

SPARQL

Comment illuminer vos mashups en consommant les données du Linked Data ?

SemWeb.pro, Paris, 3 mai 2012

RDF

Un rappel

Resource

d'une ressource

Description

de Description

Framework

Cadre/modèle

La grammaire du Web sémantique

Les **R** trois composants de

D

sont :

- les URI pour identifier ;
- les triplets pour **E**xprimer ;
- les graphes pour relier

RDF : les URIs (l'exemple)

1- Prenez des choses/concepts/entités du monde réel et placez les dans le monde numérique, en les nommant et en les identifiant par des URIs.

Signifié
dans le monde réel

Signifiant
dans le monde numérique

Tim Berners-Lee

<<http://www.w3.org/People/Berners-Lee/card#i>>

Une personne

<<http://xmlns.com/foaf/0.1/Person>>

L'article « Semantic Web » de 2001

<<http://www.sciam.com/article.cfm?id=the-semantic-web>>

est

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

Un texte

<<http://purl.org/dc/dcmitype/Text>>

Puisque ces choses/concepts/entités sont identifiées par une URI, ils sont assimilables à des ressources (RFC 3986)

RDF : le triplet (le principe)

Une série de signes reliés forme une donnée

Chien → Animal

doit comprendre
la donnée



Machine

RDF : le triplet (le principe)

une série de signes reliés forme une donnée

Chien → Animal

doit comprendre
la donnée



Machine

Ne comprend pas la donnée
mais qu'il s'agit d'un
paragraphe
la donnée

vraie
même en dehors de ce
document.

La donnée est encodée
dans le cadre d'un document

```
<html>
<head>
  <title>Le chien</title>
</head>
<body>
  <p>
    le chien est un animal
  </p>
</body>
</html>
```

De plus,
est toujours

RDF : le triplet (le principe)

une série de signes reliés forme une donnée

Chien → Animal

doit comprendre la donnée



Machine

Ne comprend pas la donnée mais qu'il s'agit d'un paragraphe

la donnée

De plus,

est toujours

vraie même en dehors de ce document

La donnée est encodée dans le cadre d'un document

```
<html>
<head>
  <title>Le chien</title>
</head>
<body>
  <p>
    le chien est un animal
  </p>
</body>
</html>
```

La machine peut traiter et analyser la donnée car elle est encodée selon une logique formelle.



La donnée elle-même est encodée sous la forme d'un triplet.

RDF : le triplet (le principe)

une série de signes reliés forme une donnée

Chien → Animal

doit comprendre la donnée

Machine



Ne comprend pas la donnée mais qu'il s'agit d'un paragraphe

la donnée

De plus,

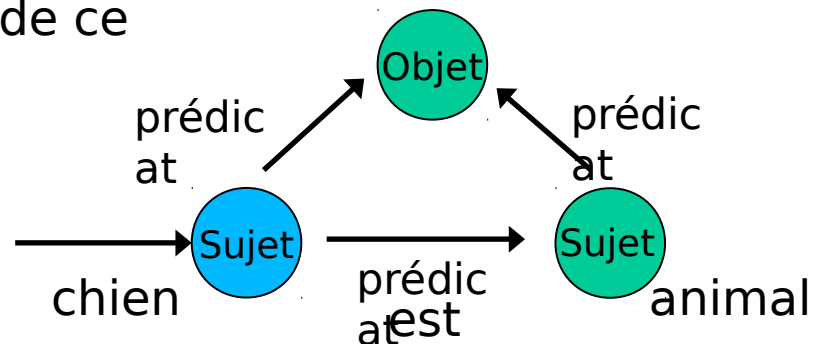
est toujours

vraie même en dehors de ce document

La donnée est encodée dans le cadre d'un document

```
<html>
<head>
  <title>Le chien</title>
</head>
<body>
  <p>
    le chien est un animal
  </p>
</body>
</html>
```

La machine peut traiter et analyser la donnée car elle est encodée selon une logique formelle.



La donnée elle-même est encodée sous la forme d'un triplet.



RDF, un modèle pour encoder les données structurées

RDF : le triplet (l'exemple)

2- Exprimez des relations entre ces ressources sous la forme de triplets ou comment décrire l'information à son niveau de granularité

le plus basique, la donnée ?
Structure d'un triplet RDF = Structure d'une phrase simple

Sujet Verbe Complément = (Sujet, Prédicat, Objet)

L'article « Semantic Web » de 2001 est un texte

(<http://www.sciam.com/article.cfm?id=the-semantic-web> ,
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type> ,
<http://purl.org/dc/dcmitype/Text>)

L'article Semantic Web de 2001 a pour créateur Tim Berners-Lee

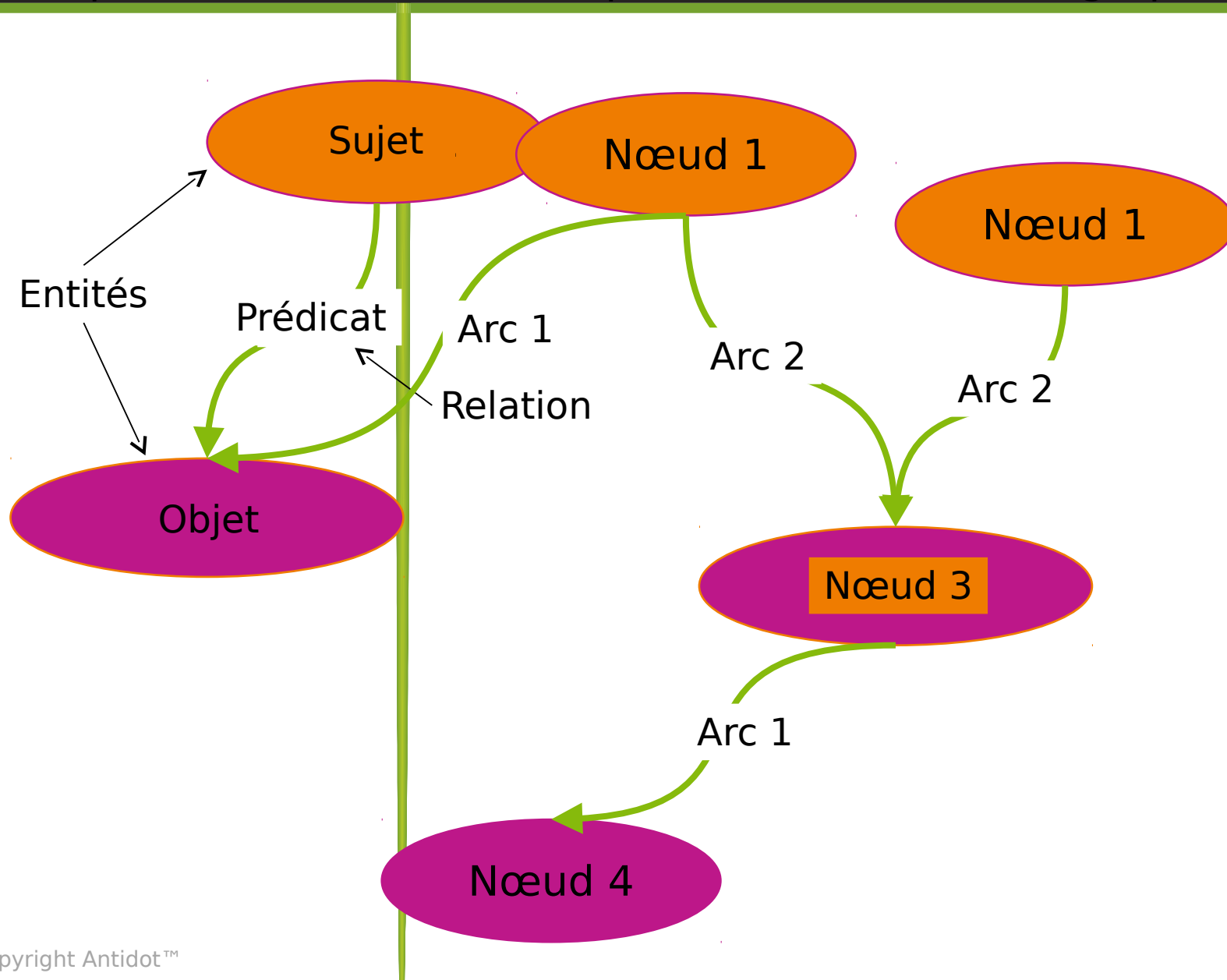
(<http://www.sciam.com/article.cfm?id=the-semantic-web> ,
<http://purl.org/dc/elements/1.1/creator> ,
<http://www.w3.org/People/Berners-Lee/card#i>)

Tim Berners-Lee est une personne

(<http://www.w3.org/People/Berners-Lee/card#i> ,
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type> ,
<http://xmlns.com/foaf/0.1/Person>)

RDF : le graphe orienté (le principe)

3- Représentez et reliez les triplets sous la forme de graphes orientés



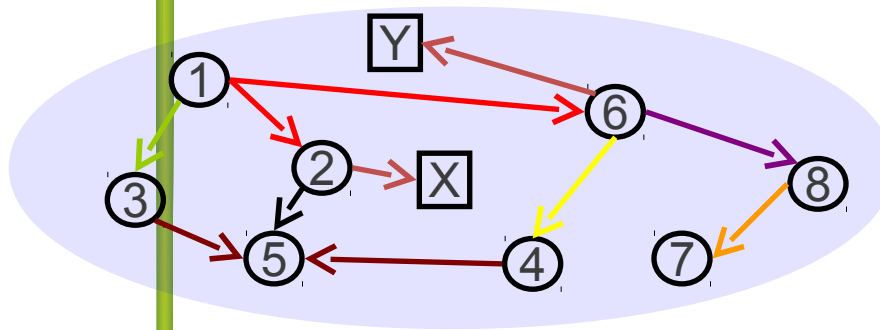
SPARQL

Principes généraux

Le parcours de graphe pas à pas : le graphe

SPARQL permet d'extraire des informations d'un graphe par l'expression de contraintes

sous la forme d'équations qui permettent de parcourir le graphe interrogé.
Soit le graphe suivant sous la forme d'une représentation graphique :

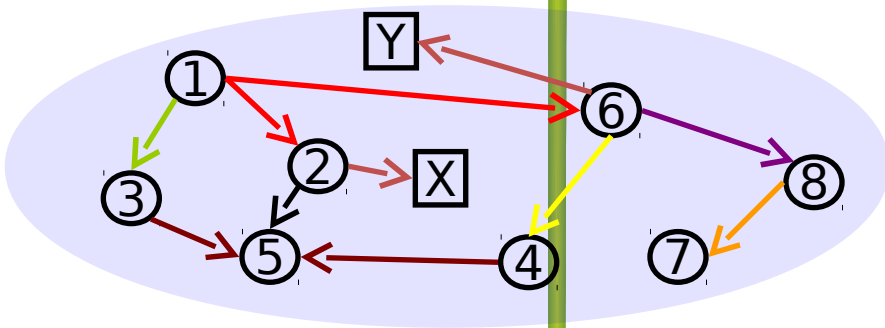


Le même graphe sérialisé en Turtle :

```
<1> <rouge> <2>.  
<1> <rouge> <6>.  
<1> <vert> <3>.  
<2> <bleu> "X".  
<2> <noir> <5>.  
<6> <bleu> "Y".
```

Le parcours de graphe pas à pas

Rechercher un prédicat



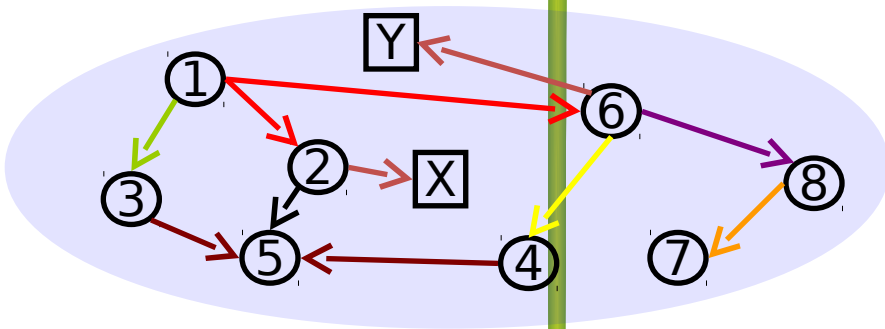
<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

Comment atteindre les ressources <2> et <6> à partir de <1> ?

