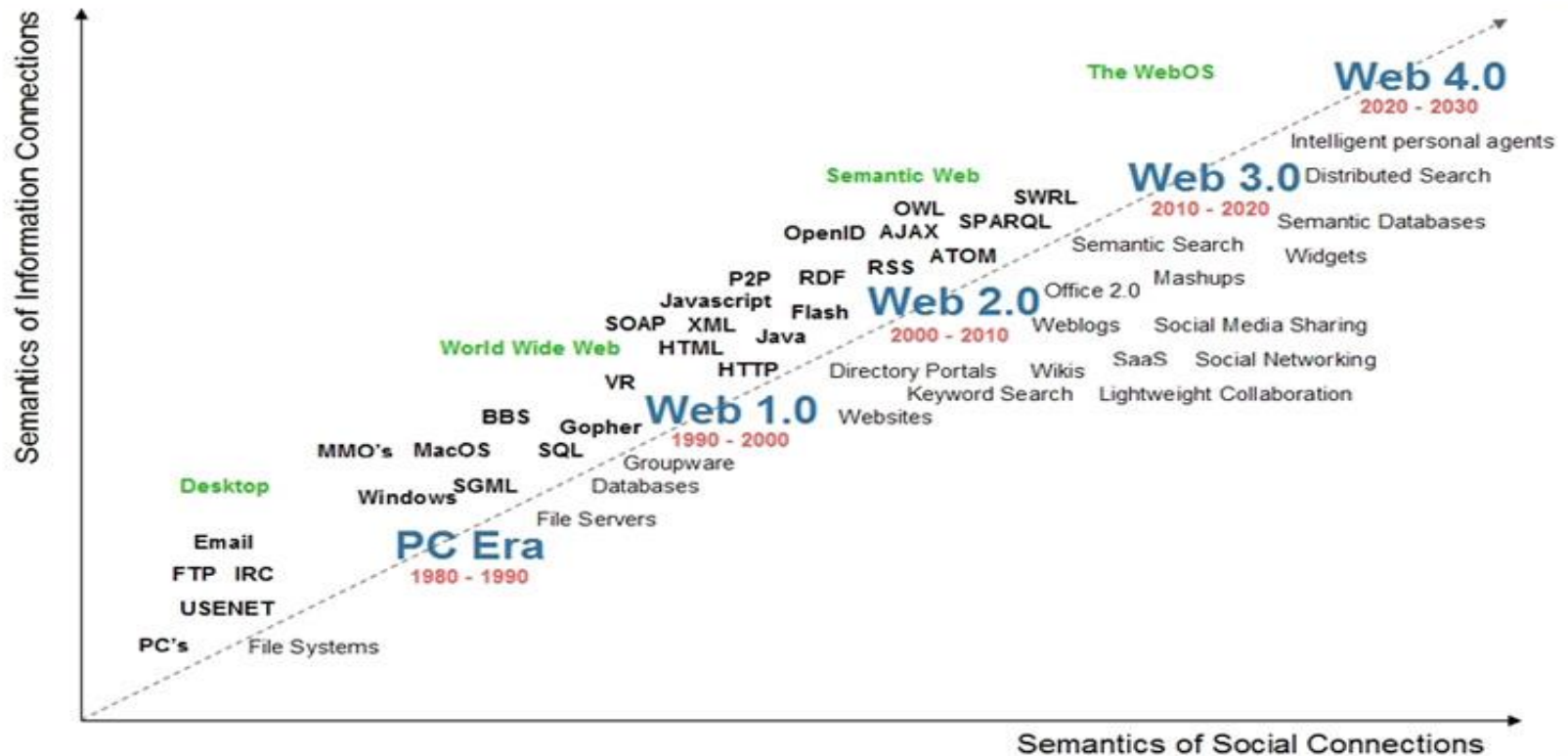


Module MLBDA
Master Informatique
Spécialité DAC

COURS 8 – RDF

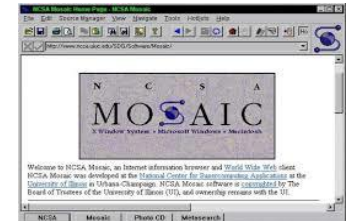
Evolution des technologies



HTML – publier et connecter des documents (texte, images)

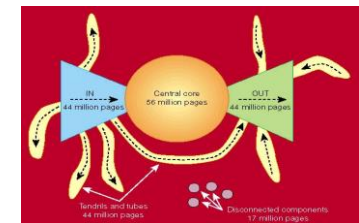
Technologies / Standards

- HTML (HyperText Markup Language) : texte structuré par des balises de présentation
- URI/URL (Uniform Resource Identifier/Locator) : identification universelle de ressources
- HTTP (Hypertext Transfer Protocol) : accès distribué (TCP/IP), serveurs web (Apache, IIS, nginx)
- CGI (Common Gateway Interface), PHP, ASP : langages de scripts



Services / Utilisation

- publication et référencement de contenus
- saisie de données par formulaires et génération dynamique de pages
- moteurs de recherche : Google, Yahoo !, AltaVista, ...



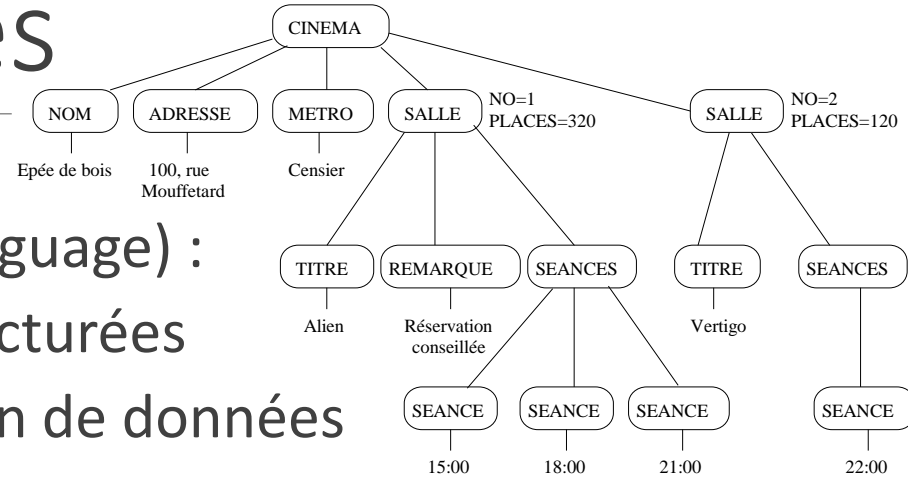
XML - publier des données semi-structurées

Technologies / Standards

- XML (eXtended Markup Language) :
- modèles données semi-structurées
- XPath/XQuery : interrogation de données
- XSLT : transformation de données

Services / Utilisation

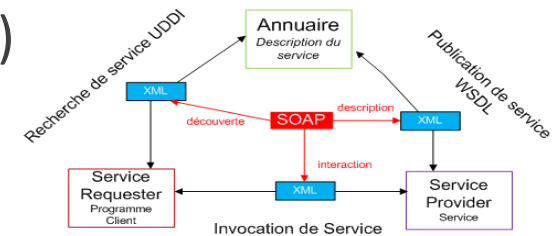
- intégration de données : entrepôts XML
- moteurs de recherche XML, RI structurée
- publication et échange de données semi-structurées



