « hypertexte » par T. Nelson
 - 1990-92: Création du HTML par T. Berners-Lee
 - 1994: Création du W3C (World Wide Web Consortium)
- Révolution de l'accès aux données
 - Distribuées sur internet, interconnectées, plateformes-indépendantes
 - d'abord largement textuelles, avec toujours plus de multimédia

Historique et contexte

Mais ces données ne sont pas pour les machines

- Les données du Web sont:
 - Peu ou pas structurées, dépendantes du contexte
 - Pas de standard commun pour exprimer le sens
 - Ex. « Paris » c'est une ville, une personne (réelle ou imaginaire), un film, une série, un livre, une pièce de musique, un pseudonyme ?
- Les données nécessitent une interprétation humaine pour:
 - Recherche
 - Agrégation
 - Comparaison, raisonnement, etc.

Historique et contexte

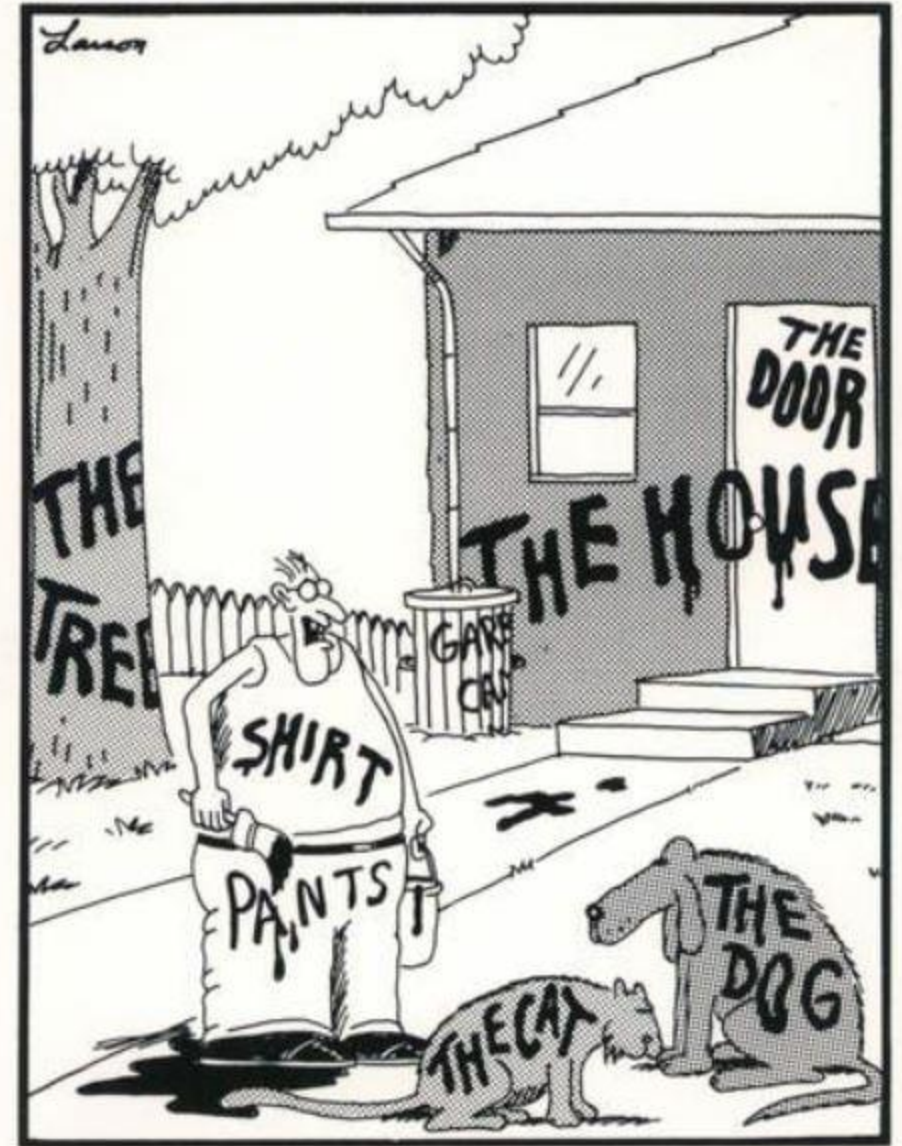
Arrive le Web sémantique

« Sémantique » : les machines doivent aussi « comprendre » les données



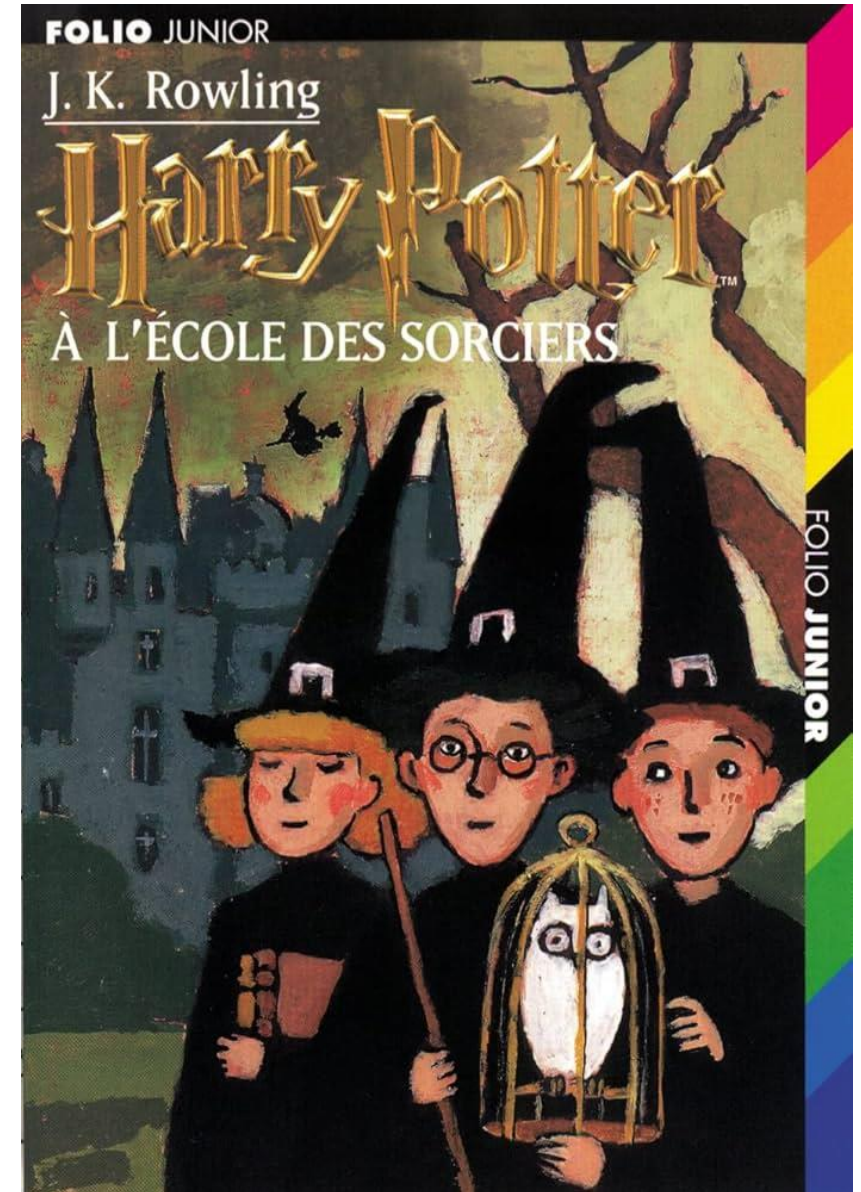
Ajouter du sens aux données:

Métadonnées (données sur les données)



"Now! . . . That should clear up a few things around here!"

