



TD/TP n° 1
RDF/RDF Schema
Master Ingénierie Informatique
U.E. « Web sémantique (3.0) »

F. BERTRAND

Année universitaire 2020-2021

Concepts abordés :

- Le format RDF
- Le langage de modélisation RDF Schéma

1 Environnement de travail

Pour ce TP (et le suivant) nous utiliserons trois outils :

- le validateur en ligne de syntaxe RDF fourni par le consortium W3. Cet outil ne permet de valider **uniquement le format RDF/XML** et offre une visualisation du graphe ;
- l'outil RDF Translator permettant de convertir les différentes syntaxes de RDF (attention il ne fonctionne qu'avec certains navigateurs : Chrome, Safari, IE).
- l'outil CORESE à la fois comme validateur de fichiers RDF (si le chargement réussi cela signifie que le fichier est valide) et comme un interpréteur de requêtes SPARQL . Cet outil est développé par l'INRIA de Sophia-Antipolis et il est téléchargeable ici.

Son lancement s'effectue avec la commande suivante :

```
java -jar nomDuFichierCorese.jar
```

Son utilisation est simple avec un onglet dédié au chargement de fichier et un autre pour l'interrogation SPARQL.

2 Une première description RDF

Cet exercice est issu du cours en ligne de France Université Numérique sur « Web sémantique et Web de données ».

Le but de cet exercice est de construire une petite description RDF. Il s'agit de représenter l'énoncé suivant :

Margot est une femme journaliste, âgée de 32 ans, mariée à Arthur qui est un homme avec qui elle a deux enfants, Marie qui est une femme et Simon qui est un homme. Pour chaque personne on spécifie aussi explicitement le nom.

2.1 Graphe à compléter

Commençons par analyser ce problème de modélisation dans la syntaxe abstraite du modèle RDF.

Compléter le graphe RDF présenté sur la figure 1 en utilisant les URI suivants : `:Arthur`, `:Simon`, `voc:name`, `voc:age`, `voc:hasChild`, `voc:hasSpouse`, `rdf:type`, `voc:Woman`, `voc:Man` et les littéraux suivants : Margot, Arthur, Marie et Simon.

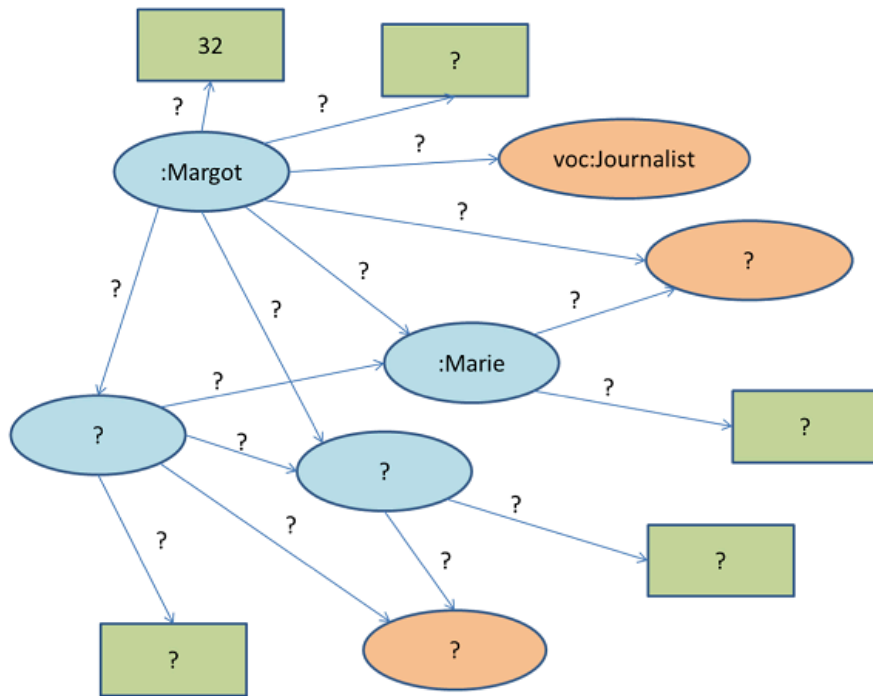


FIGURE 1 – Graphe RDF à compléter.