Services Web - connecter des services de données

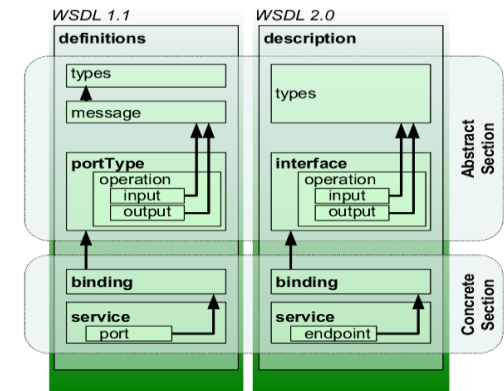
Technologies / Standards

- WSDL (Web Services Description Language) : définition d'interfaces structurées de services
- SOAP (Simple Object Access Protocol) : échange de données/messages structurés (appels de services web)
- UDDI (Universal Description, Discovery and Integration) : catalogue de services web



Services / Utilisation

- intégration d'applications
- workflows distribués



Web Social - connecter les personnes

Technologies (Standards)

- plate-formes de publication (Twitter) ou de partage de données personnelles (Facebook)
- représentation explicite de liens sociaux entre utilisateurs

Services / Utilisation

- espaces personnalisés de partage de données
- recommandation, filtrage collaboratif
- analyse de données (profils utilisateurs , traces)



(c) www.touchgraph.com

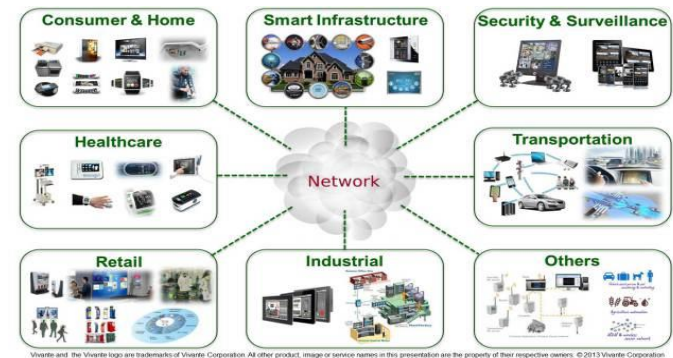
Web des Objets - connecter des objets

Technologies

- objets communicants (RFID)
- capteurs intelligents
- réseaux ubiquitaires

Services / Utilisation

- infrastructures “intelligentes” (SMART)
- transports, santé et aide médical, gestion
- d'énergie, chaînes de production,
- environnement



(c) tomorrowpaper.wordpress.com

Défis communs du Web

Hétérogénéité :

- services : accès, interface, performance
- données : structure, sémantique

Distribution :

- données / services

Qualité :

- services: disponibilité, performance
- Données: complétude, redondance, fraîcheur

Sécurité et vie privé

Passage à l'échelle :

- utilisateurs, pages, services, objets, données

XML : Modèle de Données Web ?

XML permet de

- représenter, échanger des données structurées sur le Web (sérialisation)
- intégrer des données hétérogènes (modèle de données semi-structuré)
- interroger et transformer des données avec des langages de haut niveau (XPath, XQuery)
- utiliser et adapter technologies existantes (ex. SGBD relationnels)

XML est

- un standard largement adopté et utilisé
- expressif et flexible

Mais: XML est encore limité et ne répond pas à tous les besoins de modélisation et d'intégration de ressources Web.

Un cinéma en XML

CINEMA1.XML

```
<Cinema>
  <nom>Le St. André</nom>
  <rue>
    13, rue St. André des Arts
  </rue>
  <film seance='18'>
    <genre>comédie</genre>
    <titre>Brazil</titre>
    <role>R. de Niro</role>
  </film>
</Cinema>
```

CINEMA2.XML

```
<Cinema nom='Le St. André'>
  <adresse>
    <no>13</no>
    <rue>St. André des Arts</rue>
  </adresse>
  <comédie heure='18'>
    <titre>Brazil</titre>
    <acteur>
      <prenom>Robert</prenom>
      <nom>de Niro</nom>
    </acteur>
  </comédie>
</Cinema>
```

Les limites du modèle XML

XML

Traitement syntaxique des noms

XML ne fait pas de distinction entre

- entités (`cinema`, `film`),
- propriétés (`nom`) et
- associations (`role`).

Multiples interprétations possibles de la relation élément/sous-élément:

- composition,
- héritage, ...

OBJECTIFS

Faciliter l'interprétation de ressources publiées sur le Web :

- `comédie` est un genre de film
- `seance` et `heure` sont des synonymes
- `rue` fait partie d'une adresse
- `R. de Niro` et `Robert de Niro` sont la même personne

Le Web Sémantique

Objectifs :

- Rendre le contenu des ressources du Web plus accessibles
- Mieux décrire l'information du Web

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

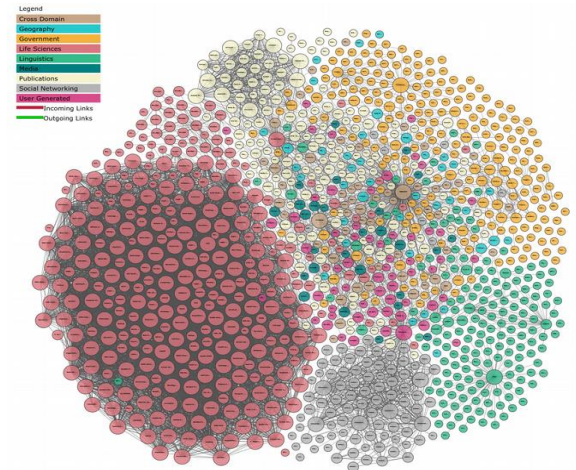
Web Sémantique - connecter des ressources et des connaissances

Technologies / Standards

- RDF (Resource Description Framework)
- SPARQL : interrogation de données RDF
- entrepôts RDF / SPARQL endpoints

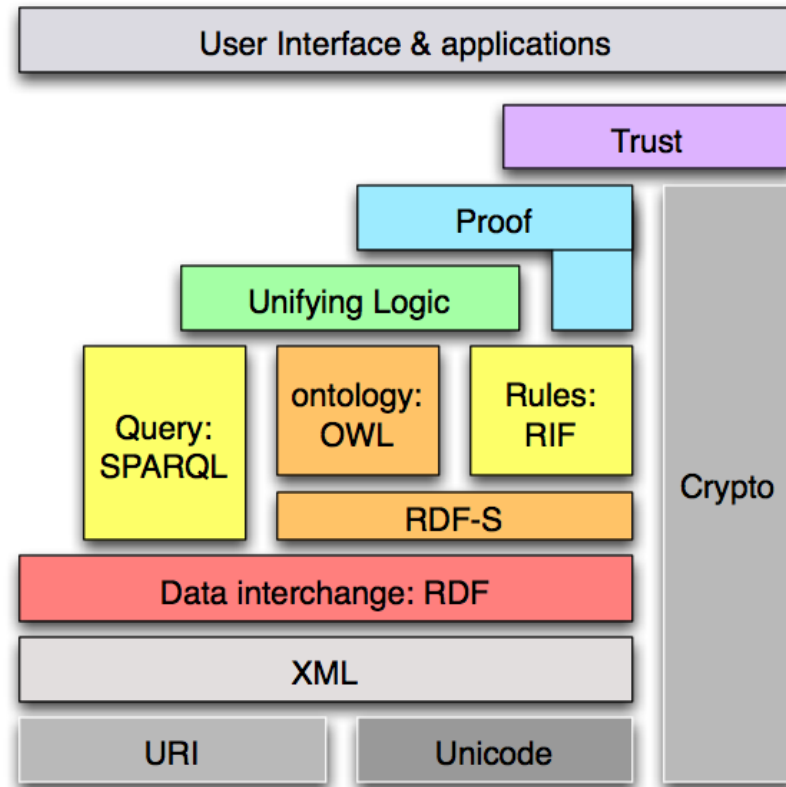
Services / Utilisation

- modélisation de données et de connaissances («données sémantique »)
- interrogation et raisonnement (« interrogation sémantique »)
- publication et intégration de données et de connaissance