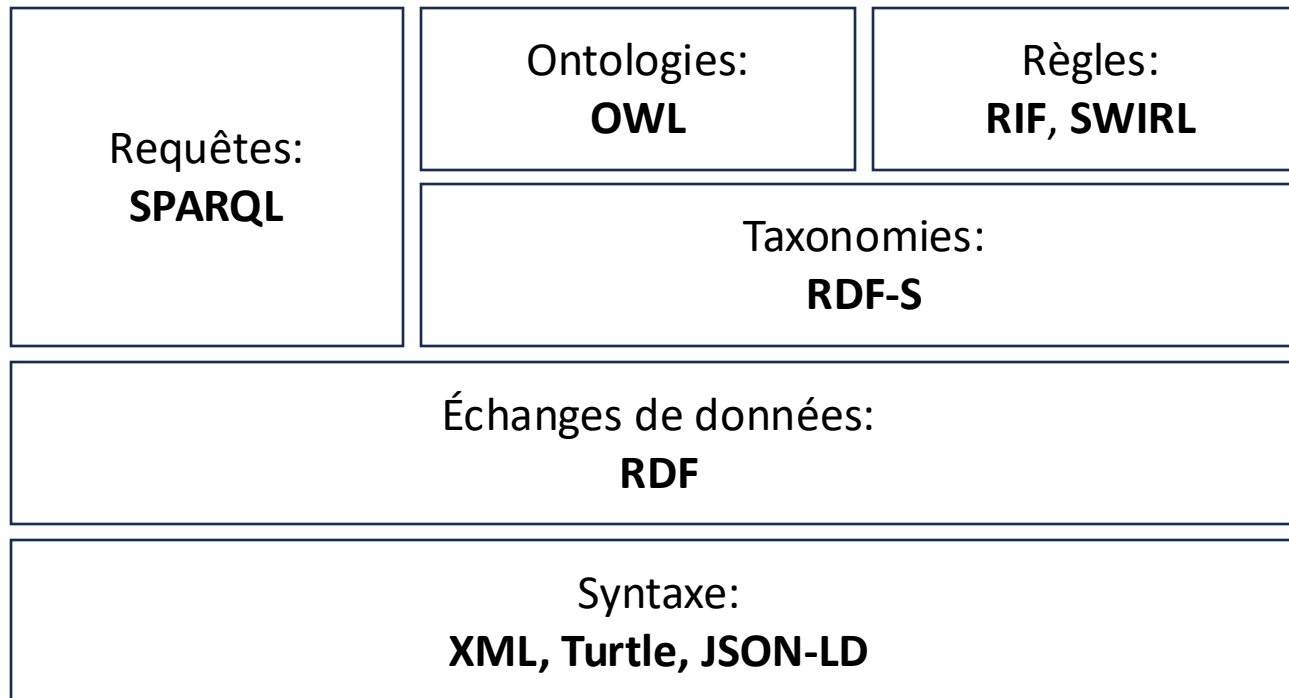
Exercice: Métadonnées



Historique et contexte

Arrive le Web sémantique

- Des langages pour exprimer cette sémantique: RDF, RDF-S, OWL et SPARQL



RDF

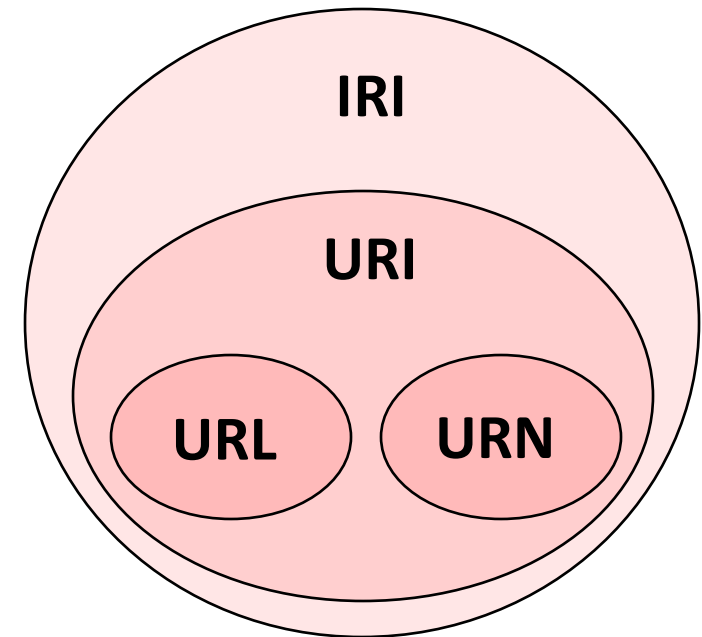
Au commencement, il y a la ressource

- Resource Description Framework (RDF)
- RDF décrit de façon formelle et explicite (sous forme de graphes) des ressources
- C'est quoi une ressource?
 - À l'origine: documents électroniques (pages web)
 - Toute entité qui peut être nommée dans un système d'information
 - Ressource abstraite: à l'extérieur d'un système d'information (individu, objet, etc.)

RDF

Au commencement, il y a la ressource

- Pour manipuler une ressource, il faut pouvoir l'identifier:
 - URL: Uniform Resource Locator
 - Emplacement d'une ressource