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Le parcours de graphe pas à pas

Rechercher un prédicat



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

Comment atteindre les ressources <2> et <6> à partir de <1> ?

Reformuler la question

Sujet = <1>

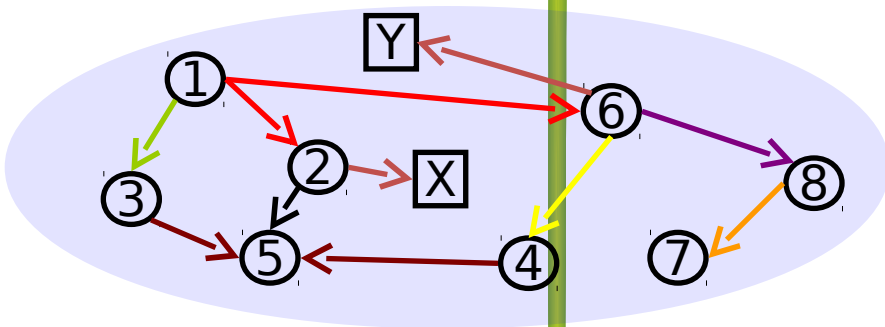
Prédicat = ?

Objet = <2> et <6>

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>
<3> <marron> <5>.

Le parcours de graphe pas à pas

Rechercher un prédicat



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

Comment atteindre les ressources <2> et <6> à partir de <1> ?

Reformuler la question

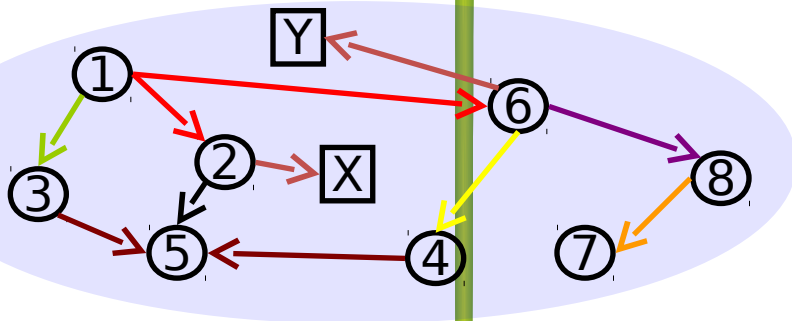
Sujet = <1>
Prédicat = ?
Objet = <2> et <6>

Sous forme de triplets

→ <6> <violet> <1> ? <2>.
<8> <orange> <1> ? <6>.
<6> <jaune> <4>.
<4> <marron> <5>
<3> <marron> <5>.

Le parcours de graphe pas à pas

Rechercher un prédicat



- <1> <rouge> <2>.
- <1> <rouge> <6>.
- <1> <vert> <3>.
- <2> <bleu> "X".
- <2> <noir> <5>.
- <6> <bleu> "Y".

Comment atteindre les ressources <2> et <6> à partir de <1> ?

Reformuler la question

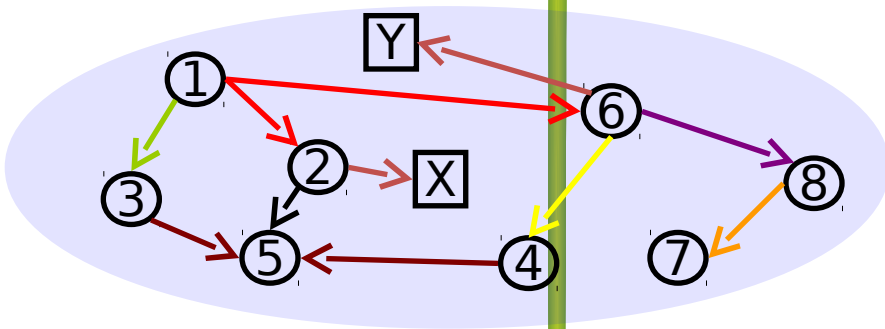
Sujet = <1>
Prédicat = ?
Objet = <2> et <6>

Sous forme de triplets

- <6> <violet> <1> ?predicat <2>.
- <8> <orange> <7>.
- <6> <jaune> <4>.
- <4> <marron> <5>
- <3> <marron> <5>.

Le parcours de graphe pas à pas

Rechercher un prédicat



- <1> <rouge> <2>.
- <1> <rouge> <6>.
- <1> <vert> <3>.
- <2> <bleu> "X".
- <2> <noir> <5>.
- <6> <bleu> "Y".

Comment atteindre les ressources <2> et <6> à partir de <1> ?

Reformuler la question

Sujet = <1>
Prédicat = ?
Objet = <2> et <6>

Sous forme de triplets

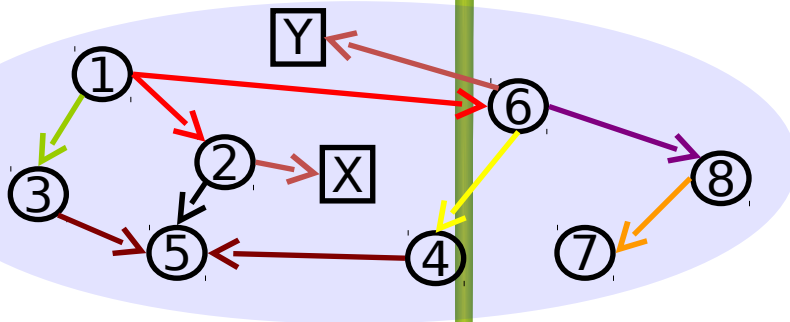
- <1> ?predicat <2>.
- <6> <violet> <8> ?predicat <6>.
- <8> <orange> <7>.
- <6> <jaune> <4>.
- <4> <marron> <5>
- <3> <marron> <5>.

Réponse

?predicat
<rouge>

Le parcours de graphe pas à pas

Rechercher un objet



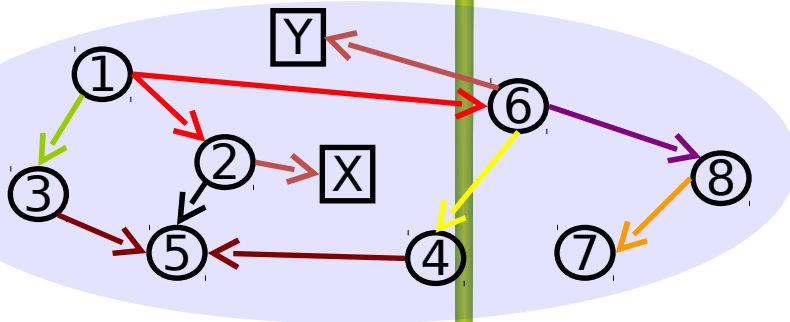
<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

Quelle est la ressources liée en tant qu'objet à la ressource <6> par le prédicat <violet> ?

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Le parcours de graphe pas à pas

Rechercher un objet



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

Quelle est la ressources liée en tant qu'objet à la ressource <6> par le prédicat <violet> ?

Reformuler la question

Sujet = <6>

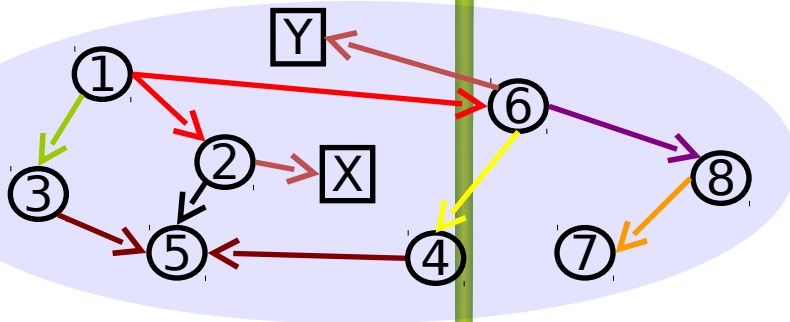
Prédicat = <violet>

Objet = ?

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>
<3> <marron> <5>.

Le parcours de graphe pas à pas

Rechercher un objet



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

Quelle est la ressource liée en tant qu'objet à la ressource <6> par le prédicat <violet> ?

Reformuler la question

Sujet = <6>

Prédicat = <violet>

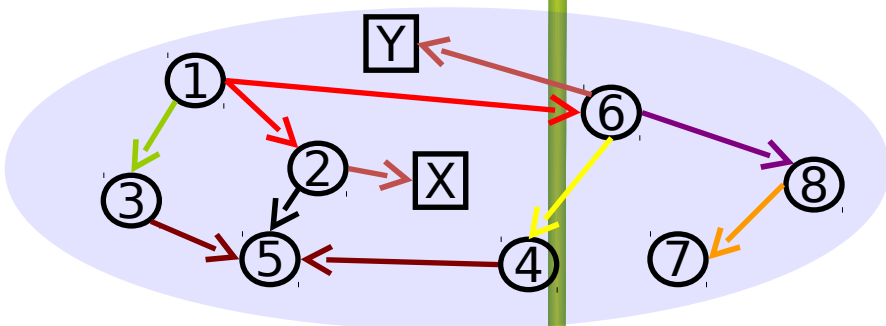
Objet = ?

Sous forme de triplets

→ <6> <violet> <6> <8> <violet> ?objet.
<8> <orange> <7>.
<6> <j jaune> <4>.
<4> <marron> <5>
<3> <marron> <5>.

Le parcours de graphe pas à pas

Rechercher un objet



- <1> <rouge> <2>.
- <1> <rouge> <6>.
- <1> <vert> <3>.
- <2> <bleu> "X".
- <2> <noir> <5>.
- <6> <bleu> "Y".

Quelle est la ressource liée en tant qu'objet à la ressource <6> par le prédicat <violet> ?

Reformuler la question

Sujet = <6>

Prédicat = <violet>

Objet = ?

Sous forme de triplets

→ <6> <violet> <6> <8> <violet> ?objet.