(c) "Linking Open Data cloud diagram 2017, by Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch and Richard Cyganiak. <http://lod-cloud.net/>"

Les standards du Web sémantique



(c) Tim Berners-Lee

RDF (Ressource Description Framework)

Premier standard de l'activité Web sémantique du W3C

- http://www.w3.org/standards/techs/rdf#w3c_all
- Un modèle de métadonnées : RDF (métadonnées) et RDFS (schémas)
- Plusieurs syntaxes : XML, Turtle, triplets, JSON, ...
- Sémantique formelle

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

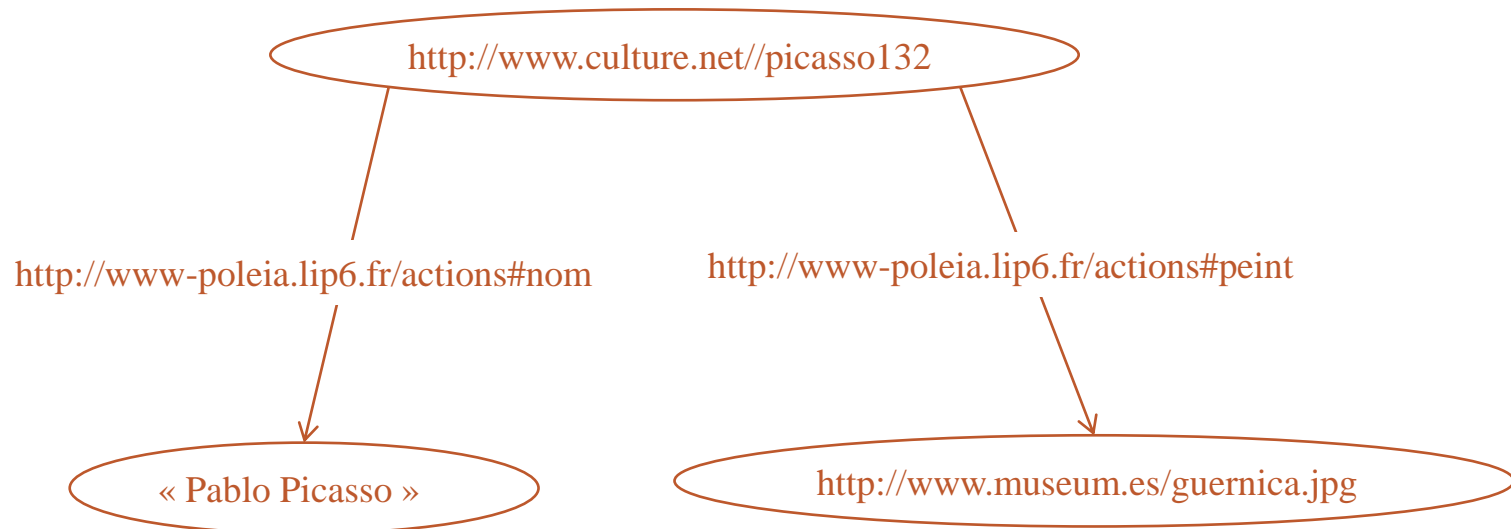
Déclaration RDF : triplet (sujet, prédicat, objet) reliant une ressource à une propriété et une valeur



Ex : Picasso a peint Guernica

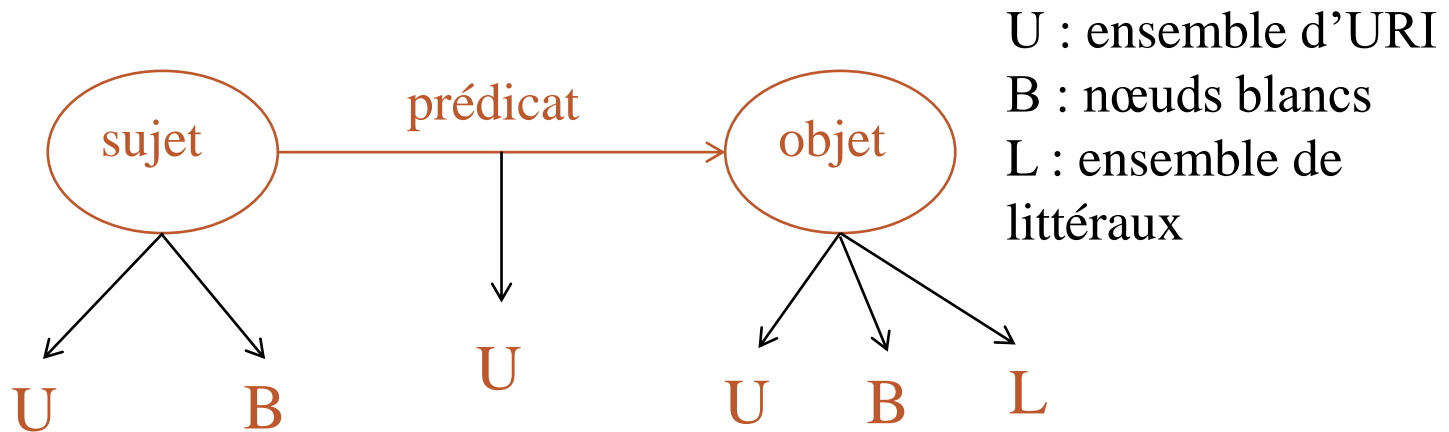


RDF



(<http://www.culture.net/picasso132>, <http://www-poleia.lip6.fr/actions#peint>, <http://www.museum.es/guernica.jpg>)