 - URN: Uniform Resource Name
 - Nom d'une ressource
 - URI: Uniform Resource Identifier
 - Identifiant d'une ressource (ASCII)
 - IRI: Internationalized Resource Identifier
 - Identifiant d'une ressource (UTF)



RDF

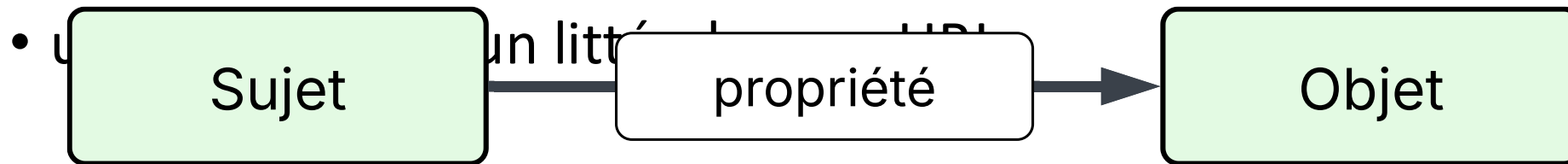
Au commencement, il y a la ressource

- Ne pas confondre la ressource avec son identifiant! (use-mention)
- Ce n'est pas <https://griis.ca/a-propos/equipe/paul-fabry/> qui fait cette présentation

RDF

Modèle - triplets

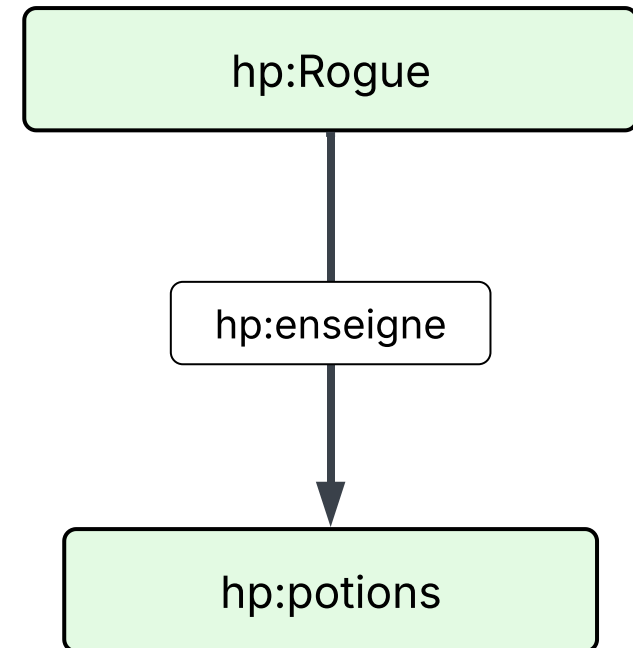
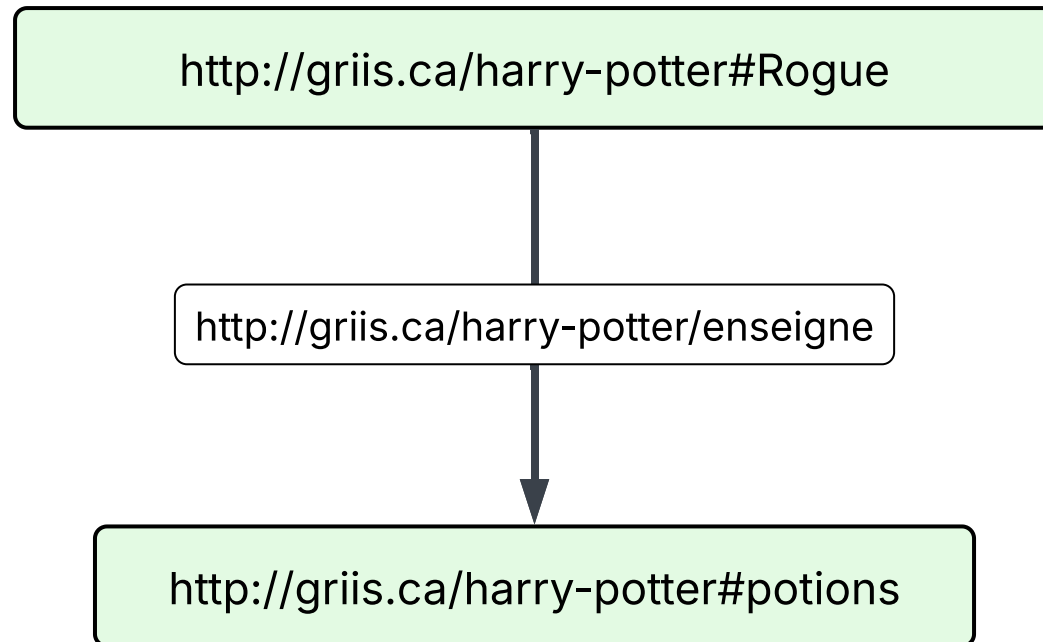
- Triplet RDF =
 - un **sujet** qui est un URI qui identifie une ressource (entité du domaine)
 - Une **propriété** (prédicat) qui est une relation binaire sur le domaine entre un sujet et un objet, identifiée par un URI.



