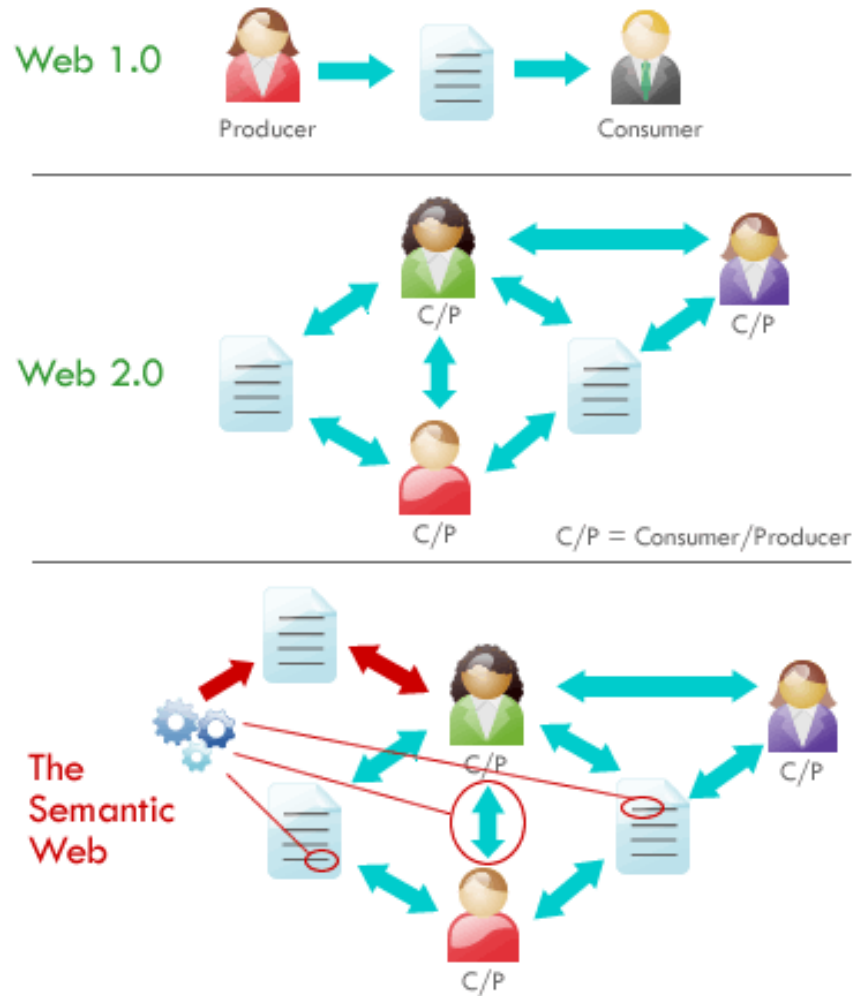


Module MLBDA
Master Informatique
Spécialité DAC

Cours 8 – RDF

Le Web sémantique



Le Web sémantique

Objectifs :

- Rendre le contenu des ressources du Web plus accessibles et plus utilisables.
- Exploiter sémantiquement les données

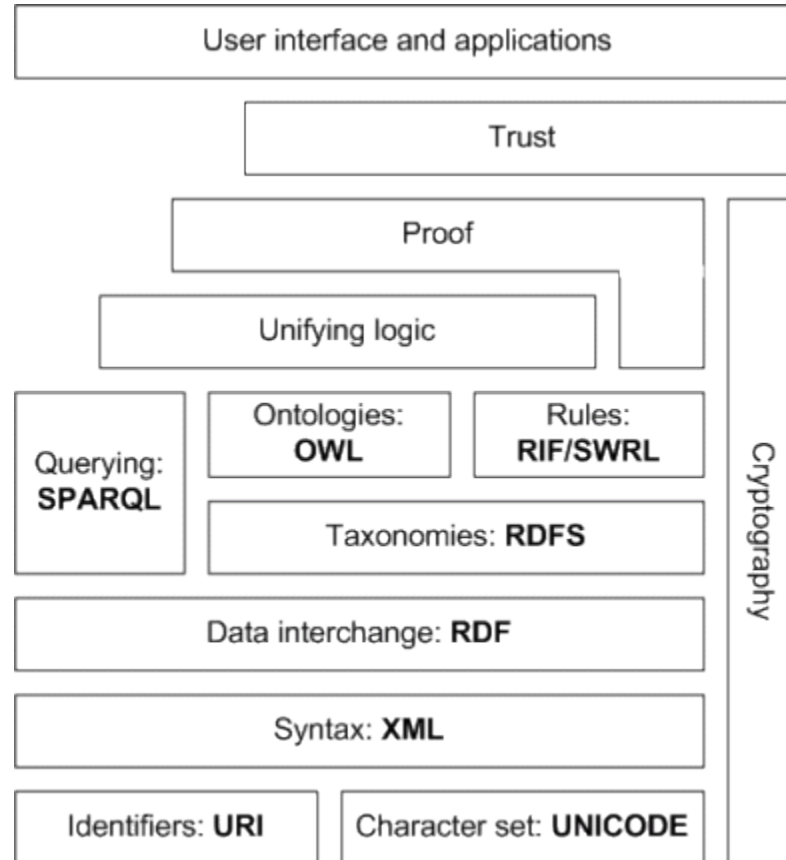
Besoins :

- Représenter la sémantique
- Donner des interprétations riches et exactes aux ressources Web
- Établir des liens entre les données
- Exploiter ces liens
 - Dédire de nouvelles informations
- Indexer et interroger

Standards du Web :

- RDF (Resource Description Framework) : description des données, annotations sémantiques
- RDFS : description des schémas
- SPARQL : interrogation des données RDF

Les standards du Web sémantique



RDF (Ressource Description Framework)

- Premier standard de l'activité Web sémantique du W3C
 - Un modèle de métadonnées : RDF (métadonnées) et RDFS (schémas)
 - Plusieurs syntaxes XML pour l'échange des métadonnées et de leurs schémas (RDFS).
- Recommandations W3C:
 - 1999 : RDF Model and Syntax Specification Recommendation
 - 2004 : RDF Vocabulary Description Language 1.0 : RDF Schema
 - 2004 : RDF Semantics