Modèle formel



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

Un « dataset » RDF : $G \subseteq (U \cup B) \times U \times (U \cup L \cup B)$

Dataset RDF = ensemble de triplets

<http://www.asws.com/rois#r1><http://www.asws.com/rois#fils><http://www.asws.com/rois#r3>
<http://www.asws.com/rois#r1><http://www.asws.com/rois#nom>"Francois I"
<http://www.asws.com/rois#r2><http://www.asws.com/rois#epouse><http://www.asws.com/rois#r8>
<http://www.asws.com/rois#r2><http://www.asws.com/rois#nom>"Charles IX"
<http://www.asws.com/rois#r3><http://www.asws.com/rois#epouse><http://www.asws.com/rois#r5>
<http://www.asws.com/rois#r3><http://www.asws.com/rois#nom>"Henri II"
<http://www.asws.com/rois#r4><http://www.asws.com/rois#epouse><http://www.asws.com/rois#r6>
<http://www.asws.com/rois#r4><http://www.asws.com/rois#nom>"Francois II"
<http://www.asws.com/rois#r5><http://www.asws.com/rois#epoux><http://www.asws.com/rois#r3>
<http://www.asws.com/rois#r5><http://www.asws.com/rois#fils><http://www.asws.com/rois#r4>
<http://www.asws.com/rois#r5><http://www.asws.com/rois#fils><http://www.asws.com/rois#r2>
<http://www.asws.com/rois#r5><http://www.asws.com/rois#fils>_:genid1
<http://www.asws.com/rois#r5><http://www.asws.com/rois#nom>"Catherine de Medici"
<http://www.asws.com/rois#r6><http://www.asws.com/rois#epoux><http://www.asws.com/rois#r4>
<http://www.asws.com/rois#r6><http://www.asws.com/rois#nom>"Mary Stuart"
<http://www.asws.com/rois#r8><http://www.asws.com/rois#epoux><http://www.asws.com/rois#r2>
<http://www.asws.com/rois#r8><http://www.asws.com/rois#nom>"Elisabeth d Autriche"
_:genid1 <http://www.asws.com/rois#nom> "Henry III«

_:genid1 : nœud blanc (ressource non-identifiée)

Espace de noms

Un espace de nom est défini par un préfixe et une URI (le préfixe peut être vide):

@prefix **asws**: http://www.asws.com/rois# .

@prefix : http://www.asws.com/test# .

Le préfixe permet de désigner une URI qui appartient au vocabulaire désigné par l'URI de l'espace de nom:

asws:r1 asws:fils asws:r3 .

asws:r1 asws:nom "Francois I" .

asws:r2 asws:epouse asws:r8 .

asws:r2 asws:nom "Charles IX" .

asws:r3 asws:epouse asws:r5 .

asws:r3 asws:nom "Henri II" .

asws:r4 asws:epouse asws:r6 .

asws:r4 asws:nom "Francois II" .

asws:r5 asws:epoux asws:r3 .

...

:r6 asws:epoux asws:r4 .

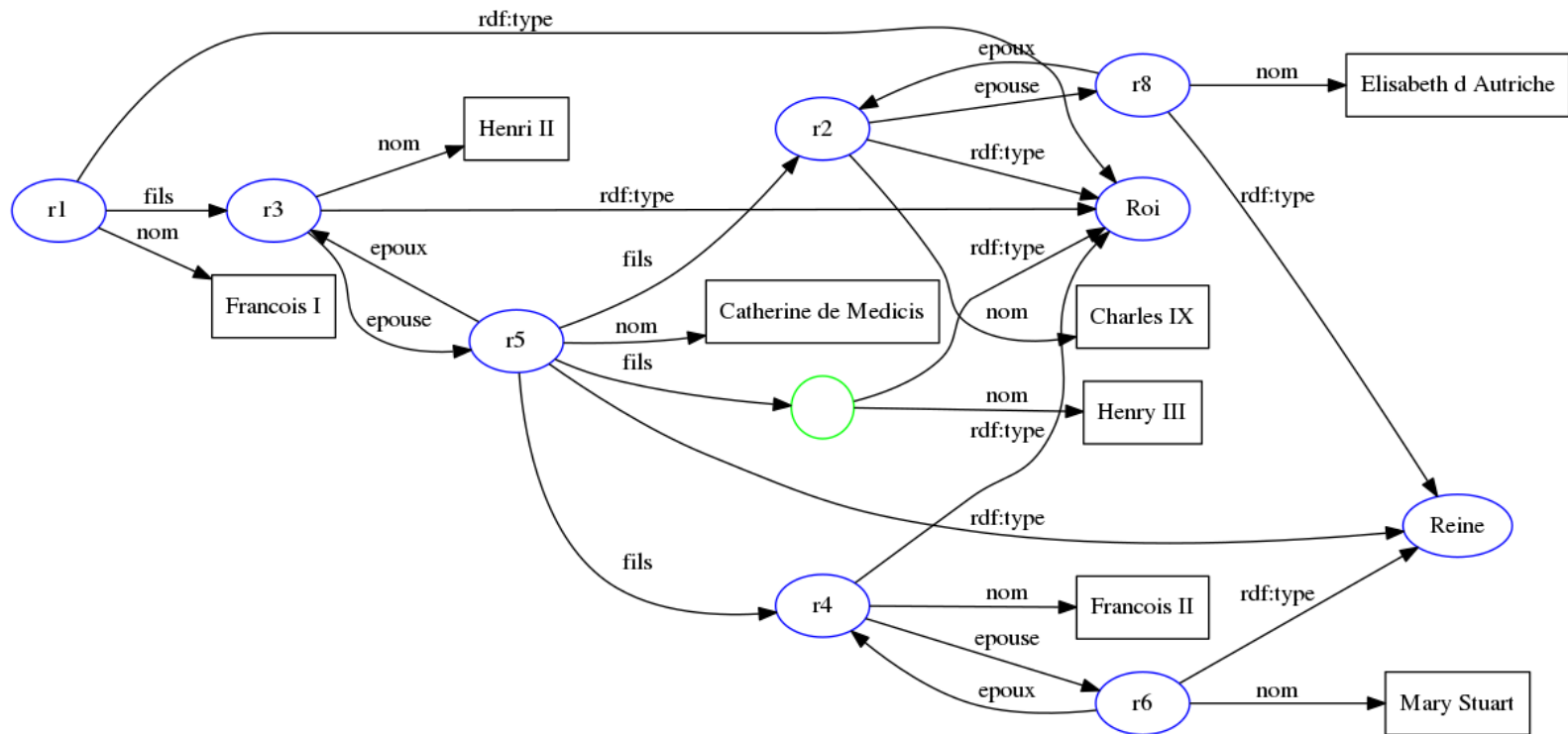
:r6 asws:nom "Mary Stuart" .

:r8 asws:epoux asws:r2 .

:r8 asws:nom "Elisabeth d Autriche" .

_:genid1asws:nom "Henry III" .

Dataset RDF = graphe



- nœuds identifiés : r1, r2, r3, ... (ovales bleus) et non identifiées (cercles verts)
- arcs étiquetés : fils, nom, époux, ...
- littéraux typées (rectangles) : 'François I', 'Mary Stuart',...