RDF

Modèle - préfixes

- Pour simplifier les notations, on définit des raccourcis (préfixes) pour les URI :
 - <http://griis.ca/harry-potter/#> → hp:



RDF

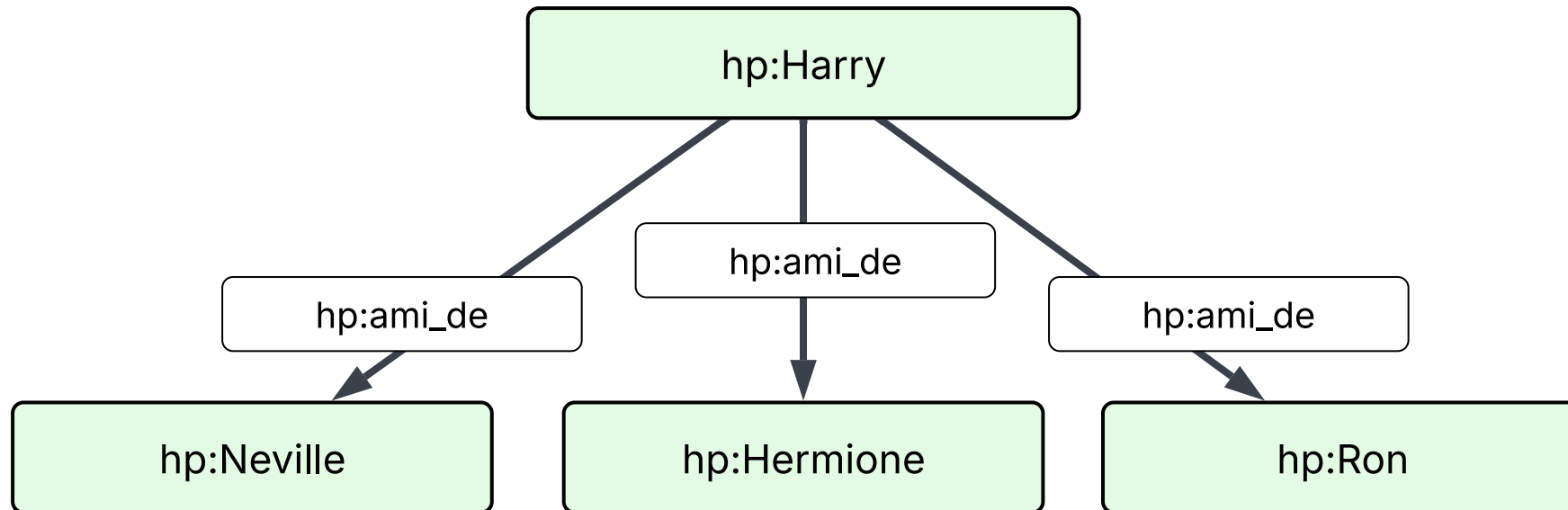
Modèle - préfixes

- Nombreux vocabulaire de métadonnées disponibles
 - Friend of a friend = <http://xmlns.com/foaf/0.1/> → foaf: (foaf:name, foaf:person)
 - Dublin Core = <http://purl.org/dc/elements/1.1/> → dc: (dc:creator, dc:date)
 - RDF = <http://www.w3.org/1999/02/22-rdf-syntax-ns#> → rdf: (rdf:type)
 - RDFS = <http://www.w3.org/2000/01/rdf-schema#> → rdfs: (rdfs:label, rdfs:subClassOf)
 - XML Schema Definition = <http://www.w3.org/2001/XMLSchema#> → xsd: (xsd:string, xsd:int)