Niveaux de modélisation

- Niveau physique : ensemble de triplets
- Types de base : ressources, propriétés, déclarations
- Types complexes : collections, listes
- Schémas RDFS : classes, types de propriétés

RDF

- **Déclaration RDF** : triplet (sujet, prédicat, objet) reliant une ressource à une propriété et une valeur

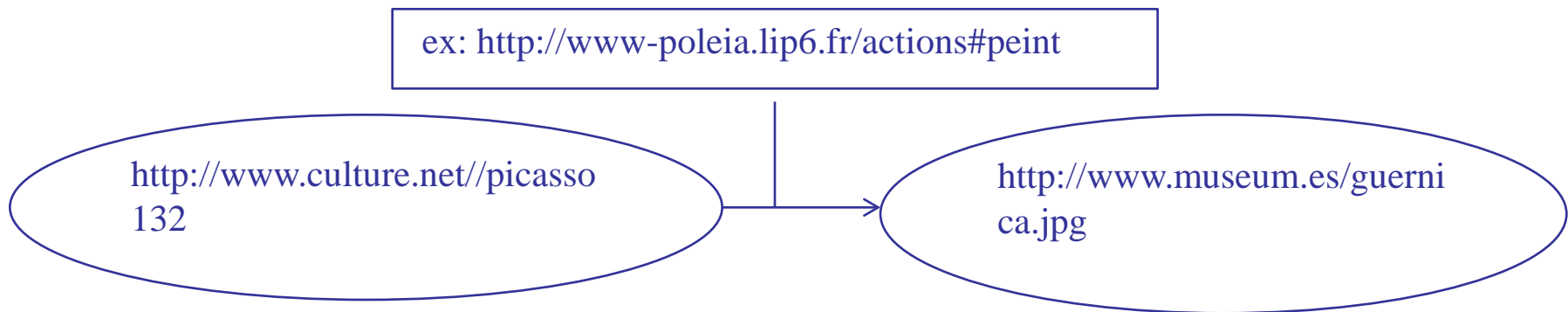


Ex : *Picasso a peint Guernica*



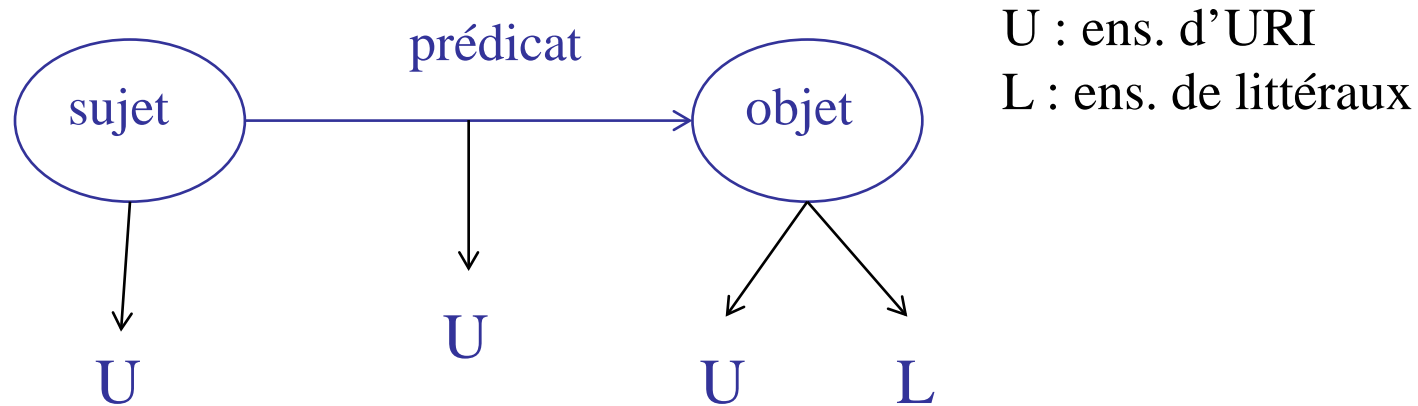
RDF

- Les ressources et les prédicats sont des URI, les objets sont soit des URI, soit des valeurs littérales (chaînes de caractères)



(`http://www.culture.net//picasso132`, `ex:peint`, `http://www.museum.es/guernica.jpg`)

Modèle formel



Un triplet RDF $(s, p, o) \in U \times U \times (U \cup L)$

Un ensemble de triplets RDF est un graphe RDF :

$$G \subseteq U \times U \times (U \cup L)$$

Exemple

(www.culture.net/picasso132, ex:peint, <http://www.museum.es/guernica.jpg>)

(www.culture.net/picasso132, ex:prenom, " Pablo ")

(www.culture.net/picasso132, ex: nom, " Picasso ")

(<http://www.museum.es/guernica.jpg>, ex:exposé, <http://www.museum.es>)