Modèle RDF

Espaces de noms RDF:

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

Types de base

- rdf:Resource (ressources) : ensemble des URI (nœuds et étiquettes d'arcs)
- rdf:Property (propriétés) : ensemble des URIs qui identifient des étiquettes d'arcs (propriétés)
- rdf:Statement (déclarations) : triplet (sujet, propriété, objet)

Associer une URI à un type:

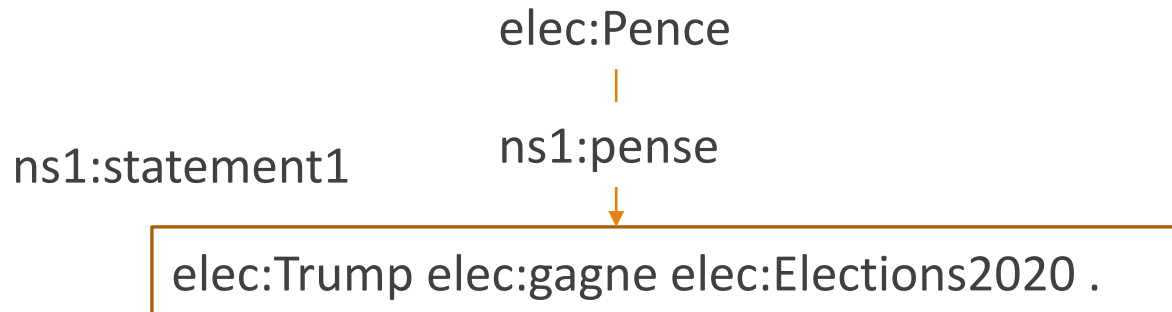
- rdf:type

Exemples:

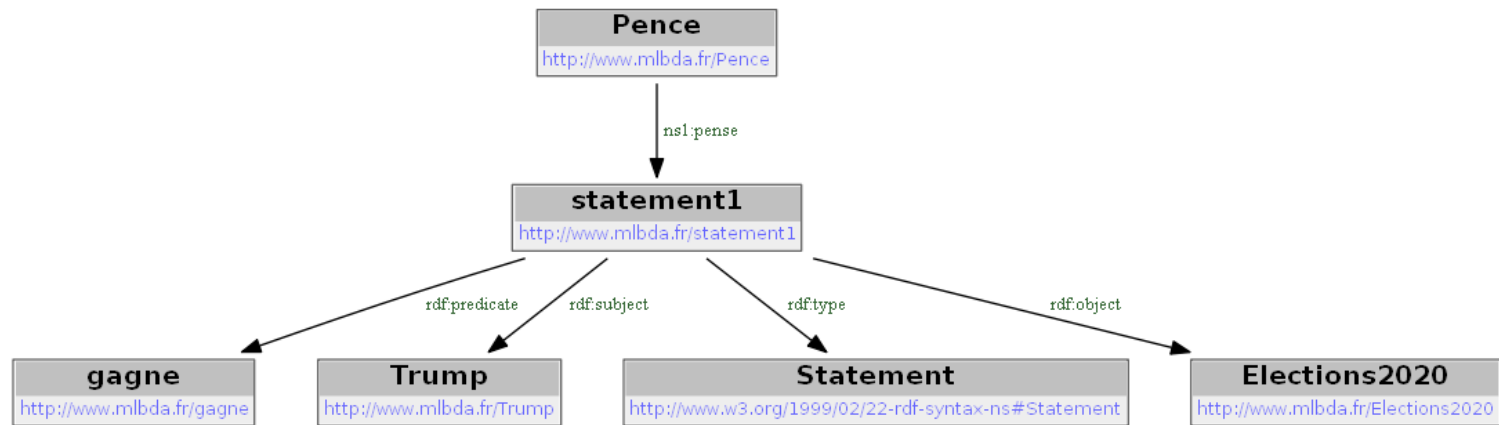
- asws:r1 rdf:type rdf:Resource .
- asws:epoux rdf:type rdf:Resource .
- asws:epoux rdf:type rdf:Property .

Déclarations et réification

```
ns1:statement1 rdf:type rdf:Statement .  
ns1:statement1 rdf:subject ns1:Trump .  
ns1:statement1 rdf:predicate elec:gagne .  
ns1:statement1 rdf:object elec:Elections2020 .  
elec:Pence ns1:pense ns1:statement1 .
```



Réification



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 (non ordonnées)

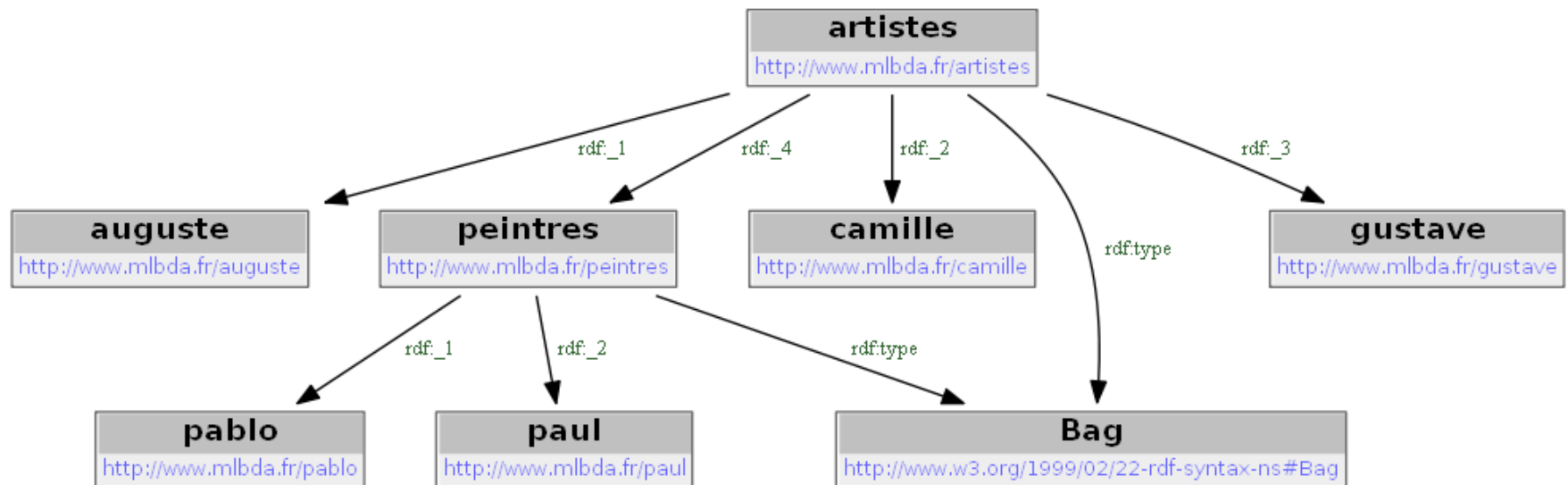
L'appartenance à une collection est encodée par les propriétés rdf:_1, rdf:_2, ...

Collections de collections

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

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

Collections de collections



Listes RDF

Une liste est une ressource de type `rdf:List`

Constructeurs :

- `rdf:first`
- `rdf:rest`
- `rdf:nil`

Ex:

@prefix : <http://www.mlbd.fr/> .

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

:artistes rdf:type rdf:List .

:artistes rdf:first :pablo .