RDF

Modèle - Propriétés

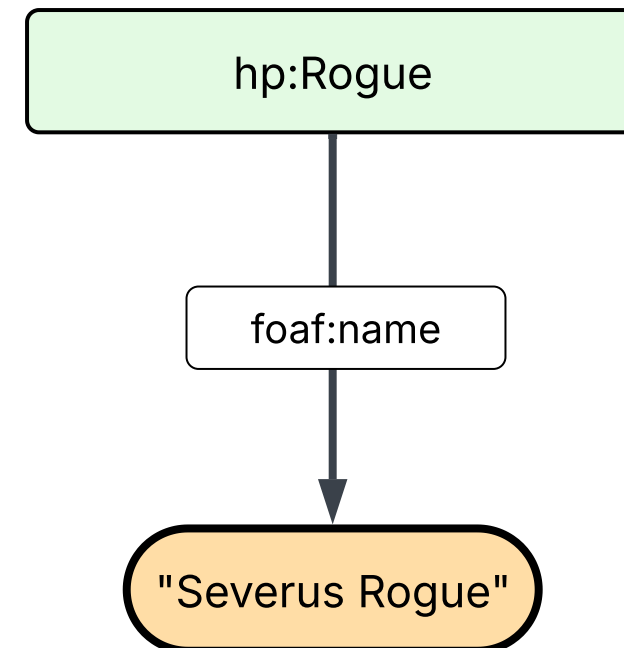
- Une ressource peut avoir plusieurs valeurs pour la même propriété



RDF

Modèle - Propriétés

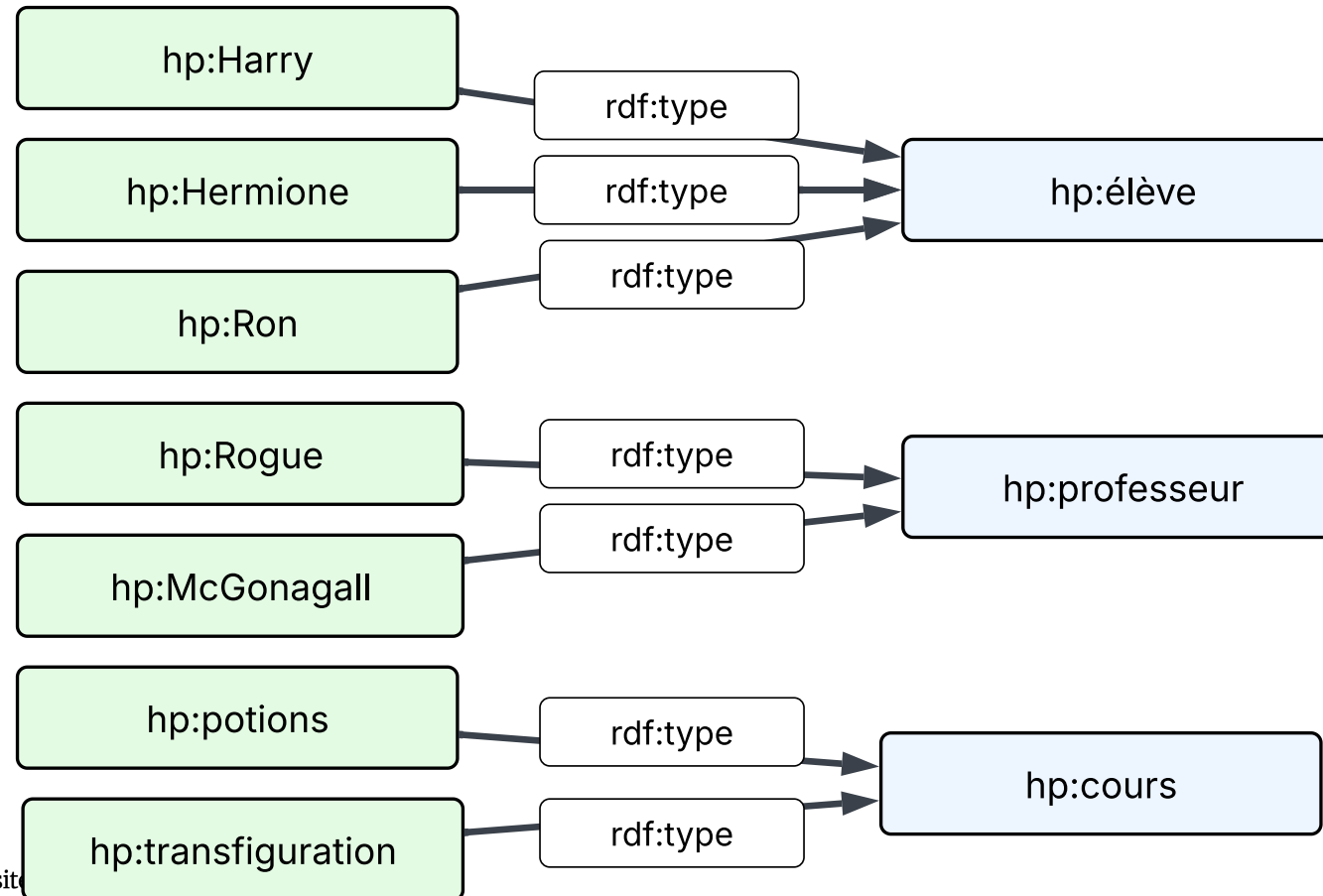
- On peut lier une ressource à une donnée typée (chaîne de caractères, entier, réel...), nommée un **littéral**.
- Mais l'inverse n'est pas possible