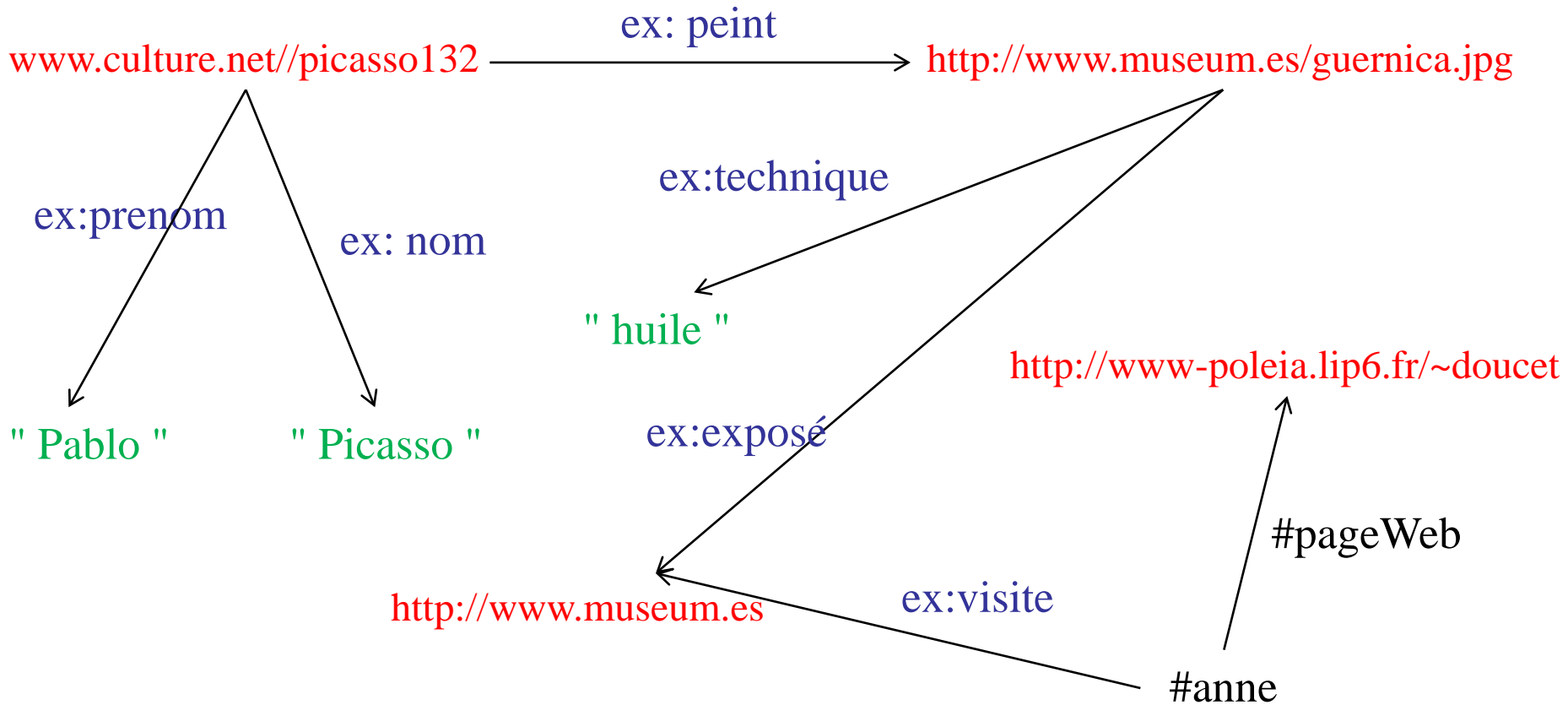
(<http://www.museum.es/guernica.jpg>, ex:technique, " huile ")

(#anne, ex:visite, <http://www.museum.es>)

(#anne, #pageweb, <http://www-poleia.lip6.fr/~doucet>)

- **URI externes** : www.culture.net/picasso132, ex:peint, ex:nom, ex:visite, ex:prenom, ex:exposé, ex:technique, <http://www.museum.es/guernica.jpg>, <http://www.museum.es>, <http://www-poleia.lip6.fr/~doucet>
- **URI locales** : #anne, #pageweb
- **Valeurs** : "Pablo " , " Picasso " , "huile "

Graphe RDF



Modèle de données RDF

- Graphe orienté avec des étiquettes
 - Les nœuds représentent des concepts, des instances, et les valeurs des propriétés
 - Les arcs représentent des propriétés entre concepts
- 3 composants :
 - *Resources* : données décrites en RDF (à l'aide d'expressions RDF et référencées par des URI)
 - *Properties* : définissent les attributs ou relations utilisés pour décrire les ressources.
 - *Statements* : assignent une valeur à une propriété pour une ressource (triplet).

Types de base

Les types de base

- **rdf:Resource**
- **rdf:Property**
- **rdf:Statement**

permettent de distinguer les différentes URL (ressources, propriétés ou déclarations).

Espaces de noms (vocabulaire):

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

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

RDFS (RDF Schema) permet de nommer des classes et des propriétés et de définir leur organisation hiérarchique.

Classes

- Les ressources sont divisées en classes
 - `rdfs:Resource`
 - `rdfs:Class`
 - `rdfs:Literal`
 - `rdfs:Datatype`
 - ...
- Les classes peuvent avoir des sous-classes
 - `rdfs:subclassOf`

Ex:

```
(#oeuvre, rdf:type, rdfs:Class)  
(#peinture, rdf:type, rdfs:Class)  
(#peinture, rdfs:subclassOf, #oeuvre)
```