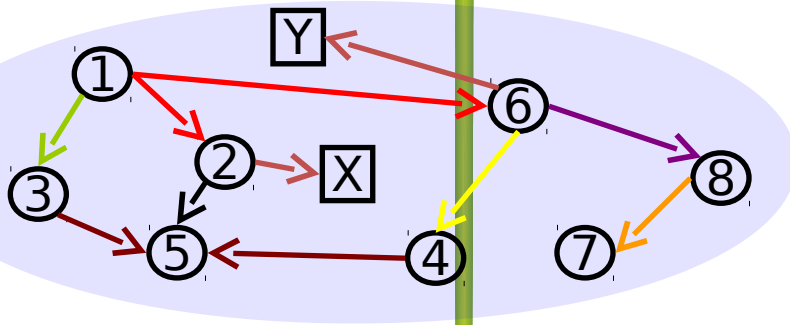
- <8> <orange> <7>.
- <6> <jaune> <4>.
- <4> <marron> <5>.
- <3> <marron> <5>.

Réponse

?objet
<8>

Le parcours de graphe pas à pas

Rechercher un sujet



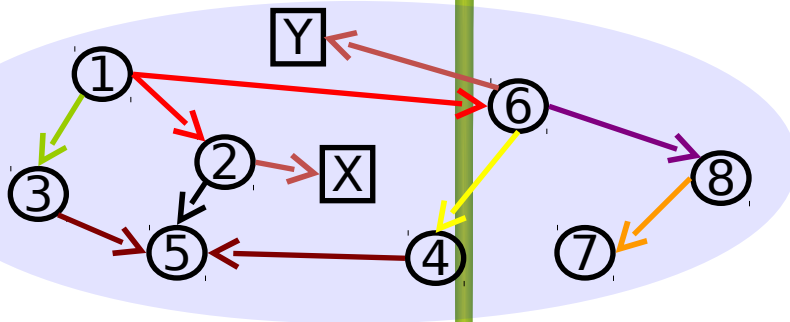
<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

Quelle est la ressource sujet dont le prédicat est <orange> et l'objet <7> ?

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Le parcours de graphe pas à pas

Rechercher un sujet



<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

Quelle est la ressource sujet dont le prédicat est <orange> et l'objet <7> ?

Reformuler la question

Sujet = ?

Prédicat = <orange>

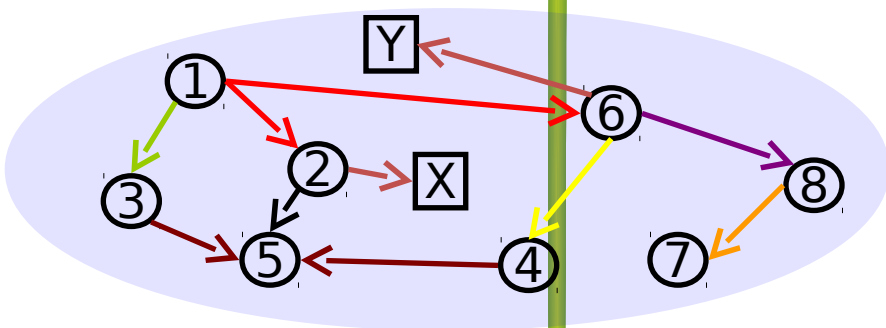
Objet = <7>

Sous forme de triplets

→ ?sujet <orange> <7>.
<6> <violet> <8>.
<8> <orange> <7>.
<6> <j jaune> <4>.
<4> <marron> <5>
<3> <marron> <5>.

Le parcours de graphe pas à pas

Rechercher un sujet



- <1> <rouge> <2>.
- <1> <rouge> <6>.
- <1> <vert> <3>.
- <2> <bleu> "X".
- <2> <noir> <5>.
- <6> <bleu> "Y".

Quelle est la ressource sujet dont le prédicat est <orange> et l'objet <7> ?

Reformuler la question

Sujet = ?

Prédicat = <orange>

Objet = <7>

Sous forme de triplets

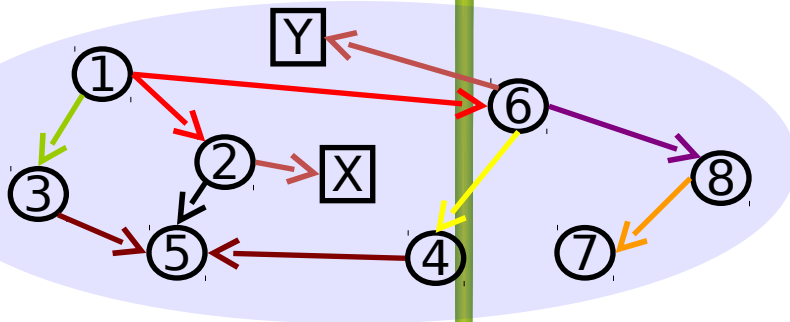
- <6> <violet> <8>.
- <8> <orange> <7>.
- <6> <jaune> <4>.
- <4> <marron> <5>.
- <3> <marron> <5>.

Réponse

?sujet
<8>

Le parcours de graphe pas à pas

Rechercher des sujets et des objets



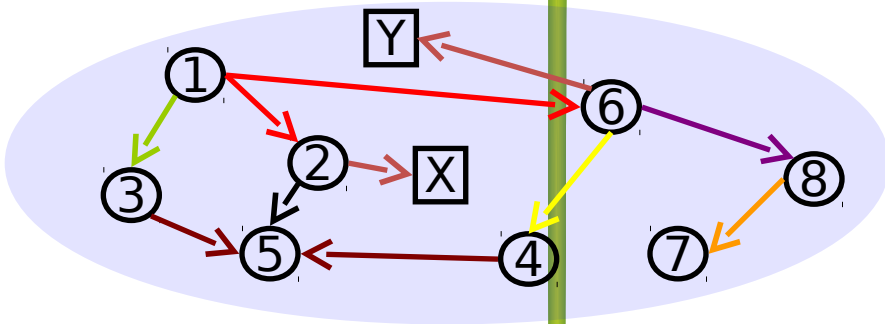
<1> <rouge> <2>.
<1> <rouge> <6>.
<1> <vert> <3>.
<2> <bleu> "X".
<2> <noir> <5>.
<6> <bleu> "Y".

Quelles sont les littéraux liés par le prédicat <bleu> aux ressources liés par le prédicat <rouge> à la ressource <1> ?

<6> <violet> <8>.
<8> <orange> <7>.
<6> <jaune> <4>.
<4> <marron> <5>.
<3> <marron> <5>.

Le parcours de graphe pas à pas

Rechercher des sujets et des objets



- <1> <rouge> <2>.
- <1> <rouge> <6>.
- <1> <vert> <3>.
- <2> <bleu> "X".
- <2> <noir> <5>.
- <6> <bleu> "Y".

Quelles sont les littéraux liés par le prédicat <bleu> aux ressources liés par le prédicat <rouge> à la ressource <1> ?

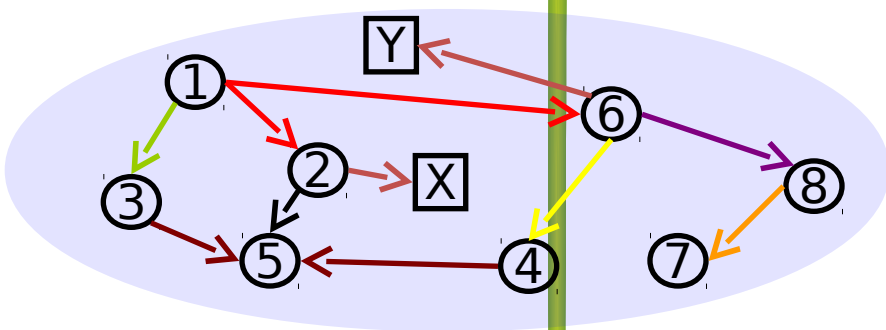
Reformuler la question

Sujet = <1> Sujet = ?ressources
 Prédicat = <rouge> Prédicat = <bleu>
 Objet = ?ressources Objet = ?littéraux

- <6> <violet> <8>.
- <8> <orange> <7>.
- <6> <jaune> <4>.
- <4> <marron> <5>
- <3> <marron> <5>.

Le parcours de graphe pas à pas

Rechercher des sujets et des objets



- <1> <rouge> <2>.
- <1> <rouge> <6>.
- <1> <vert> <3>.
- <2> <bleu> "X".
- <2> <noir> <5>.
- <6> <bleu> "Y".

Quelles sont les littéraux liés par le prédicat <bleu> aux ressources liés par le prédicat <rouge> à la ressource <1> ?

Reformuler la question

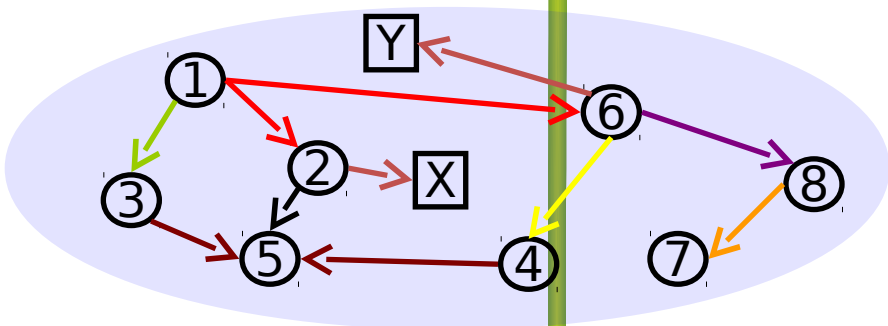
Sous forme de triplets

Sujet = <1> Sujet = ?ressources
 Prédicat = <rouge> Prédicat = <bleu>
 Objet = ?ressources Objet = ?littéraux

→ <6> <violet> <1> <rouge> ?ressources.
 <8> <orange> <7> ?ressources <bleu> ?littéraux.
 <6> <jaune> <4>.
 <4> <marron> <5>
 <3> <marron> <5>.

Le parcours de graphe pas à pas

Rechercher des sujets et des objets



- <1> <rouge> <2>.
- <1> <rouge> <6>.
- <1> <vert> <3>.
- <2> <bleu> "X".
- <2> <noir> <5>.
- <6> <bleu> "Y".

Quelles sont les littéraux liés par le prédicat <bleu> aux ressources liés par le prédicat <rouge> à la ressource <1> ?

Reformuler la question

Sous forme de triplets

Sujet = <1> Sujet = ?ressources
 Prédicat = <rouge> Prédicat = <bleu>
 Objet = ?ressources Objet = ?littéraux

→ <6> <violet> <1> <rouge> ?ressources.
 <8> <orange> <7> ?ressources <bleu> ?littéraux.
 <6> <jaune> <4>.
 <4> <marron> <5>

Réponse

?ressources	?littéraux
<2>	"X"
<6>	"Y"

Construction d'une requête SPARQL

e question et son équivalence sous forme de phrases simples avec des inconnus

Je veux connaître l'URI et les noms des personnes que connaît Tim Berners-Lee
à partir de son fichier FOAF

Tim Berners-Lee connaît des personnes.

Ces personnes s'appellent ???.