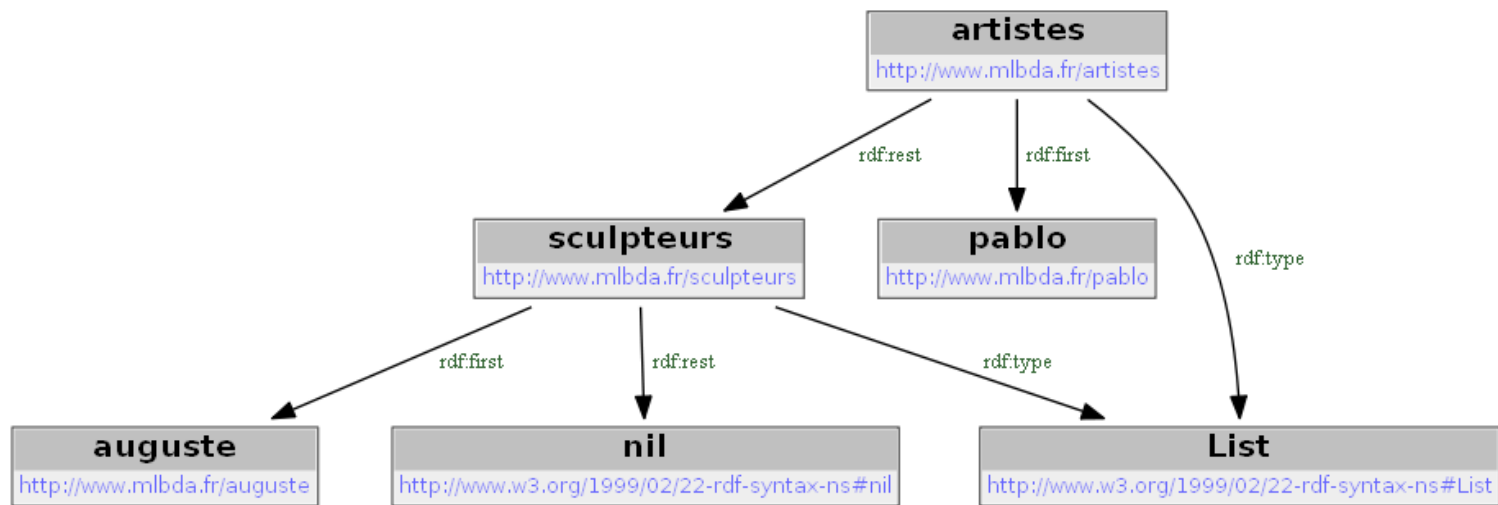
:artistes rdf:rest :sculpteurs .

:sculpteurs rdf:type rdf:List .

:sculpteurs rdf:first :auguste .

:sculpteurs rdf:rest rdf:nil .

Listes RDF



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.

Resource anonyme (nœud blanc)

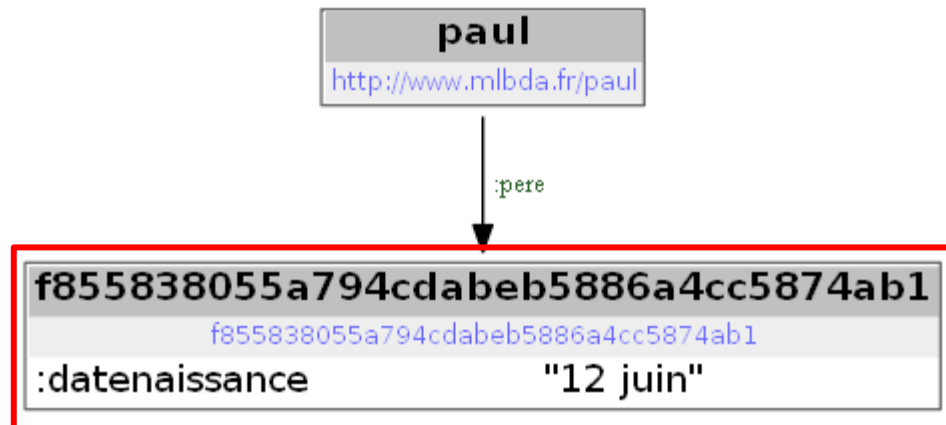
@prefix : <http://www.mlbd.fr/> .

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

:paul :pere _:xx .

_:xx :datenaissance "12 juin" .

Nœud blanc



Modélisation avec RDF

8 primitives pour les classes, 7 pour les propriétés, 1 pour les instances :

Classes :

- `rdf:resource`, `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)

Schémas RDFS : Classes

Espace de noms RDFS:

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

RDFS (RDF Schema) permet de définir des classes :

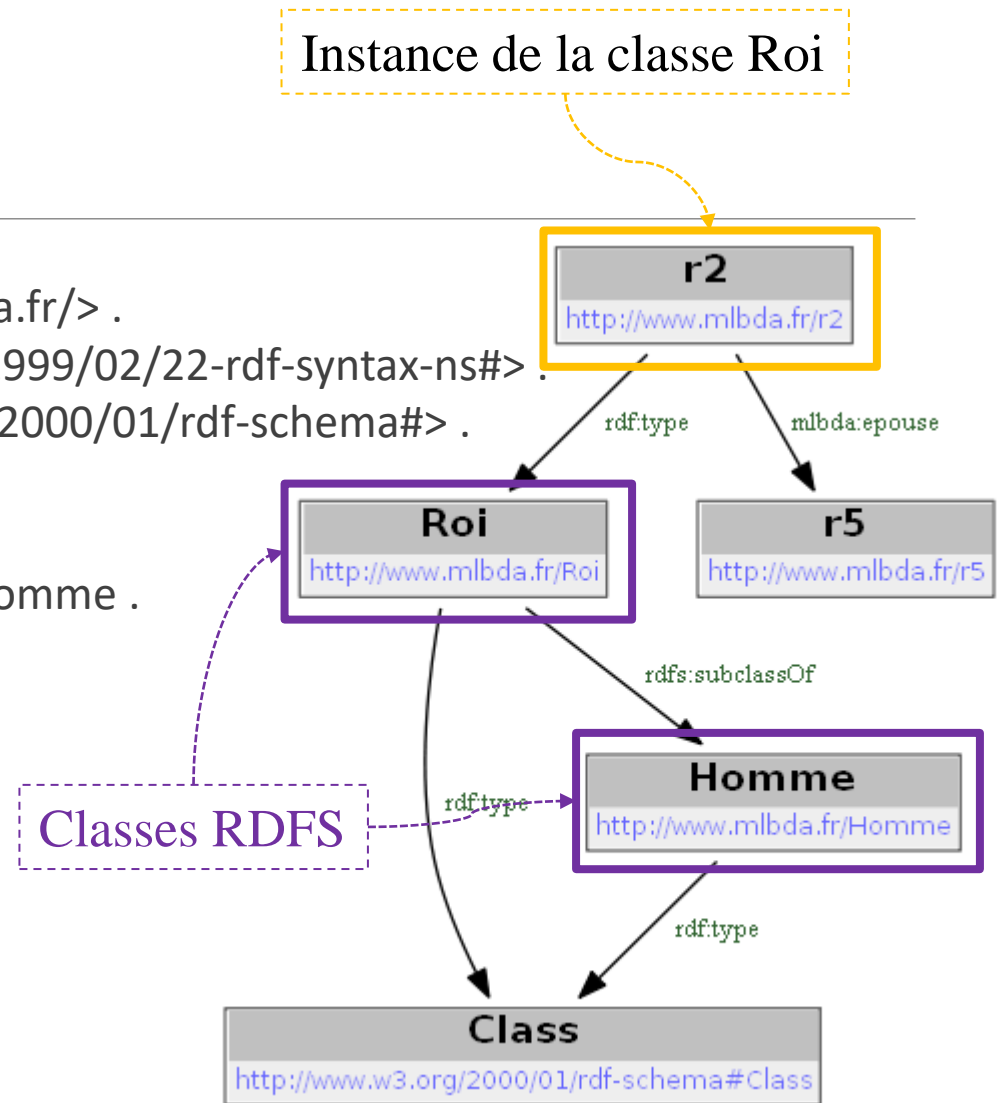
- rdfs:Class
- Les instances d'une classe sont définies avec la propriété rdfs:type
- Une classe est définie comme une instance de la classe rdfs:Class

Les classes peuvent avoir des sous-classes

- rdfs:subclassOf

Classes RDFS

```
@prefix mlbda: <http://www.mlbda.fr/> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
mlbda:Roi rdf:type rdfs:Class .  
mlbda:Homme rdf:type rdfs:Class .  
mlbda:Roi rdfs:subclassOf mlbda:Homme .  
mlbda:r2 rdf:type mlbda:Roi .  
mlbda:r2 mlbda:epouse mlbda:r5 .
```



Propriétés RDFS

Les classes peuvent être décrites par des propriétés

- `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 valeur

On peut définir une hiérarchie de propriétés:

- `rdfs:subpropertyOf` (sous-propriété)

Propriétés RDFS

@prefix asws: <http://www.mlbd.fr/> .