RDF

Modèle - Catégories

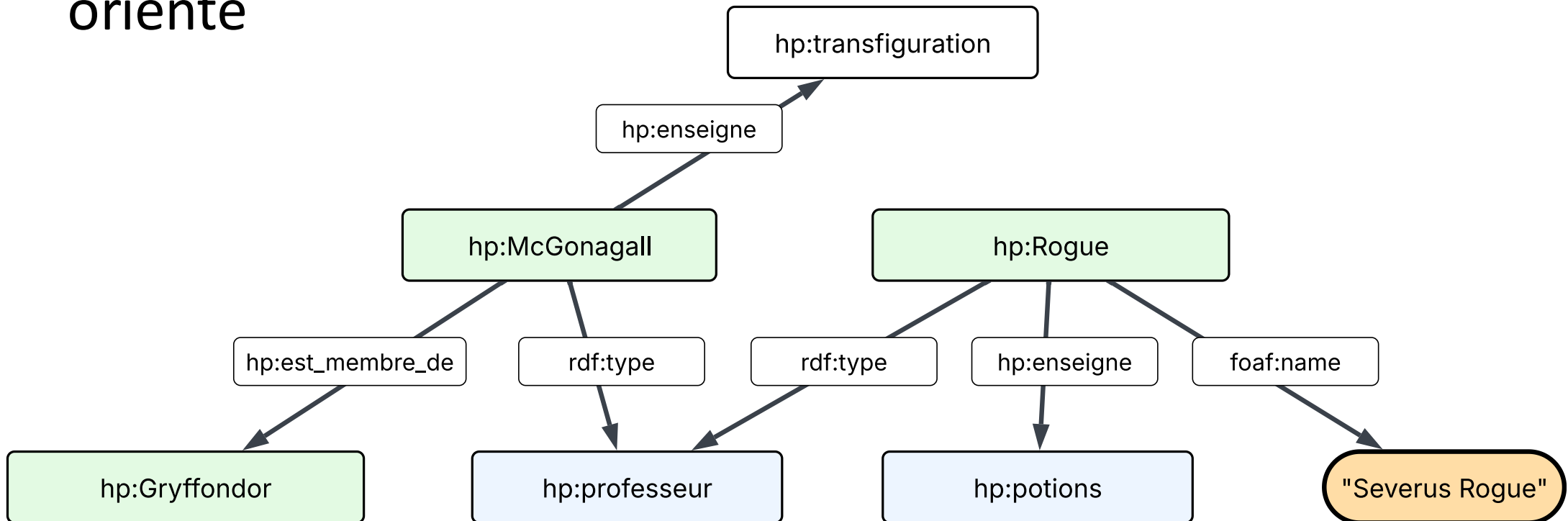
- On peut organiser les ressources en catégories (rdf:type)



RDF

Document

- Un document RDF est un ensemble de triplet
- Un document RDF peut être représenté par un graphe orienté



RDF

Syntaxe

- Plusieurs syntaxes disponibles:
 - Historiquement XML:

```
<rdf:Description rdf:about="http://griis.ca/harry-potter#Harry">
  <rdf:type rdf:resource="http://griis.ca/harry-potter#élève"/>
  <foaf:name>Harry Potter</foaf:name>
  <hp:est_membre_de rdf:resource="http://griis.ca/harry-potter#Gryffondor"/>
  <hp:ami_de rdf:resource="http://griis.ca/harry-potter#Ron"/>
  <hp:ami_de rdf:resource="http://griis.ca/harry-potter#Hermione"/>
  <hp:ennemi_de rdf:resource="http://griis.ca/harry-potter#Drago"/>
</rdf:Description>
```

- Mais aussi Turtle:

```
hp:Harry a hp:élève ;
    foaf:name "Harry Potter" ;
    hp:est_membre_de hp:Gryffondor ;
    hp:ami_de hp:Ron, hp:Hermione ;
    hp:ennemi_de hp:Drago .
```

RDF

Limites

- Avec RDF on décrit des instances, on ne peut pas exprimer des généralités taxonomiques et non taxonomiques:
- **Je peux dire:**
 - Harry Potter est un élève de Poudlard
 - Harry Potter est un sorcier
- **Je ne peux pas dire:**
 - Tous les élèves de Poudlard sont des sorciers (taxonomique)
 - Tous les élèves suivent au moins un cours (non taxonomique)