Construction d'une requête SPARQL

Les phrases simples sont transformés en triplet RDF avec la syntaxe turtle.

Je veux connaître l'URI et les noms des personnes que connaît Tim Berners-Lee
à partir de son fichier FOAF

```
<http://www.w3.org/People/Berners-Lee/card#i> foaf:knows ?person  
?personnes foaf:name ?nom.
```


Construction d'une requête SPARQL

Ajout de l'URI du graphe sur lequel porte la recherche

Je veux connaître l'URI et les noms des personnes que connaît Tim Berners-L

```
FROM <http://www.w3.org/People/Berners-  
Lee/card>
```

```
<http://www.w3.org/People/Berners-Lee/card#i> foaf:knows ?person  
?personnes foaf:name ?nom.
```

Construction d'une requête SPARQL

Ajout du mot-clé SELECT et précision des variables recherchées

```
SELECT ?personnes ?nom
```

```
FROM <http://www.w3.org/People/Berners-  
Lee/card>
```

```
<http://www.w3.org/People/Berners-Lee/card#i> foaf:knows ?personne  
?personnes foaf:name ?nom.
```

Construction d'une requête SPARQL

Ajout des éléments obligatoires de la syntaxe SPARQL

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
  SELECT ?personnes ?nom
```

```
  FROM <http://www.w3.org/People/Berners-  
Lee/card>
```

```
  WHERE {
```

```
    <http://www.w3.org/People/Berners-Lee/card#i> foaf:knows ?personne  
    ?personnes foaf:name ?nom.
```

```
  }
```

Description de la syntaxe SPARQL

```
PREFIX dbpedia: <http://dbpedia.org/resource/>  
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

Déclaration des
préfixes et des
espaces de nom

```
SELECT ?oeuvre
```

```
WHERE {
```

```
  ?oeuvre <http://dbpedia.org/ontology/author> dbpedia:Victor_Hugo.
```

```
  ?oeuvre <http://www.w3.org/2000/01/rdf-schema#label> "Les Misérables"@en.
```

```
}
```

Description de la syntaxe SPARQL

```
PREFIX dbpedia: <http://dbpedia.org/ontology/>  
PREFIX foaf: <http://xmlns.foaf.org/>
```

Déclaration du type
de requête

```
SELECT ?oeuvre
```

```
WHERE {
```

```
  ?oeuvre <http://dbpedia.org/ontology/author> dbpedia:Victor_Hugo.
```

```
  ?oeuvre <http://www.w3.org/2000/01/rdf-schema#label> "Les Misérables"@en.
```

```
}
```

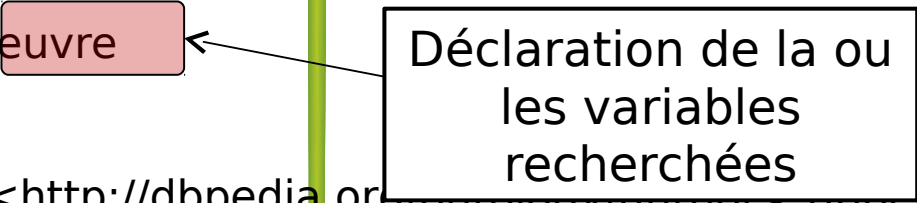
Description de la syntaxe SPARQL

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?oeuvre
```

Déclaration de la ou
les variables
recherchées



```
WHERE {
```

```
?oeuvre <http://dbpedia.org/ontology/author> dbpedia:Victor_Hugo.
```

```
?oeuvre <http://www.w3.org/2000/01/rdf-schema#label> "Les Misérables"@en.
```

```
}
```

Description de la syntaxe SPARQL

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?oeuvre
```

```
WHERE {
```

```
  ?oeuvre <http://dbpedia.org/ontology/author> dbpedia:Victor_Hugo.
```

```
  ?oeuvre <http://www.w3.org/2000/01/rdf-schema#label> "Les Misérables"@en.
```

```
}
```

Déclaration des
contraintes



Description de la syntaxe SPARQL

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT ?oeuvre
```

```
WHERE {
```

```
  ?oeuvre <http://dbpedia.org/ontology/author> dbpedia:Victor_Hugo.
```

```
  ?oeuvre <http://www.w3.org/2000/01/rdf-schema#label> "Les Misérables" @en
```

```
}
```

Déclaration d'un triplet
on note l'utilisation des
chevrons si l'URI est exprimé
complètement

Description de la syntaxe SPARQL

```
PREFIX dbpedia: <http://dbpedia.org/resource/>
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

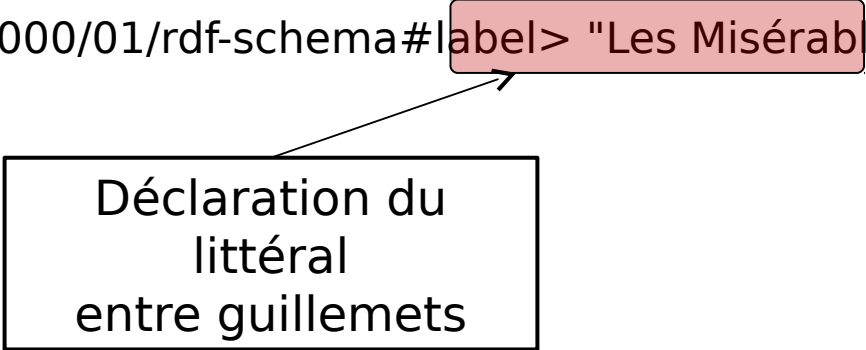
```
SELECT ?oeuvre
```

```
WHERE {
```

```
  ?oeuvre <http://dbpedia.org/ontology/author> dbpedia:Victor_Hugo.
```

```
  ?oeuvre <http://www.w3.org/2000/01/rdf-schema#label> "Les Misérables"@en.
```

```
}
```



Déclaration du
littéral
entre guillemets

Le format XML pour les résultats

```
<?xml version="1.0" ?>
```

```
<sparql
  xmlns="http://www.w3.org/2005/sparql-results#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/sw/DataAccess/rf1/result2.xsd">
```

```
<!-- En-tête -->
```

```
<head>
```

```
<variable name="personnes"/>
```

```
<variable name="nom"/>
```

```
</head>
```

Rappel des variables déclarées

```
<!-- Résultats -->
```

```
<results distinct="false" ordered="true">
```

```
<result>
```

```
<binding
```

```
name="personnes"><uri>http://dbpedia.org/resource/John_Seely_Brown</uri></binding>
```

```
<binding name="nom"><literal>John Seely Brown</literal></binding>
```

```
</result>
```

```
<result>
```

```
<binding
```

```
name="personnes"><uri>http://dbpedia.org/resource/John_Gage</uri></binding>
```

```
<binding name="nom"><literal>John Gage</literal></binding>
```

```
</result>
```

```
</results>
```

Le format XML pour les résultats

```
<?xml version="1.0" ?>
```

```
<sparql
  xmlns="http://www.w3.org/2005/sparql-results#"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/sw/DataAccess/rf1/result2.xsd">
```

```
<!-- En-tête -->
```

```
<head>
  <variable name="personnes"/>
  <variable name="nom"/>
</head>
```

```
<!-- Résultats -->
```

```
<results distinct="false" ordered="true">
```

```
<result>
  <binding
name="personnes"><uri>http://dbpedia.org/resource/John_Seely_Brown</uri> </binding
g>
  <binding name="nom"><literal>John Seely Brown</literal> </binding>
</result>
```

```
<result>
  <binding
name="personnes"><uri>http://dbpedia.org/resource/John_Gage</uri> </binding>
  <binding name="nom"><literal>John Gage</literal> </binding>
</result>
```

Un élément « result » par combinaisons possibles variables

Le format JSON pour les résultats

Le format JSON en SPARQL 1.0 n'est pas une recommandation mais une note de travail du W3C.

```
{ "head": { "link": [], "vars": ["oeuvre"] },  
  "results": { "distinct": false, "ordered": true, "bindings": [  
    { "oeuvre": { "type": "uri", "value":  
      "http://dbpedia.org/resource/Les_Mis%C3%A9rables" }} ] } }
```

SPARQL Pratique

Exercice : effectuer des requêtes SPARQL sur dbpedia

Deux interfaces Web sont disponibles pour effectuer des requêtes

- Interface classique : <http://dbpedia.org/sparql>
- Interface améliorée : <http://dbpedia.org/snorql>

N'oubliez pas d'indiquer la syntaxe de sortie HTML pour visualiser directement les résultats.

Découvrir les données d'un sparql endpoint pas à pas

Découvrir les types de ressources présentes dans l'entrepôt :

```
select distinct ?Concept where
{
  ?s a ?Concept
}
```

Le mot « a » en position de prédicat correspond à rdf:type

Explorer les prédicats distincts liés à une ressource d'un type

```
select distinct ?predicat where
{
  ?sujet a <http://xmlns.com/foaf/0.1/Person>.
  ?sujet ?predicat ?objet.
}
```

Rechercher les ressources de ce type en associant un des prédicats trouvés

```
select distinct ?sujet ?objet
where
{
  ?sujet a <http://xmlns.com/foaf/0.1/Person>.
  ?sujet <http://www.w3.org/2000/01/rdf-schema#label> ?objet.
}
```

Découvrir les données d'un sparql endpoint pas à pas

chercher tous les triplets dont une ressource particulière est sujet

```
select distinct ?predicat ?objet
where
{
  <http://dbpedia.org/resource/Bill_Clinton> ?predicat ?objet.
}
```

Complicquer la requête pour commencer à parcourir le graphe

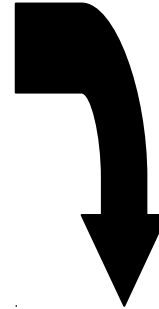
```
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix dbpr: < http://dbpedia.org/ontology/>
select distinct ?label
where
-
<http://dbpedia.org/resource/Bill_Clinton> dbpr:birthPlace ?objet.
?objet rdfs:label ?label
-
```