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

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

asws:enfant rdf:type rdf:Property .

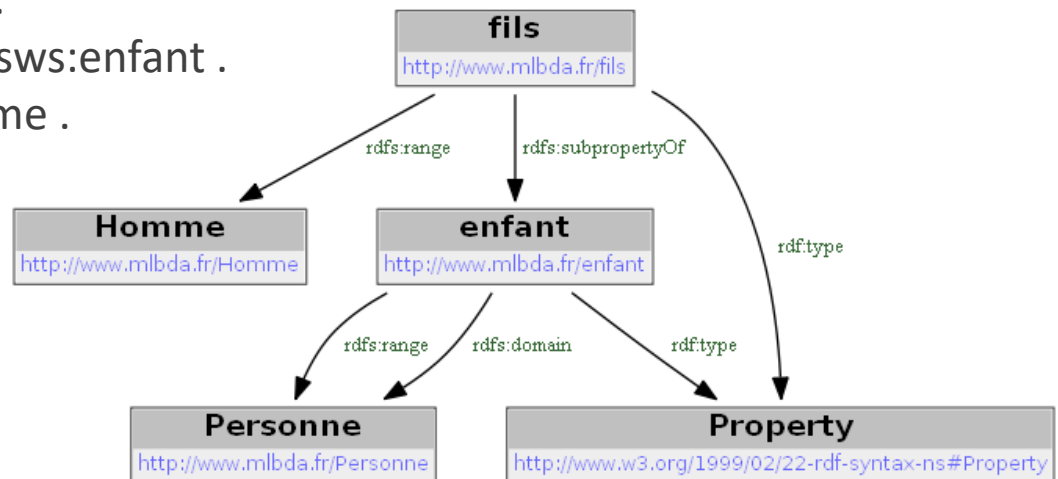
asws:enfant rdfs:domain asws:Personne .

asws:enfant rdfs:range asws:Personne .

asws:fils rdf:type rdf:Property .

asws:fils rdfs:subpropertyOf asws:enfant .

asws:fils rdfs:range asws:Homme .



Exemples de schémas RDFS

Exemples de schémas RDFS (vocabulaire de classes et propriétés):

- @prefix foaf: <<http://xmlns.com/foaf/0.1/>> .
- @prefix dc: <<http://purl.org/dc/elements/1.1/>> .
- @prefix xsd: <<http://www.w3.org/2001/XMLSchema#>> .

Langages RDF

Plusieurs syntaxes pour RDF

- N3 / Turtle : Notation Turtle
- nt/ ntriples : notation triplets
- RDFJSON : notation JSON
- RDF-XML : notation XML (syntaxe étendue)
- RDF-XML/Abrev : notation XML (syntaxe abrégée)

Syntaxe XML

Vocabulaire XML pour définir des ressources et des propriétés:

Élément racine <rdf:RDF>

- Définition des espaces de noms

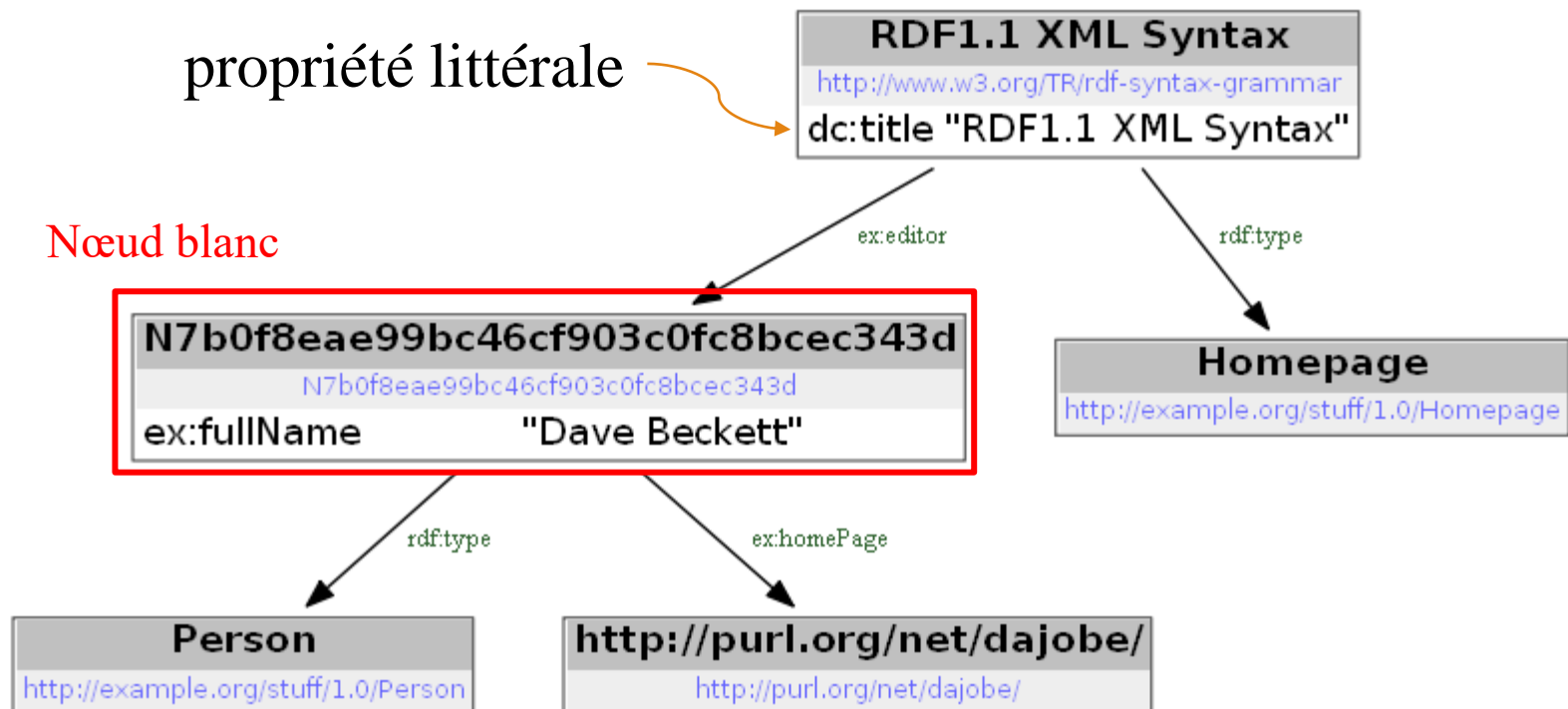
Définition d'une *ressource* (sujet ou objet) :