Propriétés

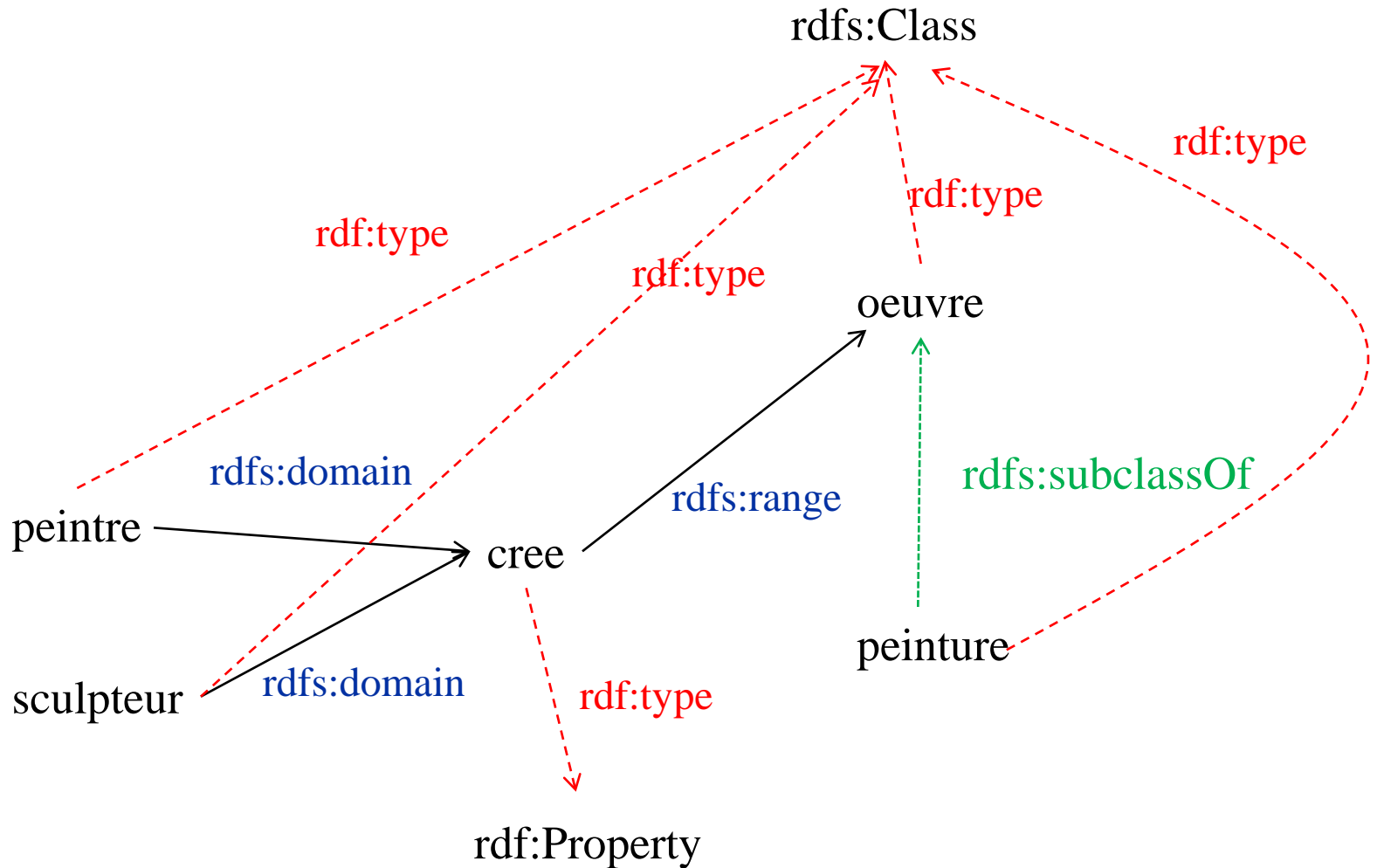
- Les classes peuvent être décrites par des propriétés
- Un type de propriété (`rdf:type`) est une instance de `rdf:Property`
- RDFS permet de restreindre le domaine et le co-domaine d'un type de propriété:
 - `rdfs:subpropertyOf` (`sous-propriété`)
 - `rdfs:domain` (`domaine d'une propriété`) déclare les classes dans lesquelles une propriété prend ses sujets
 - `rdfs:range` (`co-domaine`) déclare les classes dans lesquelles une propriété prend ses valeurs

Un type de propriété peut avoir plusieurs domaines, mais un seul co-domaine

Exemple

```
(#oeuvre, rdf:type, rdfs:Class)
(#peinture, rdf:type, rdfs:Class)
(#peinture, rdfs:subclassOf, #oeuvre)
(#peintre, rdf:type, rdfs:Class)
(#sculpteur, rdf:type, rdfs:Class)
(#cree, rdf:type, rdf:Property)
(#cree, rdfs:range, #oeuvre)
(#cree, rdf:domain, #peintre)
(#cree, rdf:domain, #sculpteur)
```


Exemple



Statement

- `rdf:Statement` : instance de `rdfs:Class`
- `rdf:subject` : instance de `rdf:Property`
 - `S rdf:subject R` indique que S a pour sujet R
- `rdf:predicate` : instance de `rdf:Property`
 - `S rdf:predicate P` indique que le prédicat de S est P
- `rdf:object` : instance de `rdf:Property`
 - `S rdf:object O` indique que O est l'objet de S

Ex : (`www.culture.net/picasso132`, `ex:peint`, `http://www.museum.es/guernica.jpg`)

(`#statement1`, `rdf:subject`, `www.culture.net/picasso132`)

(`#statement1`, `rdf:predicate`, `ex:peint`)

(`#statement1`, `rdf:object`, `http://www.museum.es/guernica.jpg`)

Réification

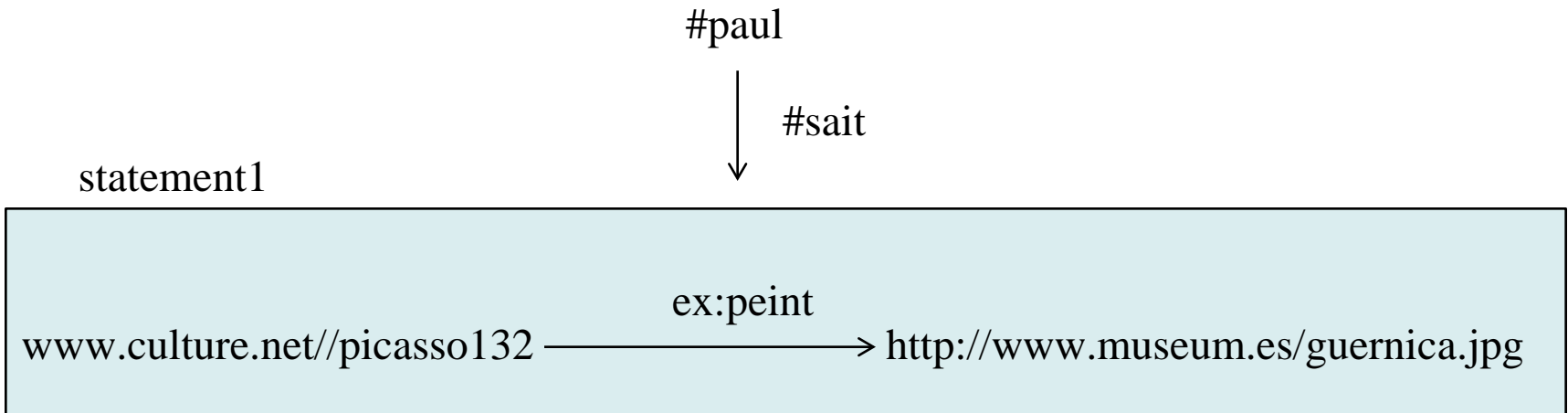
- Une déclaration (statement) peut être identifiée par une URI.
- On peut créer des déclarations en utilisant d'autres déclarations (réification)

(#statement1, rdf:subject, www.culture.net/picasso132)

(#statement1, rdf:predicate, ex:peint)

(#statement1, rdf:object, <http://www.museum.es/guernica.jpg>)

(#paul, #sait, #statement1)



Types Complexes

- **rdfs:Container** permet de représenter des collections, et possède 3 sous-classes :
 - **rdf:Bag** : multi-ensemble de ressources
 - **rdf:Seq** : séquence ordonnée de ressources
 - **rdf:Alt** : énumération de ressources

L'appartenance à une collection est encodée par les propriétés **rdf:_1**, **rdf:_2**, ...