Trouver une URI dans Dbpedia

http://en.wikipedia.org/wiki/Victor_Hugo



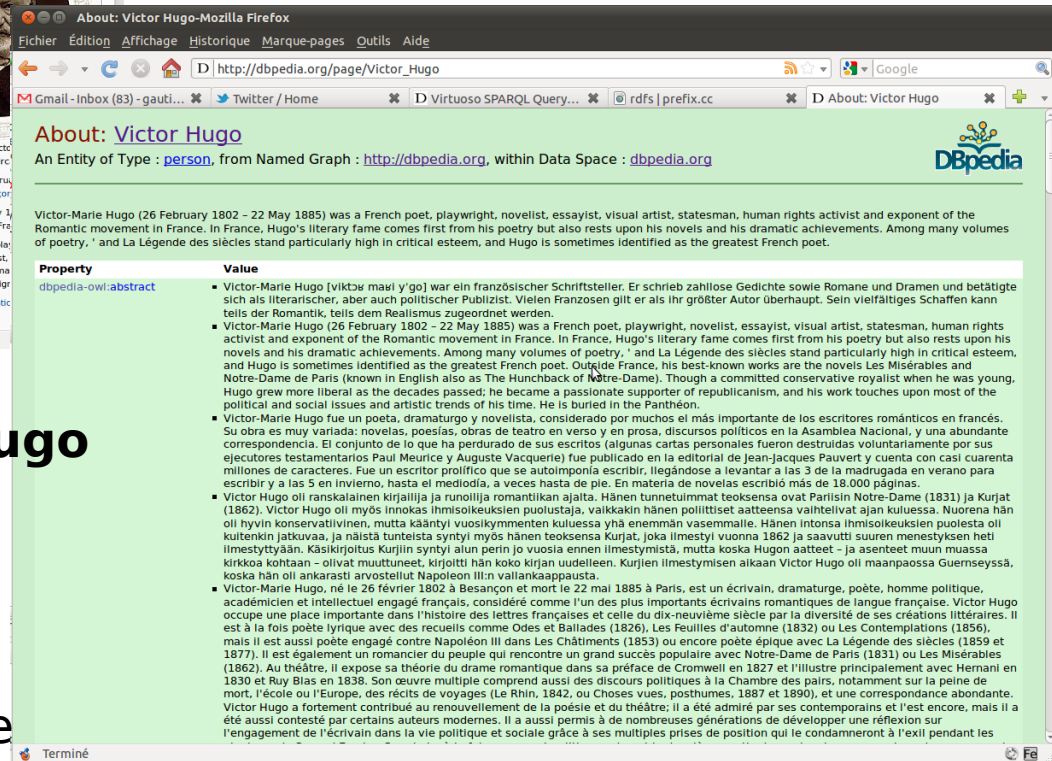
The screenshot shows the Wikipedia page for Victor Hugo. The browser address bar displays the URL http://en.wikipedia.org/wiki/Victor_Hugo. The page content includes a portrait of Victor Hugo, a biographical summary, and a table of contents. The table of contents lists sections such as Life, Writings, Political life and exile, Religious views, Victor Hugo and music, Declining years and death, Drawings, Memorials, Works, References, and External links.



http://dbpedia.org/resource/Victor_Hugo



La redirection du Linked Data page



The screenshot shows the Dbpedia page for Victor Hugo. The browser address bar displays the URL http://dbpedia.org/page/Victor_Hugo. The page content includes a portrait of Victor Hugo, a biographical summary, and a table of contents. The table of contents lists sections such as Life, Writings, Political life and exile, Religious views, Victor Hugo and music, Declining years and death, Drawings, Memorials, Works, References, and External links.

Exercice 1

Formuler une requête qui affiche toutes les informations associées à l'URI de Victor Hugo.

URI utile

Victor Hugo : http://dbpedia.org/resource/Victor_Hugo

Exercice 1

Formuler une requête qui affiche toutes les informations associées à l'URI de Victor Hugo.

URI utile

Victor Hugo : http://dbpedia.org/resource/Victor_Hugo

Requête

```
select ?predicat ?objet
where
{
  <http://dbpedia.org/resource/Victor\_Hugo> ?predicat ?
  objet.
}
```

Exercice 2

Rechercher une ressource ayant pour auteur Victor Hugo et pour type « Livre ».

URI utile

Victor Hugo : http://dbpedia.org/resource/Victor_Hugo

Propriété auteur : <http://dbpedia.org/ontology/author>

Type Livre : <http://dbpedia.org/ontology/Book>

Exercice 2

Rechercher une ressource ayant pour auteur Victor Hugo et pour type « Livre ».

URI utile

Victor Hugo : http://dbpedia.org/resource/Victor_Hugo

Propriété auteur : <http://dbpedia.org/ontology/author>

Type Livre : <http://dbpedia.org/ontology/Book>

Requête

```
select ?livres
where
{
  ?livres <http://dbpedia.org/ontology/author>
  <http://dbpedia.org/resource/Victor\_Hugo>.
  ?livres a <http://dbpedia.org/ontology/Book>
}
```

Exercice 3

Rechercher des romanciers français nés à Paris.

URI utile

Prédicat « Sujet » : <http://purl.org/dc/terms/subject>

Romanciers français : http://dbpedia.org/resource/Category:French_novelists

Prédicat « Lieu de naissance » : <http://dbpedia.org/ontology/birthPlace>

Paris : <http://dbpedia.org/resource/Paris>

Exercice 3

Rechercher des romanciers français nés à Paris.

URI utile

Prédicat « Sujet » : <http://purl.org/dc/terms/subject>

Romanciers français : http://dbpedia.org/resource/Category:French_novelists

Prédicat « Lieu de naissance » : <http://dbpedia.org/ontology/birthPlace>

Paris : <http://dbpedia.org/resource/Paris>

Requête

```
select ?personnes
where
{
  ?personnes <http://purl.org/dc/terms/subject>
  <http://dbpedia.org/resource/Category:French\_novelists>.
  ?personnes <http://dbpedia.org/ontology/birthPlace>
  <http://dbpedia.org/resource/Paris>
}
```

SPARQL
Aller plus loin

CONSTRUCT : récupérer ou créer des triplets

Ce type de requêtes permet de récupérer un ensemble de triplets stockés dans le triple store pour les manipuler localement, de « re-modéliser » des nouveaux triplets à partir des triplets existants voire d'appliquer des règles d'inférence.
Deux parties

Construction des triplets cible

Une ressource a pour titre une étiquette

Recherche dans les triplets existants

Une ressource a pour étiquette une étiquette.

CONSTRUCT : récupérer ou créer des triplets

Ce type de requêtes permet de récupérer un ensemble de triplets stockés dans le triple store pour les manipuler localement, de « re-modéliser » des nouveaux triplets à partir des triplets existants voire d'appliquer des règles d'inférence.
Deux parties

Construction des triplets cible

?resource dc:title ?label

Recherche dans les triplets existants

?resource rdfs:label ?label

CONSTRUCT : récupérer ou créer des triplets

Ce type de requêtes permet de récupérer un ensemble de triplets stockés dans le triple store pour les manipuler localement, de « re-modéliser » des nouveaux triplets à partir des triplets existants voire d'appliquer des règles d'inférence.
Deux parties

```
CONSTRUCT {
```

```
    ?resource dc:title ?label
```

```
}
```

```
WHERE {
```

```
    ?resource rdfs:label ?label
```

```
}
```

ASK : vérifier la présence d'un graphe

Ce type de requêtes permet de vérifier la présence dans l'ensemble de données RDF interrogé des contraintes exprimées.

Y-a-t'il un triplet dont le prédicat est dc:title ?

Une ressource a pour titre une étiquette

ASK : vérifier la présence d'un graphe

Ce type de requêtes permet de vérifier la présence dans l'ensemble de données RDF interrogé des contraintes exprimées.

Y-a-t'il un triplet dont le prédicat est dc:title ?

?resource dc:title ?label

ASK : vérifier la présence d'un graphe

Ce type de requêtes permet de vérifier la présence dans l'ensemble de données RDF interrogé des contraintes exprimées.

```
ASK {
```

```
  ?resource dc:title ?label
```

```
}
```

SPARQL : DISTINCT/OPTIONAL

Le mot-clé DISTINCT

```
SELECT ?predicat
WHERE {
  <http://dbpedia.org/resource/Victor_Hugo> ?predicat ?objet.
}
```

Autant de résultats que de triplets dont Victor Hugo est sujet, plusieurs résultats identiques

```
SELECT DISTINCT ?predicat
WHERE {
  <http://dbpedia.org/resource/Victor_Hugo> ?predicat ?objet.
}
```

Dédoublonnage automatique des résultats

Le mot-clé OPTIONAL

```
PREFIX dbpedia: <http://dbpedia.org/ontology/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/ >
SELECT ?auteur ?depiction WHERE {
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>
  ?auteur foaf:depiction ?depiction.
  ?auteur dbpedia:birthPlace <http://dbpedia.org/resource/Paris>.
}
```

Résultats que si toutes les contraintes sont vérifiées

```
SELECT ?auteur ?depiction WHERE {
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>
  OPTIONAL {?auteur foaf:depiction ?depiction.}
  ?auteur dbpedia:birthPlace <http://dbpedia.org/resource/Paris>.
}
```

Résultats même si la contrainte optionnelle n'est pas vérifiée (mais affichage du résultat si elle existe)

SPARQL : UNION/ORDER/LIMIT

Le mot-clé UNION

```
SELECT ?auteur WHERE {  
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>  
  {  
    {?auteur <http://dbpedia.org/ontology/birthPlace>  
    <http://dbpedia.org/resource/Paris> .}  
    UNION  
    {?auteur <http://dbpedia.org/ontology/birthPlace> <http://dbpedia.org/resource/Besan  
    %C3%A7on> .}  
  }  
}
```

UNION permet d'exprimer
l'opérateur booléen « OU »
(inclusif)

Le mot-clé ORDER

```
SELECT ?auteur ?name WHERE {  
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>  
    foaf:name ?name  
}  
ORDER BY ?name
```

ORDER BY permet de classer
les résultats suivants une
variable déclarée

Le mot-clé LIMIT

```
SELECT ?auteur ?name WHERE {  
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>  
    foaf:name ?name  
}  
ORDER BY ?name  
LIMIT 20
```

LIMIT permet de limiter le
nombre de résultats

SPARQL : les filtres

Il est possible d'appliquer des filtres à une requête SPARQL pour limiter les résultats.

Filtre lang()

```
SELECT ?abstract
WHERE {
  <http://dbpedia.org/resource/Victor_Hugo>
  <http://dbpedia.org/ontology/abstract> ?abstract.
  FILTER (lang(?abstract)="fr")
}
```

Filtrer selon la langue du littéral

Filtre de comparaison

```
SELECT distinct ?auteur ?date WHERE {
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>.
  ?auteur <http://dbpedia.org/ontology/birthDate> ?date.
  FILTER ( xsd:dateTime(?date) > xsd:dateTime("1950-01-01")).
}
```

Filtrer les données avec les signes de comparaison

Filtre de regexp

```
SELECT ?auteur ?name WHERE {
  ?auteur a <http://dbpedia.org/class/yago/FrenchNovelists>.
  ?auteur foaf:name ?name
  FILTER regex(?name, '^Jean').
}
```

Filtrer un littéral en y appliquant une regexp

Consulter la recommandation SPARQL pour découvrir les autres filtres

SPARQL 1.1 : quelques nouveautés

1.1 en cours de standardisation mais déjà implémenté ajoute des fonctions d'aggrégation
Compter un nombre d'éléments

```
SELECT (COUNT(?person) as ?nbperson)
WHERE {
  ?person a foaf:Person.
}
```

Somme des valeurs associées à une expression

```
SELECT (SUM(?population) as ?nbpeople)
WHERE {
  ?ville <http://dbpedia.org/ontology/populationTotal> ?population.
  ?ville <http://dbpedia.org/ontology/region> <http://dbpedia.org/resource/%C3%8Ele-
de-France_%28region%29>
}
```

Moyenne des valeurs associées à une expression

```
SELECT (AVG(?population) as ?avgpeople)
WHERE {
  ?ville <http://dbpedia.org/ontology/populationTotal> ?population.
  ?ville <http://dbpedia.org/ontology/region> <http://dbpedia.org/resource/%C3%8Ele-
de-France_%28region%29>
}
```