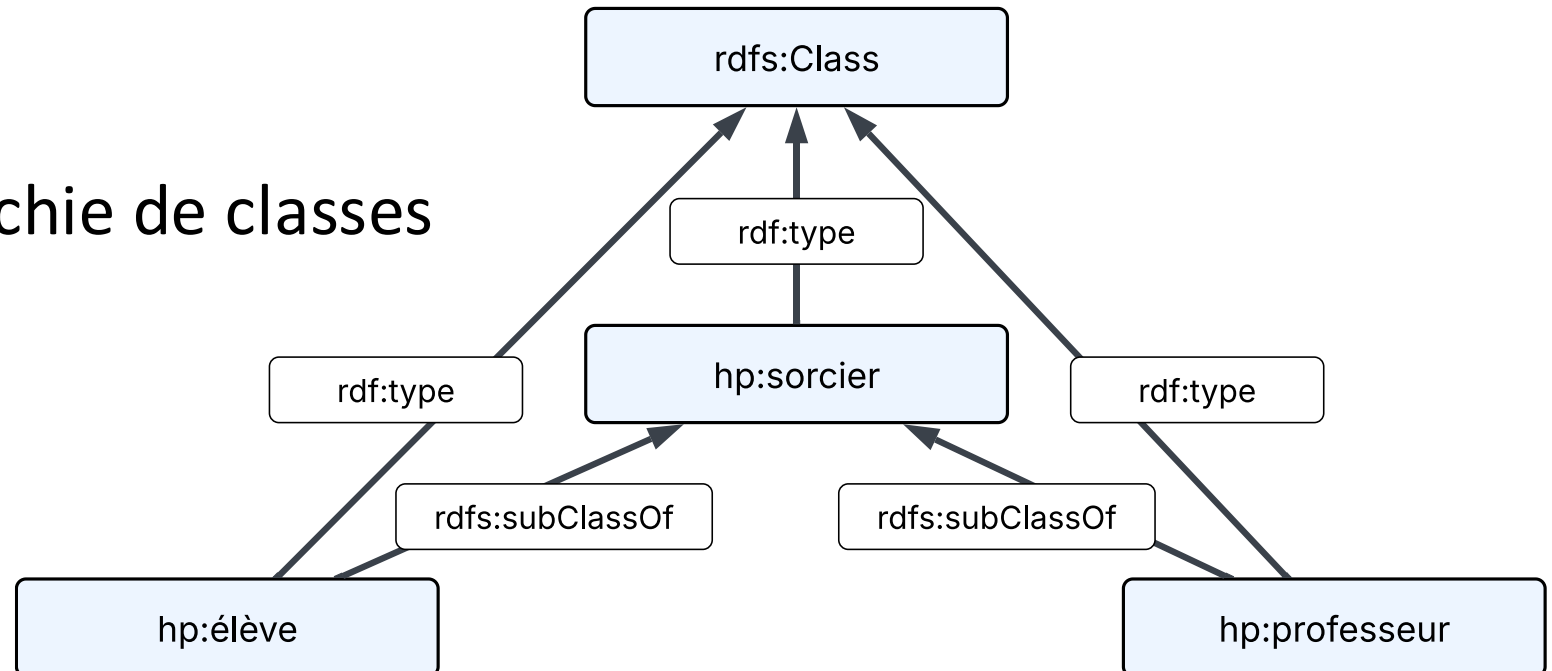
RDFS

- RDFS étend le modèle RDF, en permettant de définir :
 - des classes et relation de sous-classe
 - le typage des propriétés (domaine et co-domaine)
 - une relation de sous-propriété
- RDFS: standard W3C
 - premier langage pour représenter des ontologies dites « légères », en utilisant un standard déployé à l'échelle du Web
 - expressivité réduite et inférence simple
 - permet d'assurer l'interopérabilité entre les systèmes qui voudraient échanger de telles ontologies, de tels schémas.

RDFS

Classes et relation de sous-classe

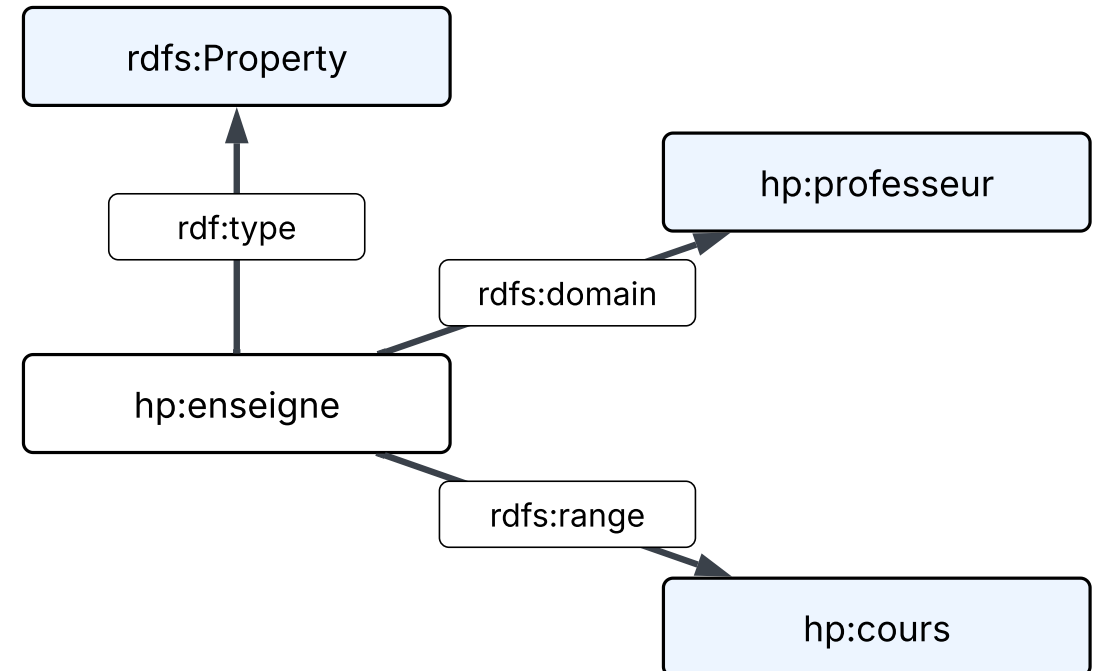
- `rdfs:Class`
 - C'est un ensemble d'éléments (ressources)
- `rdfs:subClassOf`
 - Définit une hiérarchie de classes



RDFS

Propriétés

- Une propriété est de type **rdfs:Property**
- On peut définir le type du sujet (domaine) et/ou l'objet (co-domaine) d'une propriété:
 - **rdfs:domain**: domaine de la propriété
 - **rdfs:range**: co-domaine de la propriété



RDFS

Inférence

- Des raisonneurs peuvent inférer de nouveaux triplets à partir :
 - des triplets existants,
 - des relations de sous-classe,
 - de sous-propriétés,
 - des renseignements sur les domaines et co-domaines.