Collections de collections

- Une collection est une ressource, il est possible de construire des collections de collections.

```
(#peintres, rdf:type, rdf:Bag)
(#peintres, rdf:_1, #pablo)
(#peintres, rdf:_2, #paul)
(#artistes, rdf:type, rdf:Bag)
(#artistes, rdf:_1, #auguste)
(#artistes, rdf:_2, #camille)
(#artistes, rdf:_3, #gustave)
(#artistes, rdf:_4, #peintres)
```

Listes

- Une liste est une ressource de type `rdf:List`
- Constructeurs :
 - `rdf:first`
 - `rdf:rest`
 - `rdf:nil`

Ex:

```
(#artistes, rdf:type, rdf:List)
(#artistes, rdf:first, #pablo)
(#artistes, rdf:rest, #sculpteurs)
(#sculpteurs, rdf:type, rdf:List)
(#sculpteurs, rdf:first, #auguste)
(#sculpteurs, rdf:rest, rdf:nil)
```

Ressources anonymes

- Il est possible d'utiliser des ressources non identifiées par une URL. Les identifiants de ces nœuds blancs (*blank node*) sont précédés de _ au lieu du préfixe de nommage. (ex: _abc)
- Une ressource anonyme peut être vue comme une quantification existentielle.
- Une ressource anonyme peut être sujet ou objet d'un triplet.
- Ex:

(#paul, ex:père, _xx)

(_xx, ex:datenaissance, '12 juin')

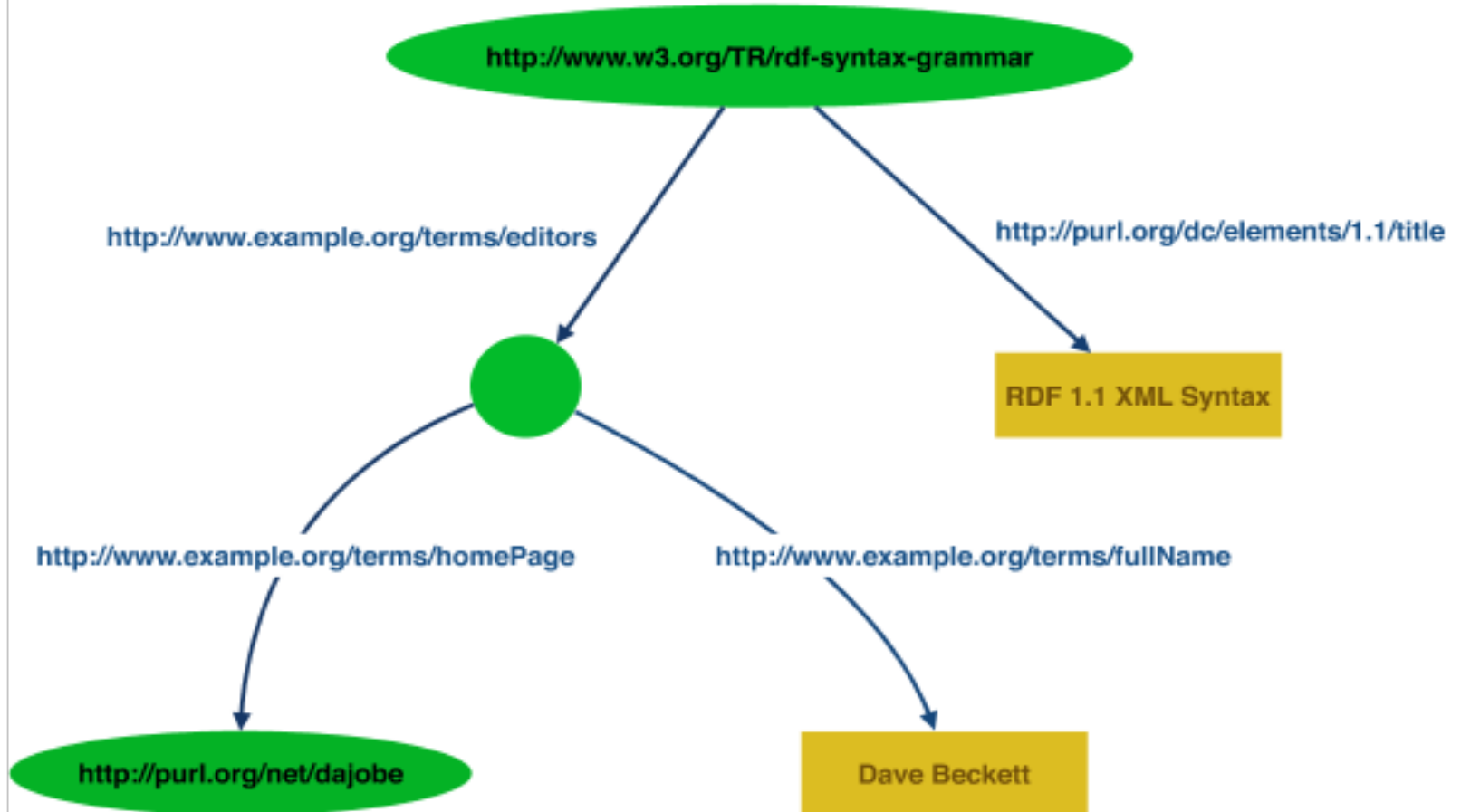
Modélisation avec RDF

- 7 primitives pour les classes, 7 pour les propriétés, 1 pour les instances :
- Classes :
 - `rdf:Statement`, `rdf:Property`, `rdf:Bag`, `rdf:Seq`,
`rdf:Alt`, `rdf:List`, `rdf:XMLLiteral`
- Propriétés :
 - `rdf:first`, `rdf:rest`, `rdf:predicate`,
`rdf:subject`, `rdf:object`, `rdf:type`, `rdf:value`
- Instances :
 - `rdf:nil` (pour décrire une liste vide)

Syntaxes pour RDF

- Plusieurs syntaxes pour RDF
 - N3 : Notation3 (triplets)
 - Ntriples
 - RDFJSON
 - **RDF/XML** : format standard, conçu pour une utilisation par des machines
 - **Turtle** : syntaxe plus concise et compacte
 - ...