SPARQL 1.1 : quelques nouveautés

Identifier une expression qui n'existe pas

```
SELECT ?person
WHERE {
  ?person a foaf:Person.
  FILTER NOT EXISTS {?person rdfs:label "Victor Hugo"@fr }
}
```

Simplifier l'union

```
SELECT ?label
WHERE {
  ?person (foaf:name|rdfs:label) ?label.
}
```

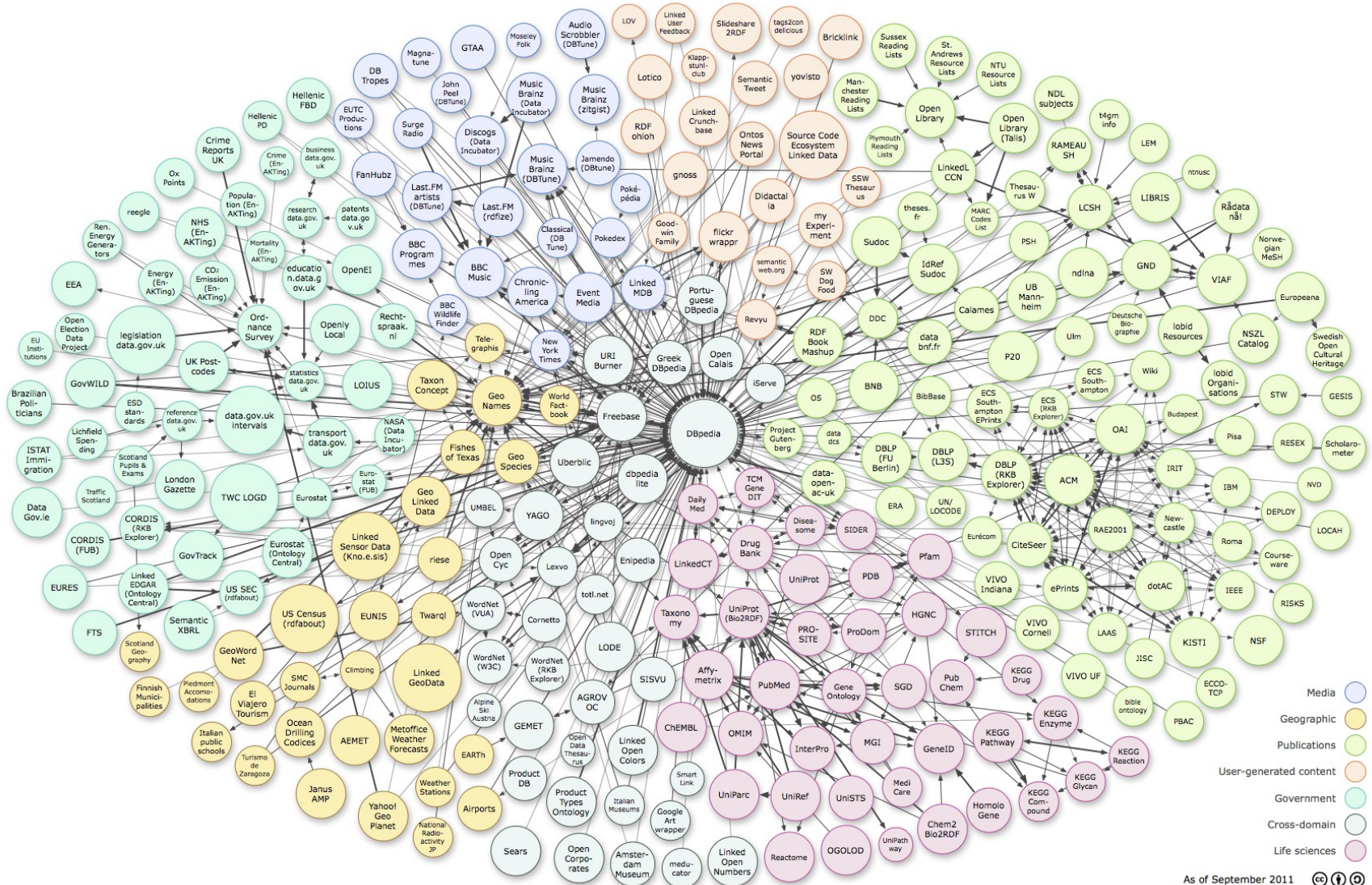
Simplifier les imbrications

```
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix dbpr: <http://dbpedia.org/ontology/>
prefix db: <http://dbpedia.org/resource/>
select distinct ?label
where
{
  db:Bill_Clinton dbpr:birthPlace ?objet.
  ?objet rdfs:label ?label
}
```

```
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix dbpr: <http://dbpedia.org/ontology/>
prefix db: <http://dbpedia.org/resource/>
select distinct ?label
where
{
  db:Bill_Clinton dbpr:birthPlace/rdfs:label ?label
}
```

Consommer les données du linked Data avec SPARQL

Linked Open Data cloud diagram



As of September 2011

Exemples de sparql endpoint disponibles

sparql endpoint lié à un ensemble de données

- Dbpedia : <http://dbpedia.org/sparql>
- Data.gov : <http://services.data.gov/sparql>
- Data.gov.uk : <http://services.data.gov.uk/sparql>
- Music Brainz : <http://dbtune.org/musicbrainz/>
- Linked Geo Data : <http://linkedgeo.org/sparql/>
- GeoLinkedData : <http://geo.linkeddata.es/web/guest/endpoints>
- Données bibliographiques de la British Library : <http://bnb.bibliographica.org/sparql>
- Open University : <http://data.open.ac.uk/query>

sparql endpoint agrégeant plusieurs ensemble de données

- Uberblic : <http://api.talis.com/stores/uberblic/services/sparql>
- Linked Data semantic repository : <http://ldsr.ontotext.com/sparql>
- LOD cache : <http://lod.openlinksw.com/sparql>

Etat et disponibilité des sparql endpoint :

<http://labs.mondeca.com/sparqlEndpointsStatus/index.html>

Librairies pour manipuler du RDF

Il existe de très nombreuses librairies pour manipuler du RDF. Quelques exemples :

- Java (triples) :
 - Jena
 - Sesame
 - Trialox SCB
 - RDF2Go
 - Topaz
 - RDFReactor
 - So(m)mer
 - Elmo
 - jenabean
 - Linked Data API

- PHP :
 - RAP
 - ARC

Interroger directement un sparql endpoint : Jquery SPARQL

Jquery-SPARQL est un module JQuery qui permet d'interroger un sparql endpoint et de récupérer les données en JSON.

<https://github.com/jgeldart/jquery-sparql>

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.4.4/jquery.min.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<script src="urlencode.js"></script>
<script src="jquery.sparql.js"></script>
<script>
  var cbfunc = function(results) {
    $("#results").val(JSON.stringify(results));
  };

  $(document).ready(function() {
    $("#run_sparql").click(function(e) {
      $.sparql("http://dbpedia.org/sparql")
        .prefix("rdfs","http://www.w3.org/2000/01/rdf-schema#")
        .select(["?label"])
        .where("<http://dbpedia.org/resource/Tim_Berners-Lee>", "rdfs:label", "?label")
        .execute(cbfunc);
      return false;
    });
  });
</script>
```

<http://www.lespetitescases.net/semweblabs/semwebpro/sparql/jquery-sparql/ex>

Interroger directement un sparql endpoint : ARC2 et PHP

ARC2 est un framework PHP mis au point par Benjamin Nowack qui permet de créer, gérer et interroger un triple store local ou interroger un sparql endpoint distant.

<https://github.com/semsol/arc2>

```
<?php
/* configuration */
$remoteconfig = array(

    /* remote endpoint */
    'remote_store_endpoint' => 'http://dbpedia.org/sparql',
);

/* instantiation */
$remotestore = ARC2::getRemoteStore($remoteconfig);

$query="SELECT ?label WHERE {<http://dbpedia.org/resource/Tim_Berners-Lee> rdfs:label ?label
FILTER (lang(?label)='fr')}";

$result = $remotestore->query($query, 'row');
print $result['label'];
?>
```

Exemple :

<http://www.lespetitescases.net/semweblabs/semwebpro/sparql/arc2/remote.php>

et source :

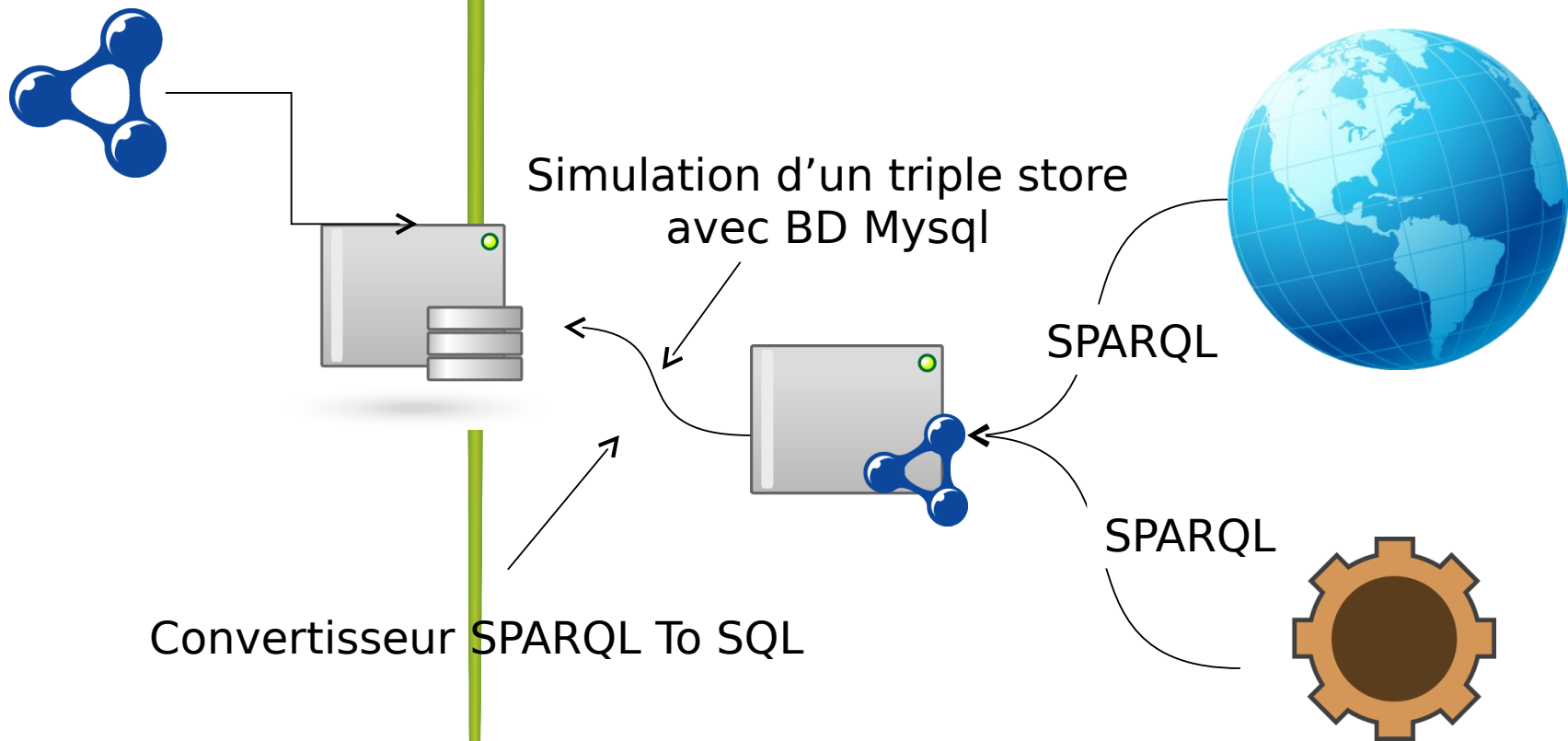
<http://www.lespetitescases.net/semweblabs/semwebpro/sparql/arc2/remote.php>

- Stocker les données dans un triple store RDF
 - Mémoire : Corese, Redstore
 - Triple store natif : Mulgara, AllegroGraph, BigOWLIM, 4store, Neo4j
 - BDR paramétrée : CubicWeb, Open Anzo, Virtuoso, ARC, Oracle 11g, Sesame, [3store](#)
- Column store : [Cstore](#), [Heart](#), [BigData](#), [Cloudera](#)

Principes de fonctionnement de ARC

es les interactions avec la base de données MySQL se font en SPARQL

Depuis l'extérieur via un sparql endpoint



Ou directement via des méthodes PHP

Configurer ARC