2.2 Sérialisation du graphe en RDF/XML

Ci-dessous le code en syntaxe RDF/XML qui formalise le graphe solution de l'exercice précédent.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY vocabulaire "http://www.unice.fr/voc">
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#"> ]>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:voc="&vocabulaire;"#
  xml:base="http://www.unice.fr/data">

  <AAA rdf:about="#Margot">
    <voc:name>Margot</voc:name>
    <voc:age rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">32</voc:age>
    <BBB rdf:resource="#Arthur"></BBB>
    <voc:hasChild rdf:resource="#Simon"></voc:hasChild>
    <voc:hasChild>
      <rdf:Description rdf:about="#Marie">
        <voc:name>Marie</voc:name>
        <rdf:type CCC="&vocabulaire;#Woman"></rdf:type>
      </rdf:Description>
    </voc:hasChild>
    <DDD rdf:resource="&vocabulaire;#Journalist"></DDD>
  </AAA>

  <EEE rdf:about="#Arthur">
    <voc:name>Arthur</voc:name>
    <voc:hasChild rdf:resource="#Simon"></voc:hasChild>
    <voc:hasChild rdf:resource="#Marie"></voc:hasChild>
  </EEE>

  <voc:Man rdf:about="#Simon">
    <voc:name>Simon</voc:name>
  </voc:Man>

</rdf:RDF>

```

Plusieurs éléments ont été supprimés et remplacés par les marques AAA, BBB, CCC, DDD et EEE. Complétez ces éléments manquants.

2.3 Sérialisation avec la syntaxe N3 (Turtle)

Ci-dessous le code en syntaxe N3 qui formalise le même énoncé que dans les deux exercices précédents.

```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix voc: <http://www.unice.fr/voc#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://www.unice.fr/data#Margot> a voc:Journalist AAA BBB ;
    voc:age "32"^^xsd:string ;
    voc:hasChild <http://www.unice.fr/data#Marie>,
        <CCC>;
    voc:hasSpouse <http://www.unice.fr/data#Arthur> ;
    voc:name "Margot" .

<http://www.unice.fr/data#Arthur> DDD voc:Man ;
    voc:hasChild <http://www.unice.fr/data#Marie>,
        <http://www.unice.fr/data#Simon> ;
    voc:name "Arthur" .

<http://www.unice.fr/data#Marie> a voc:Woman ;
    voc:name "Marie" .

<EEE> a FFF ;
    GGG HHH .

```

Là encore des éléments ont été supprimés et remplacés par les marques AAA, BBB, CCC, DDD, EEE, FFF, GGG et HHH.

2.4 Utilisation du validateur RDF du W3

Utilisez le validateur RDF du W3 avec le format RDF/XML et répondez aux questions suivantes :

1. Combien de triplets ont été analysés?
2. Combien de nœuds compte le graphe généré?

3 Création de graphes RDF

Soit les informations suivantes :

- « Le livre ayant l'URL <http://www.springerlink.com/content/978-3-540-10235-9/> à pour auteur Robin Milner »
- « Le site web de l'université de La Rochelle a pour administrateur Gaëlle Foret qui possède l'adresse électronique webmestre@univ-lr.fr. »

1. Écrire le (ou les) triplet(s) RDF exprimant l'information contenue dans chaque phrase sans utiliser de nœuds anonymes (ou blancs). En utilisant le support de cours, un guide sur la notation N3 et la spécification RDF/XML exprimer ces triplets selon ces deux formats. Puis pour tester la correction syntaxique de votre code RDF, utiliser :
 - le validateur du W3 pour le format RDF/XML (qui permet en plus de visualiser le graphe RDF);
 - CORESE pour le format RDF/N3.
2. Modifier les triplets précédents pour utiliser cette fois-ci au moins un nœud anonyme (donner les deux formats : N3 et RDF/XML) ;

4 Les schémas RDF

1. Créer un schéma RDF (en notation N3) à partir des spécifications suivantes :

- utiliser les classes suivantes : Institution, InstitutionRecherche, InstitutionEnseignement, Université, Personne, Etudiant, Enseignant, EtudiantMaster et EtudiantDoctorat;
- utiliser la propriété de subsomption (rdfs:subClassOf) pour relier les classes;
- définir des propriétés estEmployéPar et dirige (au sens où une personne peut diriger un ensemble de personnes).

Note : une solution est présentée sur l'image *ontologieUniv.png* (à récupérer sur Moodle) mais essayez d'abord de le faire vous-même.

2. Le schéma RDF suivant :

```
1 ex:ClasseA rdfs:type rdfs:Class
2 ex:ClasseB rdfs:subClassOf ex:ClasseA
3 ex:unB rdfs:type ex:ClasseB
```

peut être écrit en RDF/XML de la manière suivante :

```
1 <?xml version="1.0"?>
2 <!DOCTYPE rdf:RDF [
3   <!ENTITY ex "http://example.org#">
4   <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
5 ]>
6 <rdf:RDF
7   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
8   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
9   xmlns:ex="http://example.org#"
10
11   <rdf:Description rdf:about="&ex;ClasseA">
12     <rdf:type rdf:resource="&rdfs;Class"/>
13   </rdf:Description>
14
15   <rdf:Description rdf:about="&ex;ClasseB">
16     <rdf:type rdf:resource="&rdfs;Class"/>
17     <rdfs:subClassOf rdf:resource="&ex;ClasseA"/>
18   </rdf:Description>
19
20   <rdf:Description rdf:about="&ex;unB">
21     <rdf:type rdf:resource="&ex;ClasseB"/>
22   </rdf:Description>
23 </rdf:RDF>
```

ou de manière abrégée comme cela :