Example



Example

```
<rdf:Description >
  <ex:editor>
    <rdf:Description >
      <ex:homePage>
        <rdf:Description >
          </rdf:Description>
        </ex:homePage>
      </rdf:Description>
    </ex:editor>
  </rdf:Description>
```

Ajout des URI

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-  
syntax-grammar" >  
  <ex:editor>  
    <rdf:Description >  
      <ex:homePage>  
        <rdf:Description  
          rdf:resource="http://purl.org/net/dajobe/" >  
        </rdf:Description>  
      </ex:homePage>  
    </rdf:Description>  
  </ex:editor>  
</rdf:Description>
```

Document RDF/XML

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/"
         xmlns:ex="http://example.org/stuff/1.0/">
  <rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-
    grammar"  dc:title="RDF1.1 XML Syntax">
    <ex:editor>
      <rdf:Description ex:fullName="Dave Beckett">
        <ex:homePage rdf:resource="http://purl.org/net/dajobe/" />
      </rdf:Description>
    </ex:editor>
  </rdf:Description>
</rdf:RDF>
```

Description d'une ressource

- Élément **Description**
 - Attribut **about** pour le sujet
 - Sous-élément pour la propriété
 - On peut regrouper dans un même élément **Description** toutes les propriétés dont la ressource est sujet.
- Lorsqu'une propriété contient un littéral chaîne de caractères, on peut exprimer cette propriété comme un attribut

```
<rdf:Description rdf:about="http://www.exemple.org/index.html">  
  <dc:language>en</dc:language>  
</rdf:Description>
```

devient

```
<rdf:Description rdf:about="http://www.exemple.org/index.html"  
  dc:language="en" />
```

Document Turtle

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

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
```

```
@prefix ex: <http://example.org/stuff/1.0/> .
```

```
<http://www.w3.org/TR/rdf-syntax-grammar>
```

```
dc:title "RDF/XML Syntax" ;
```

```
ex:editor [
```

```
ex:fullName "Dave Beckett";
```

```
ex:homePage <http://purl.org/net/dajobe/>
```

```
].
```

Syntaxe Turtle (1)

L'ensemble des **termes RDF** est défini par $T = U \cup L \cup B$ où

U : ensemble des URI

`<http://www.w3.org/TR/rdf-syntax-grammar>`

`dc:title`

L : littéraux RDF (valeurs) : "valeur", @motcle, ^^type

"Dave Beckett", @prefix, "false"^^xsd:boolean

B : nœuds blancs

Un **triplet RDF** est un élément de l'ensemble $(U \cup B) \cup T$

Un **graphe RDF** est un ensemble de triplets RDF.

Syntaxe Turtle (2)

Expression Turtle

- $s\ p\ o$. est une expression Turtle qui désigne l'ensemble de triplets $\{(s, p, o)\} \subset (U \cup B) \times U \times T$
- $s\ p1\ o1; p2\ o2; \dots$ est une expression Turtle qui désigne l'ensemble $\{(s, p1, o1), (s, p2, o2), \dots\}$
- $(e1\ e2\ \dots)$ est une expression Turtle qui désigne une liste RDF $(_b1, rdf:first, e1), (_b1, rdf:rest, _b2), (_b2, rdf:first\ e2), \dots$
- $s\ p1\ o1, o2$ est une abréviation pour $(s\ p1\ o1)\ (s\ p1\ o2)$

Nœuds blancs

- $[_]$ désigne un nouveau nœud blanc $_a$
- $[p1\ o1; p2\ o2; \dots]$ désigne un ensemble de triplets $\{(_x, p1, o1), (_x, p2, o2), \dots\}$

Définition de préfixes

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

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
```

```
@prefix ex: <http://example.org/stuff/1.0/> .
```

Rdf:type

- L'URI <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> peut être abrégée par le token **a**

Ex :

```
@prefix ex:<http://example.org/stuff/1.0/> .
```

```
<www.culture.net//picasso132> a ex:Person .
```

Est un raccourci pour

```
<www.culture.net//picasso132> rdf:type  
ex:Person .
```

Collections

```
@prefix : <http://example.org/stuff/1.0/> .  
:article :auteurs (:paul :marie) .
```

est un raccourci pour

```
@prefix : <http://example.org/stuff/1.0/> .  
:article :auteurs  
[ rdf:first :paul ;  
  rdf:rest [ rdf:first :marie ;  
             rdf:rest rdf:nil ]  
] .
```

Création de propriétés

Définition des propriétés d'une ressource existante

```
<rdf:Description rdf:about='http://www.lip6.fr' >
  <rdf:type>
    <rdf:Description rdf:about='#PageWeb' />
  </rdf:type>
  <dc:author>
    <rdf:Description rdf:about='#jean' />
  </dc:author>
</rdf:Description>
```

Turtle :

```
http://www.lip6.fr rdf:type #pageWeb .
http://www.lip6.fr dc:author #jean .
```

Création de ressources locales

Définition d'une nouvelle ressource locale (avec les propriétés) :

```
<rdf:Description rdf:ID='lip6' >  
<directeur resource='#JC' />  
</rdf:Description>
```

Turtle : #lip6 #directeur #JC .