```
<?php
/* Fichier de configuration pour ARC 2 */
$arc_config = array(

/* Configuration de l'accès à la base de données MySQL */
'db_host' => "", /* Domaine de la base de données ; par défaut : localhost */
'db_name' => "", /* Nom de la base de données */
'db_user' => "", /* Nom de l'utilisateur de la base de données */
'db_pwd' => "", /* Mot de passe de l'utilisateur de la base de données */
/* Nom de l'entrepôt RDF
Cela permet de créer plusieurs entrepôts distincts avec la même base de données
*/
'store_name' => "",
/* Nombre d'erreurs maximums avant l'arrêt d'un script
'max_errors' => 100,
*/
/* Configuration complémentaire en cas de présence d'un proxy
'proxy_host' => '192.168.1.1',
'proxy_port' => 8080,
*/
/* Préfixe pour les nœuds blancs
'bnode_prefix' => 'bn',
*/
/* Formats supportés par l'extracteur de données structurées dans les pages Web */
'sem_html_formats' => 'rdfa microformats',
);
?>
```

LOAD : Charger des triplets

```
<?php
$url=$_GET['url'];
include_once('arc/ARC2.php');
include_once('config.php');

/*Instantiation*/
$store = ARC2::getStore($arc_config);

/*Si les tables ne sont pas créées, création des tables */
if (!$store->isSetUp()) {
    $store->setUp();
}

/* Chargement du flux distant */
if (isset($url)) {
    $load="LOAD <".$url.">";
    if ($rows = $store->query($load)) {
        $added_triples = $rows['result']['t_count'];
        echo $added_triples." triplets ont été ajoutés<br/>";
    }
}
else {
    print "Vous n'avez pas saisi d'URL valide; charger";
}
?>
```

Renvoie un tableau

Exploitation du tableau

SELECT : effectuer une requête

```
<?php
include_once('arc/ARC2.php');
include_once('config.php');

/* Instantiation de l'entrepôt à partir de la configuration */
$store=ARC2::getStore($arc_config);

/*Si les tables ne sont pas créées, création des tables */
if (!$store->isSetUp()) {
    $store->setUp();
}

$query="PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
        PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?s ?nom WHERE {
    ?s skos:subject <http://dbpedia.org/resource/Category:Star_Wars_characters>;
    rdfs:label ?nom.
    FILTER (lang(?nom)='fr')
}";

$res=$store->query($query);

if($store->getErrors()) {
    print_r($store->errors);
    echo "***\nQuery = $query";
}
else {
    print_r($res);
}
}
```

Le tableau de réponse

```
Array
(
  [query_type] => select
  [result] => Array
    (
      [variables] => Array
        (
          [0] => s
          [1] => nom
        )
      [rows] => Array
        (
          [0] => Array
            (
              [s] => http://dbpedia.org/resource/Anakin_Skywalker
              [s type] => uri
              [nom] => Anakin Skywalker
              [nom type] => literal
              [nom lang] => fr
            )
          [1] => Array
            (
              [s] => http://dbpedia.org/resource/Luke_Skywalker
              [s type] => uri
              [nom] => Luke Skywalker
              [nom type] => literal
              [nom lang] => fr
            )
        )
    )
  [query_time] => 0.17298412322998
)
```

Exploiter le tableau de réponses

```
<?php
/* [SNIP] */

$query="PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
SELECT DISTINCT ?s ?nom WHERE {
  ?s skos:subject <http://dbpedia.org/resource/Category:Star_Wars_characters>;
  rdfs:label ?nom.
  FILTER (lang(?nom)='fr')
}";

$res=$store->query($query);