RDFS

Limites

- Je ne peux pas:
 - Exprimer des relations entre classes (Un professeur enseigne au moins un cours)
 - Préciser la nature des relations entre ressources (symétrie, transitivité, etc.)
 - Exprimer la négation, la complémentarité, l'union, etc.
- Pour tout cela, et plus encore... OWL!

En résumé

- RDF décrit des données sous forme de triplets sujet-propriété-objet afin de faciliter leur interopérabilité
- RDFS est une extension de RDF qui introduit des notions de base (classes, propriétés, relation taxonomiques) pour structurer les données
- Mais pour exploiter RDF/RDFS il faut être capable de les interroger: SPARQL

SPARQL

Généralités

- SPARQL = SPARQL Protocol and RDF Query Language
- Langage standard du W3C pour interroger des données RDF/RDFS/OWL
- Similaire à SQL, mais pour des données en graphe

SPARQL

Structure d'une requête

- 4 parties
 - Déclaration des préfixes (optionnelle)
 - Liste des variables dont on souhaite obtenir les valeurs
 - Contraintes sur les variables
 - Tri/filtre sur les résultats (optionnelle)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX hp: <http://griis.ca/harry-potter/#>

SELECT ?sujet ?prop ?objet

WHERE {
    ?sujet ?prop ?objet .
}

ORDER BY ?sujet
```

Cette requête sort tous les triplets ordonnés alphabétiquement par le sujet

SPARQL

Variables

- Variables = ce que l'on cherche
 - Commence avec « ? »: `?variable`
- Contraintes sous forme de triplets sujet, propriété, objet avec:
 - Sujet: URI ou variable
 - `?variable rdf:type hp:cours`
 - Propriété: URI ou variable
 - `hp:Harry ?variable hp:Drago`
 - Objet: URI, variable ou littéral
 - `hp:Rogue foaf:name ?variable`
- Ensemble de triplets forme un patron de graphe


SPARQL

Exercice – Lancer le « endpoint »

- Pré-requis:
 - Installer Python
 - Installer rdflib-endpoint (pip install rdflib-endpoint)
 - Récupérer le fichier « harry_potter.ttl »
- Ouvrir un terminal dans le dossier où se trouve le fichier « harry_potter.ttl »
- Executer la commande:

```
rdflib-endpoint serve harry_potter.ttl
```
- Ouvrir le navigateur à l'adresse: <http://localhost:8000>

Query × +



1

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

2

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

3

4

5

SELECT ?s ?o ?p

6

7

WHERE {

8


?s ?o ?p .


9

}

10


11

 Table

 Response

181 results in 0.019 seconds

s

 0

SPARQL

Exercice 1

- Qu'est-ce qu'il y a dans ce fichier?
 - (lister tous les triplets)

```
SELECT *
```

```
WHERE {
```

```
    ?sujet ?prop ?objet .
```

```
}
```

```
SELECT ?sujet ?prop ?objet
```

```
WHERE {
```

```
    ?sujet ?prop ?objet .
```

```
}
```


SPARQL

Exercice 1

- Qu'est-ce qu'il y a dans ce fichier?
- Les trier
 - (ORDER BY)
 - (LIMIT)

```
SELECT ?sujet ?prop ?objet  
  
WHERE {  
    ?sujet ?prop ?objet .  
}  
ORDER BY ?sujet
```

SPARQL

Exercice 1

- Qu'est-ce qu'il y a dans ce fichier?
- Les trier
- Afficher toutes les données à propos de Harry Potter
 - (hp:Harry)

```
SELECT ?sujet ?prop ?objet

WHERE {
    hp:Harry ?prop ?objet .
}
```

```
SELECT ?sujet ?prop ?objet

WHERE {
    ?sujet ?prop ?objet .
    FILTER (?sujet = hp:Harry)
}
```

SPARQL

Exercice 2

- Faire la liste des élèves (hp:élève)

SPARQL

Exercice 2

- Faire la liste des élèves (hp:élève)
- Rajouter dans la liste leurs noms (foaf:name)

SPARQL

Exercice 3

- Quel cours enseigne le Prof. McGonagall ?
 - (hp:enseigne, hp:McGonagall)

SPARQL

Exercice 3

- Quel cours enseigne le Prof. McGonagall ?
 - (hp:enseigne, hp:McGonagall)

SPARQL

Exercice 3

- Quel cours enseigne le Prof. McGonagall ?
- Quels sont les noms des professeurs qui ont enseigné le cours de Défense contre les Forces du Mal ?
 - (hp:DCLFDM, foaf:name)

SPARQL

Exercice 4

- Lister les noms des membres des maisons précédés par leurs noms de maisons (noms)
 - (hp:maison, rdfs:label, hp:est_membre_de, foaf:name)

SPARQL

Exercice 4 - BONUS

- Classer les noms de maison par nombre de membres

SPARQL

Exercice 5

- Lister tous les sorciers

SPARQL

- Lister tous les sorciers – Pourquoi pas ?

```
SELECT ?sorcier  
  
WHERE {  
    ?sorcier rdf:type hp:sorcier .  
}
```

- Limites de rdflib-endpoint: pas de raisonneur
- Prochaine étape: Protégé et les raisonneurs