Remarque : <directeur resource='#JC' />

est une version compacte de

```
<directeur>  
  <rdf:Description rdf:about="#JC" />  
</directeur>
```

Description de nœuds blancs

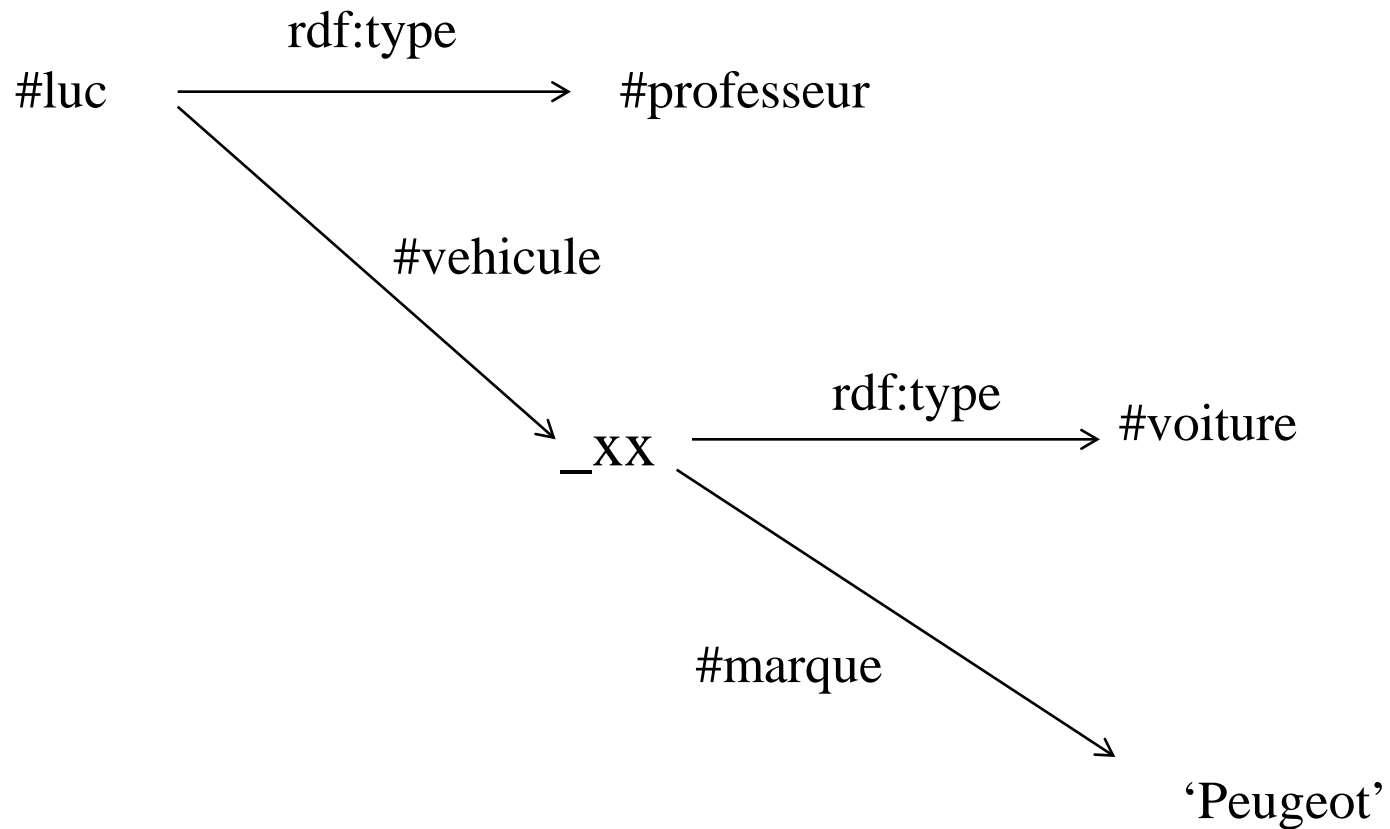
Définition d'une nouvelle ressource locale sans identificateur
(ressource anonyme/noeud blanc) :

```
<rdf:Description rdf:ID='luc' >
  <rdf:type resource='#Professeur' />
  <vehicule>
    <rdf:Description>
      <rdf:type resource='#Voiture' />
      <marque>Peugeot</marque>
    </rdf:Description>
  </vehicule>
</rdf:Description>
```

Turtle :

```
#luc a #Professeur ;
    #vehicule [ a #Voiture ; #marque ``Peugeot'' ] .
```

Exemple



Syntaxe abrégée

Description :

```
<rdf:Description rdf:ID='luc' >
  <rdf:type resource='#Professeur' />
<vehicule>
  <rdf:Description>
    <rdf:type resource='#Voiture' />
    <marque>Peugeot</marque>
  </rdf:Description>
</vehicule>
</rdf:Description>
```

Syntaxe abrégée :

```
<Professeur rdf:ID='luc'>
  <vehicule>
    <voiture marque='Peugeot' />
  </vehicule>
</Professeur>
```

Définition de collections

Définition de collections (syntaxe “containers”) :

```
<rdf:Description rdf:about='#lip6'>
  <members>
    <rdf:Bag rdf:ID='membresLip6'>
      <rdf:li rdf:resource='#amine' />
      <rdf:li rdf:resource='#hubert' />
      <rdf:li rdf:resource='#nelly' />
    </rdf:Bag>
  </members>
</rdf:description>
```

Turtle :

```
#lip6 #members #membresLip6 .
#membresLip6 rdf:first #amine ;
rdf:rest (#hubert #nelly).
```

Ressources et Applications

- [WordNet](#) : base de données de la langue anglaise
- [Dbpedia](#) : source de données RDF extraites de Wikipedia
- [FOAF](#) : Friend of a Friend
- Données gouvernementales :
 - [data.gov.uk](#), [data.gouv.fr](#), [LOGD](#), [Eurostat](#) ...
- Données géographiques
 - [LinkedGeoData](#), [US Census data](#), [INSEE](#),...
- Données bibliographiques
 - [Dublin Core](#), [DBLP](#)
- Médias, vocabulaires techniques, ...