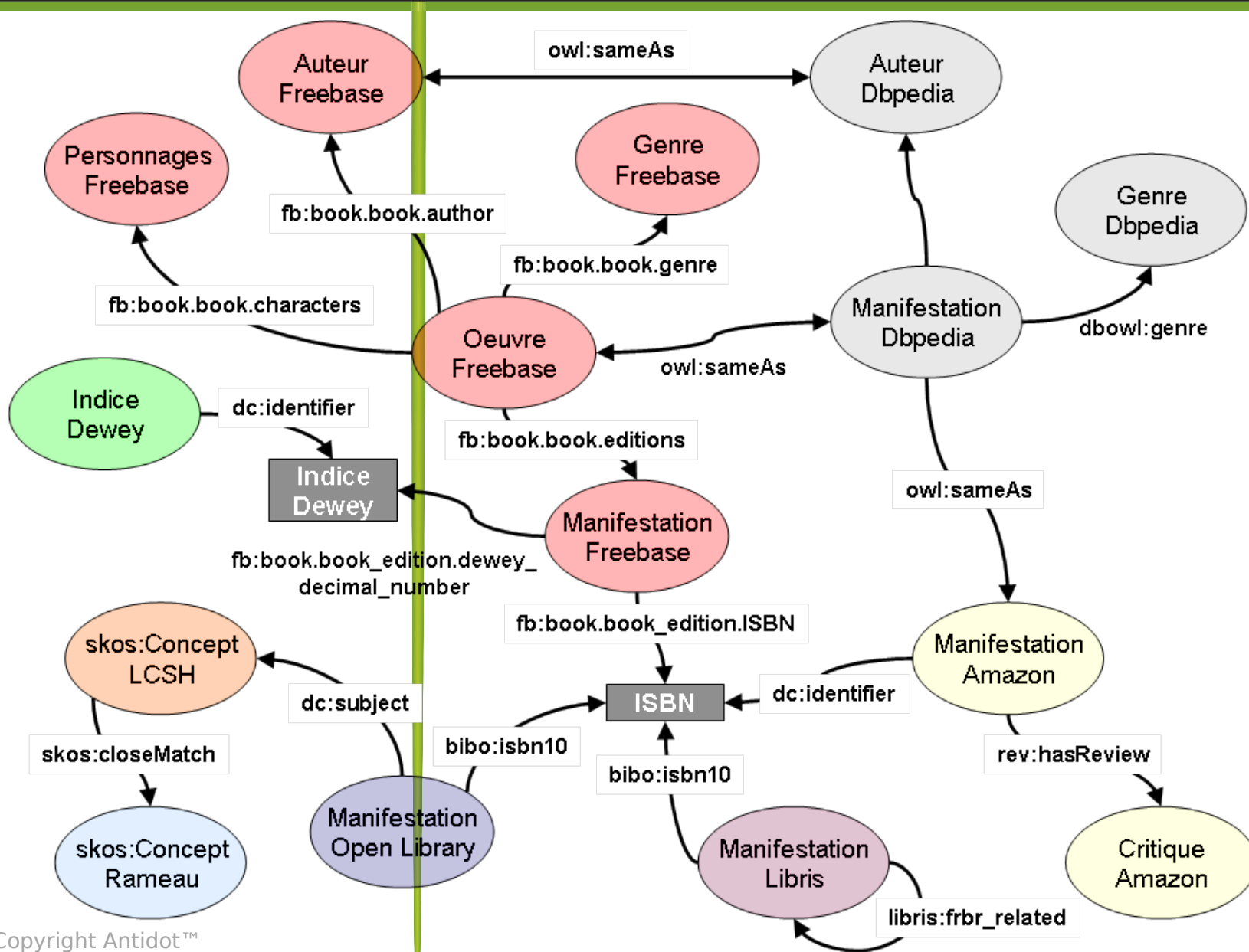
if($store->getErrors() {
  print_r($store->errors);
  echo "***\nQuery = $query";
}
else {
  print "<ul>";
  foreach ($res['result']['rows'] as $result) {
    print "<li><a href=\"". $result['s']. "\">". $result['nom']. "</a></li>";
  }
  print "</ul>";
}
```


Exemples de mashup

Exemple d'utilisation 1 : *Linked book Mashup*

Mashup

<http://www.lespoitescases.net/semweblabs/linkedbookmashup/>



Exemple d'utilisation 1 : *Linked book Mashup*



Linked book mashup



Le Seigneur des Anneaux

Par J. R. R. Tolkien

Infos bibliographiques

Éditions

Personnages

Infos bibliographiques

Le Seigneur des anneaux (The Lord of the Rings) est un roman en trois volumes de J. R. R. Tolkien paru entre 1954 et 1955. Quoique les deux histoires puissent être lues de façon indépendante, il est la suite de *Bilbo le Hobbit*, suite demandée par son éditeur à Tolkien. Durant les douze années de sa rédaction, il s'attache à faire vivre le monde dont il est le créateur, la Terre du Milieu, en truffant sa nouvelle œuvre de références et d'allusions qui la relient au monde du *Silmarillion*, sur lequel il travaille depuis 1917 et dans lequel Bilbo le Hobbit a été attiré « contre l'intention première » de son auteur. C'est une des œuvres fondamentales de la littérature dite de fantasy, terme que Tolkien explicite dans son essai de 1939 « Du conte de fées », publié dans divers recueils. Tolkien lui-même considérait son livre comme « un conte de fées [...] pour des adultes », écrit « pour amuser ; pour être agréable à lire ». Cette œuvre est composée de six livres, qui ne portent pas de titres. À l'origine, Tolkien souhaite publier *Le Seigneur des anneaux* en un seul volume, mais le prix du papier étant trop prohibitif en cette période d'après-guerre, l'œuvre est divisée en trois volumes : *La Communauté de l'anneau*, *Les Deux Tours*, et *Le Retour du roi* (The Return of the King). On fait souvent référence à cette œuvre comme à « la trilogie du Seigneur des anneaux », terme techniquement incorrect car l'œuvre fut écrite et conçue d'un seul tenant. Néanmoins, Tolkien lui-même reprend dans ses lettres, de temps à autres, le terme de « trilogie » lorsqu'il est employé par ses correspondants.

(Source : [Wikipedia](#))

Langue originale de la publication : Anglais

Indices Dewey : 823 : English fiction ;

Genres : Adventure novel ; Speculative fiction ; Fiction ; High fantasy ; Chivalric romance ; Fantasy ;

Indexation matières avec LCSH : [Fiction](#) ; [Baggins, Frodo \(Fictitious character\)](#) ; [Middle Earth \(Imaginary place\)--Fiction](#) ; [Fantasy fiction, English](#) ;

Correspondances du LCSH avec Rameau : [Nouvelles fantastiques anglaises](#) ;

Biographie de l'auteur



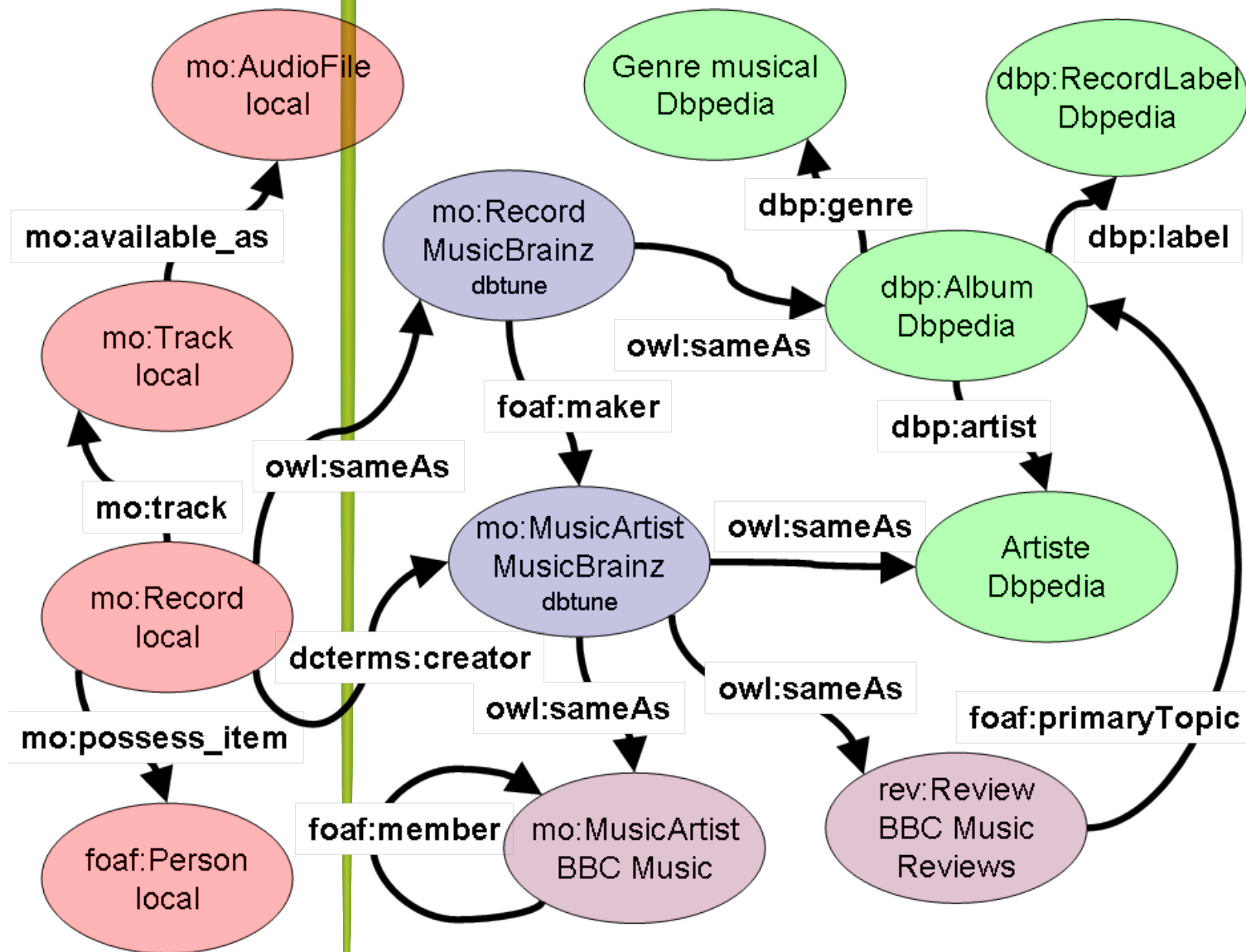
John Ronald Reuel Tolkien, plus connu sous sa signature : J. R. R. Tolkien, est un écrivain, poète, philologue et professeur d'université anglais, né le 3 janvier 1892 à Bloemfontein et mort le 2 septembre 1973 à Bournemouth. Il est principalement connu en tant qu'auteur des romans de high fantasy *Bilbo le Hobbit* et *Le Seigneur des anneaux*. Tolkien est professeur d'anglo-saxon à l'université d'Oxford de 1925 à 1949, et professeur de langue et de littérature anglaise à Merton de 1945 à 1959. Ami proche de C. S. Lewis, il est, comme lui, membre du groupe littéraire connu sous le nom d'*Inklings*. Tolkien est nommé commandeur de l'Ordre de l'Empire britannique par la reine Élisabeth II le 28 mars 1972. Après sa mort, son troisième fils Christopher publie plusieurs ouvrages basés sur les nombreuses notes et manuscrits inédits de son père, dont *Le Silmarillion*. Avec *Bilbo le Hobbit* et *Le Seigneur des anneaux*, ces livres forment un ensemble uni de récits, poèmes, essais et langues construites concernant le monde imaginaire d'Ardà, dont la Terre du Milieu est le continent principal. Dans les années 1950, Tolkien donne le nom de *legendarium* à ces écrits. De nombreux auteurs ont publié des ouvrages de fantasy avant Tolkien, mais le succès majeur remporté par *Bilbo le Hobbit* et *Le Seigneur des anneaux* au moment de leur publication en poche aux États-Unis a eu pour conséquence directe une renaissance populaire du genre, qui a conduit à considérer Tolkien comme le « père » de la fantasy moderne. Ses ouvrages ont eu une influence majeure sur les écrivains de fantasy ultérieurs. En 2008, le *Times* l'a classé sixième d'une liste des « 50 plus grands écrivains britanniques depuis 1945 ».

(Source : [Wikipedia](#))

Exemple d'utilisation 2 : *Linked My Music*

Music

www.lespetitescases.net/semweblabs/linkedmymusic/



Linked My Music

A mashup with linked musical data



Kid A

[Radiohead](#)

Date de parution : 1999
Nombre de pistes : 10
Genre que vous avez renseigné : Alt. Rock
Label : Parlophone
Genre dans wikipedia : Electronic music

Description

Kid A est le quatrième album du groupe de rock britannique Radiohead, il est sorti en 2000. Alors que les albums précédents restent dans un style rock alternatif, les albums suivants sont beaucoup plus psychédélics : Kid A marque l'apogée de ce style expérimental de Radiohead. Pour cette raison, il est considéré par beaucoup comme un chef-d'œuvre. Dans cet album, les guitares ont quasiment disparu au profit de synthétiseurs et de sampleurs. Le nom donné à l'album, Kid A (littéralement « Enfant A »), évoque pour certains un premier enfant cloné. Pour d'autres, il laisse penser que le groupe le considère comme son premier enfant. Avec Kid A, l'album suivant de Radiohead, Amnesiac, forme un diptyque de musique expérimentale, un prolongement : Kid A et Amnesiac forment en réalité le diptyque Kid Amnesiac. Ce disque comporte une majorité de chansons composée principalement de synthétiseurs et de boîtes à rythmes (Kid A, Idioteque, Everything in Its Right Place...), tout en gardant des sonorités pop/rock (In Limbo) et en explorant d'autres univers comme le free-jazz (The National Anthem). Lorsqu'on lance deux albums Kid A sur une chaîne hifi et sur un PC par exemple, à 17 secondes d'intervalle (16.8 s pour être précis) l'un de l'autre, on obtient quelque chose d'étonnant : les sons se marient à la perfection,

Biographie du créateur



Radiohead est un groupe de rock alternatif

anglais originaire d'Abingdon dans l'Oxfordshire ayant vendu près de 60 millions d'albums depuis leur début. Le groupe est composé de Thom Yorke au chant, de Jonny Greenwood à la guitare, de Colin Greenwood à la basse et au synthétiseur, de Ed O'Brien à la guitare et de Phil Selway à la batterie. Le premier single du groupe "Creep" paraît en 1992 et leur premier album Pablo Honey suivra en 1993. La popularité de Radiohead au Royaume-Uni augmente avec la sortie de leur second album intitulé The Bends en 1995 cependant c'est en 1997 avec la sortie de OK Computer qu'ils acquièrent une réputation mondiale. Les sorties consécutives de Kid A en 2000 et de Amnesiac en 2001 marquent l'apogée du groupe bien que la critique soit divisée. Cette période marque un changement dans la ligne de conduite du groupe, en effet, l'incorporation dans leurs morceaux de musique électronique, de post-punk et de jazz donne une nouvelle dimension à leur musicalité. La sortie de Hail to the Thief en 2003 marque la séparation entre Radiohead et leur principal label EMI.

[En savoir plus](#)

La boîte à souvenirs

<http://www.lespetitescases.net/souvenirs/>

Boîte à souvenirs



Memories
box

Ce site a deux vocations :

- rassembler différents souvenirs personnels de visites et/ou d'événements sous la forme d'albums photos améliorés ;
- démontrer les avantages des technologies du Web sémantique et les possibilités du Web de données.

[En savoir plus](#)



Visite de La Garde-Adhémar

13 August 2007



Visite du Musée Rolin

04 August 2007

Visiter par carte



Visiter par nuage

[architecturereligieuse](#) ;
[herault](#) ; [nature](#) ; [paysage](#) ;
[sculpture](#) ;
[moyenage](#) ;
[saintbertranddecomminges](#) ; [hauteqaronne](#) ;
[montagne](#) ; [conques](#) ; [antiquite](#) ; [grece](#) ;
[mosaique](#) ; [morvan](#) ; [bourgogne](#)
; [chateau](#) ; [peinture](#) ; [epoquemoderne](#) ;
[renaissance](#) ; [gardeadhemar](#) ; [drome](#) ; [leur](#) ;
[animal](#) ; [rue](#) ; [aude](#) ; [nuage](#) ; [architecture](#) ;
[mycenes](#) ; [architecturefuneraire](#) ; [louvre](#) ;
[pierrefonds](#) ; [oise](#) ; [d50](#) ;
[epoquecontemporaine](#) ; [rodin](#) ; [gers](#)
; [larressingle](#) ; [meteores](#) ; [meteora](#) ; [paysae](#) ;
[oxford](#) ; [vitrail](#) ; [maison](#) ; [seine](#) ; [75006](#) ;
[75018](#) ;
<http://lespetitescases.net/gautierpoupeaurd75019> ;

Quelques jours dans le

Merci pour votre attention