```
1 <?xml version="1.0"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
4   xmlns:ex="http://example.org#"
5   xml:base="http://example.org">
6   <rdfs:Class rdf:ID="ClasseA" />
7   <rdfs:Class rdf:ID="ClasseB" >
8     <rdfs:subClassOf rdf:resource="#ClasseA" />
9   </rdfs:Class>
10   <ex:ClasseB rdf:ID="unB" />
11 </rdf:RDF>
```

Donner la forme RDF/XML de schéma précédent (université) et vérifier sa validité.
Puis tester les classes définies dans le schéma via la requête SPARQL suivante :

```
PREFIX ex: <http://example.org#>
```

```
SELECT ?x
WHERE
{
  ?x rdfs:type rdfs:Class .
  FILTER (?x ^ ex: )
}
```

La fonction du filtre est de sélectionner uniquement les classes définies dans l'espace de noms `http://example.org` (l'opérateur `^` vérifie que le nom de la ressource débute par l'espace de noms).

- Étendre le schéma précédent en incluant un ensemble d'individus (instances) des classes définies au point précédent (enseignants, étudiants...). Pour la relation `dirige` une collection sera utilisée comme objet pour indiquer que l'on peut diriger un ensemble de personnes. Utiliser la notation `N3` qui est plus concise à écrire puis utiliser les convertisseurs *RDF Distiller* ou *RDF Translator* pour le convertir au format RDF/XML. À partir de la requête SPARQL précédente, écrire une nouvelle requête sélectionnant uniquement les individus.
- Vérifier les faits qui peuvent être déduits des données précédentes (cf. règles de déduction, figure 2) en vous inspirant de la requête SPARQL précédente.

RDFS entailment rules.

Rule Name	If E contains:	then add:
rdfs1	uuu aaa lll. where lll is a plain literal (with or without a language tag).	_:nnn rdfs:type rdfs:Literal . where _:nnn identifies a blank node allocated to lll by rule rule lg.
rdfs2	aaa rdfs:domain XXX . uuu aaa yyy .	uuu rdfs:type XXX .
rdfs3	aaa rdfs:range XXX . uuu aaa vvv .	vvv rdfs:type XXX .
rdfs4a	uuu aaa xxx .	uuu rdfs:type rdfs:Resource .
rdfs4b	uuu aaa vvv.	vvv rdfs:type rdfs:Resource .
rdfs5	uuu rdfs:subPropertyOf vvv . vvv rdfs:subPropertyOf xxx .	uuu rdfs:subPropertyOf xxx .
rdfs6	uuu rdfs:type rdfs:Property .	uuu rdfs:subPropertyOf uuu .
rdfs7	aaa rdfs:subPropertyOf bbb . uuu aaa yyy .	uuu bbb yyy .
rdfs8	uuu rdfs:type rdfs:Class .	uuu rdfs:subClassOf rdfs:Resource .
rdfs9	uuu rdfs:subClassOf xxx . vvv rdfs:type uuu .	vvv rdfs:type xxx .
rdfs10	uuu rdfs:type rdfs:Class .	uuu rdfs:subClassOf uuu .
rdfs11	uuu rdfs:subClassOf vvv . vvv rdfs:subClassOf xxx .	uuu rdfs:subClassOf xxx .
rdfs12	uuu rdfs:type rdfs:ContainerMembershipProperty .	uuu rdfs:subPropertyOf rdfs:member .
rdfs13	uuu rdfs:type rdfs:Datatype .	uuu rdfs:subClassOf rdfs:Literal .

FIGURE 2 – Règles de déduction de RDF Schéma

5 Référencement de produits

Un magasin offre les produits suivants :

```
1 <?xml version="1.1"?>
2
3 <products>
4   <book>
5     <title>Learning XML</title>
6     <author>EricRay</author>
7     <price>30.00</price>
8   </book>
9   <cd>
10    <title>for you</title>
11    <interpret>Frank Chastenier</interpret>
12    <price>15.00</price>
13  </cd>
14  <dvd>
15    <title>Goldrush</title>
16    <director>Charly Chaplin</director>
17    <price>19.00</price>
18  </dvd>
19 </products>
```

Représenter l'ensemble des produits avec RDF et un schéma RDF (le format peut être RDF/N3 ou RDF/XML) en décrivant le vocabulaire de manière aussi précise que possible. On souhaite notamment faire apparaître que le magasin offre une liste de produits. Pour les livres, utiliser la norme Dublin Core qui possède un schéma RDF.