Linked Open Data

- Linked Data (Web des données) : mise en relation des données pour constituer un réseau global (initiative W3C)
 - Standard URL pour nommer les ressources
 - Standard RDF pour décrire les ressources et les relier
 - Les liens peuvent aussi être utilisés par des machines
- Données ouvertes :
 - * Les données sont sur le Web sous licence libre
 - ** Elles sont explicites et structurées
 - *** Dans un format non propriétaire
 - **** Sujets et objets sont identifiés par des URI
 - ***** Les données sont liées à d'autres données

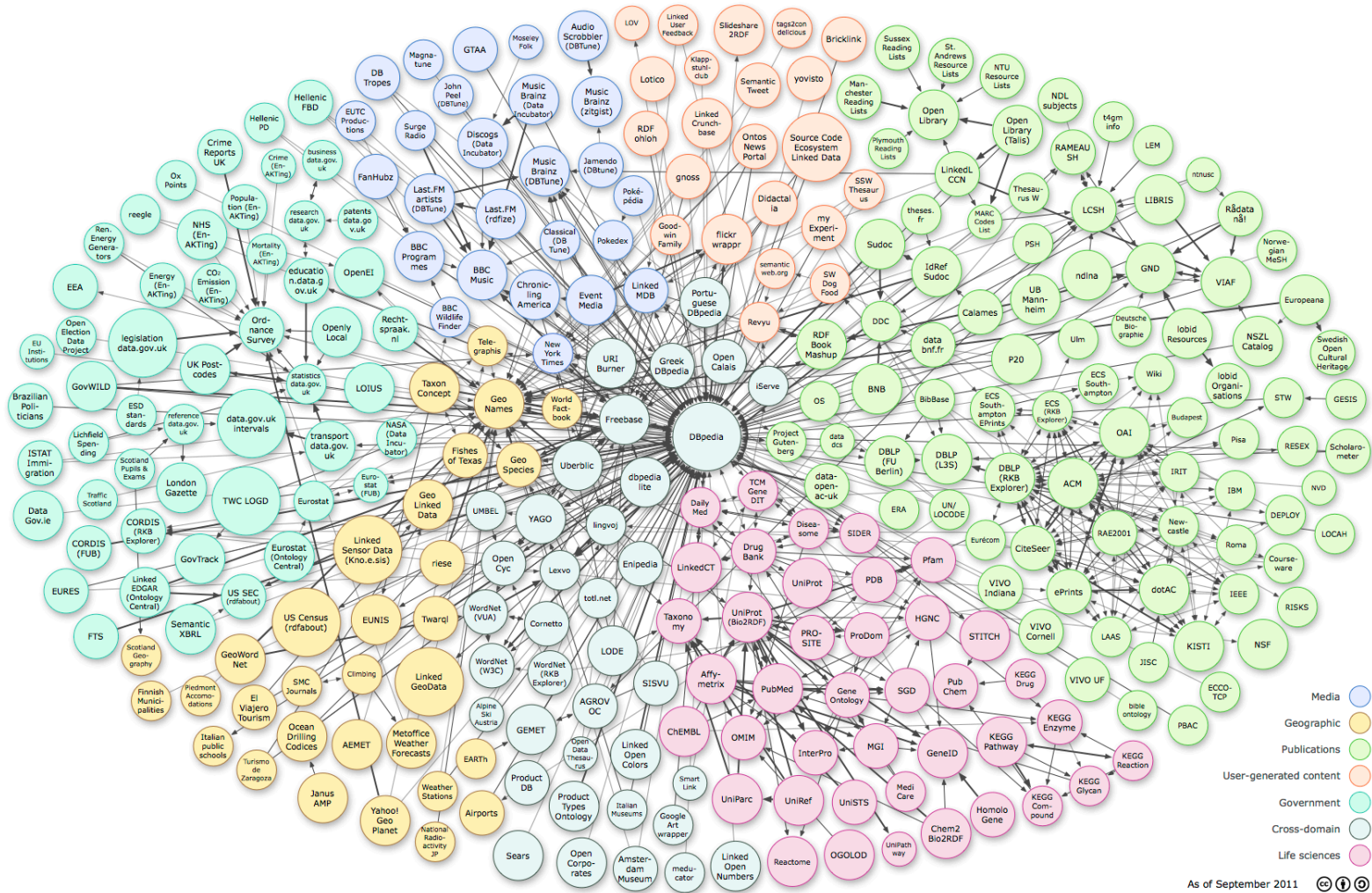
Linking Open Data (LOD)

- En 2007, le projet W3C SWEO (Semantic Web Education and Outreach) a initié le projet LOD visant à relier les sources de données ouvertes.

Domain	Number of datasets	Triples	%	(Out-)Links	%
Media	25	1,841,852,061	5.82 %	50,440,705	10.01 %
Geographic	31	6,145,532,484	19.43 %	35,812,328	7.11 %
Government	49	13,315,009,400	42.09 %	19,343,519	3.84 %
Publications	87	2,950,720,693	9.33 %	139,925,218	27.76 %
Cross-domain	41	4,184,635,715	13.23 %	63,183,065	12.54 %
Life sciences	41	3,036,336,004	9.60 %	191,844,090	38.06 %
User-generated content	20	134,127,413	0.42 %	3,449,143	0.68 %
	295	31,634,213,770		503,998,829	

Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. <http://lod-cloud.net/>
Statistiques de 2011

Linking Open Data



Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch.
<http://lod-cloud.net/>

Conclusion

RDF : modèle de données pour les données semi-structurées (graphes dirigés et étiquetés)

RDFS : modèle de connaissances

- Classification multiple de ressources
- Spécialisation de classes et de propriétés
- Plusieurs domaines possibles pour une propriété

RDF/RDFS permet de créer un graphe par l'union de descriptions de ressources.

- Une classe RDFS (un concept) est définie par un nom, un ensemble de propriétés et sa position dans une hiérarchie définie par la relation **rdfs :subClassOf**.
- L'intégration et le raisonnement sont limités à l'utilisation des types de propriétés et des relations **rdfs :subClassOf** et **rdfs :subPropertyOf**.

Conclusion

OWL est une recommandation W3C , qui définit une famille de langages d'ontologies (plus expressifs que RDFS) qui

- facilitent l'intégration et l'évolution d'ontologies et de métadonnées,
- permettent de définir des concepts complexes,
- permettent de vérifier la cohérence de ces définitions et
- **d'inférer des relations sémantiques** (inclusion, équivalence) entre les concepts.

Interroger les schémas RDF : SPARQL