- Élément XML <rdf:Description>
- Identification:
 - attribut `rdf:about` : URI de la nouvelle ressource
 - attribut `rdf:resource` : URI d'une ressource existante (référence)
 - Si aucun des deux attributs est présent: nœud « blanc » (non identifiés)
- Sous-éléments : propriétés de la ressource

Définition d'un *propriété* :

- Élément XML <nom_propriété>
 - text (valeur littérale)
 - attribut `rdf:resource` avec une référence vers un élément <rdf:Description>
 - sous-élément <rdf:Description> identifié ou non-identifié

Exemple



Exemple RDF/XML

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:ex="http://example.org/stuff/1.0/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
    <ex:editor>
      <rdf:Description >
        <rdf:type rdf:resource="http://example.org/stuff/1.0/Person"/>
        <ex:fullName>Dave Beckett</ex:fullName>
        <ex:homePage rdf:resource="http://purl.org/net/dajobe"/>
      </rdf:Description>
    </ex:editor>
    <rdf:type rdf:resource="http://example.org/stuff/1.0/Homepage"/>
    <dc:title>RDF1.1 XML Syntax</dc:title>
  </rdf:Description>
</rdf:RDF>
```



Nœud blanc

Syntaxe étendue et syntaxe abrégé:

Règles de simplification:

1. Lorsqu'un sous-élément propriété contient un **littéral**, on peut exprimer cette propriété comme un attribut.
2. On peut remplacer la balise `<rdf:Description>` par une classe à laquelle la ressource appartient.

Exemple RDF/XML

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:ex="http://example.org/stuff/1.0/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>
  <ex:Homepage rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
    <ex:editor>
      <ex:Person>
        <ex:fullName>Dave Beckett</ex:fullName>
        <ex:homePage rdf:resource="http://purl.org/net/dajobe"/>
      </ex:Person>
    </ex:editor>
    <dc:title>RDF1.1 XML Syntax</dc:title>
  </ex:Homepage>
</rdf:RDF>
```



Nœud blanc

Syntaxe Turtle

@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> a ex:Homepage ;

ex:editor [a ex:Person ;

ex:fullName "Dave Beckett" ;

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

dc:title "RDF1.1 XML Syntax" .

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
 - @motcle : information sur la valeur: langue, monnaie, encodage, ... (*optionnel*)
 - ^^type : type XML Schema (*optionnel*)
 - "RDF1.1 XML Syntax"@en, "Dave Beckett", "false"^^xsd:boolean
- B : nœuds blancs

Un triplet RDF est un élément de l'ensemble $(U \cup B) \times U \times (U \cup L \cup B)$

Un graphe RDF est un ensemble de triplets RDF.

Syntaxe Turtle (2)

Expression Turtle → triplets représentés

`s p o .` → (s, p, o)

`s p1 o1; p2 o2;` → (s, p1, o1), (s, p2, o2), ... - *factorisation sujet*

`s p1 o1, o2` → (s p1 o1) (s p1 o2) ... - *factorisation sujet et propriétés*

`(e1 e2 ...)` → (`_b1`, `rdf:first`, e1), (`_b1`, `rdf:rest`, `_b2`), (`_b2`, `rdf:first`, e2), ... - *liste RDF*

Nœuds blancs

`[]` → `_:a`

`[p1 o1; p2 o2; ...]` → (`_:x`, p1, o1), (`_:x`, p2, o2), ...

Définition de préfixes

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

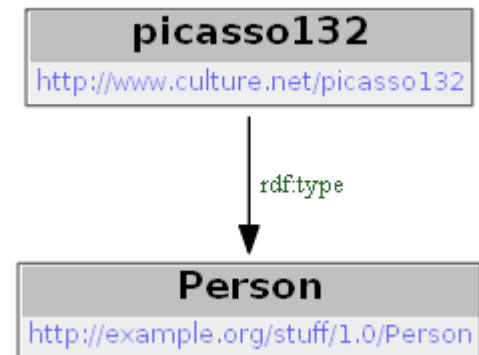
@prefix dc: <http://purl.org/dc/elements/1.1/> .

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

a est un raccourci pour `rdf:type` :

@prefix ex: <http://example.org/stuff/1.

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

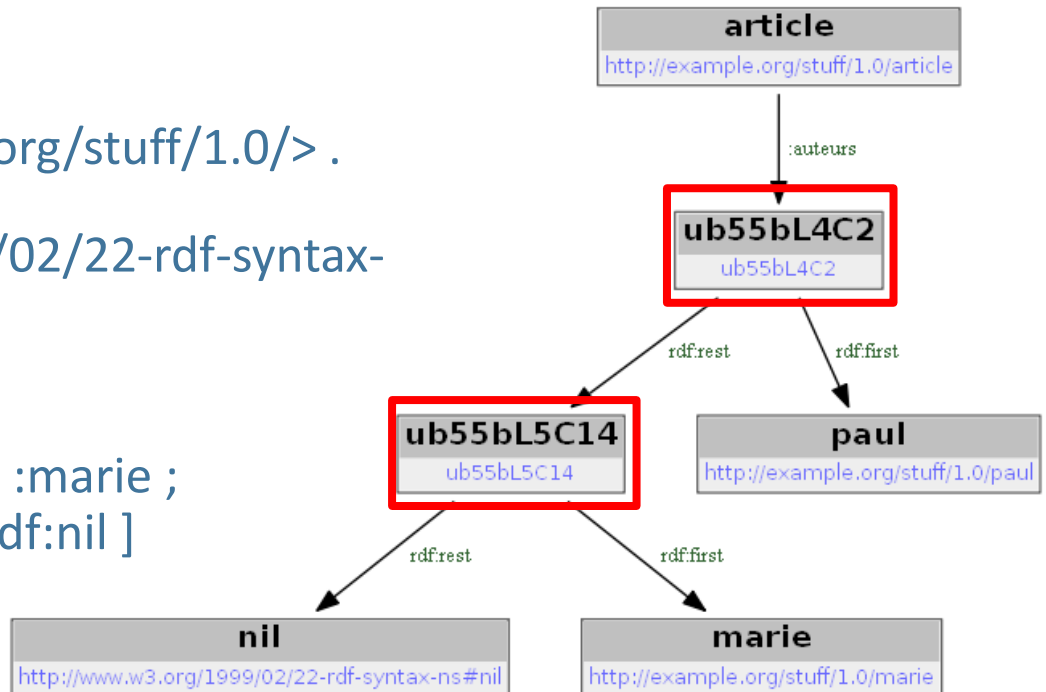


Listes

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

est un raccourci pour

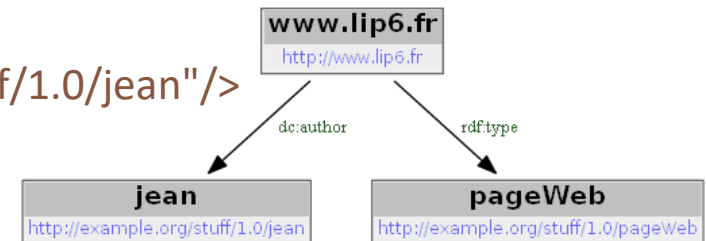
@prefix : <http://example.org/stuff/1.0/> .
@prefix rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-
ns#> .
:article :auteurs
[rdf:first :paul ;
rdf:rest [rdf:first :marie ;
rdf:rest rdf:nil]
] .



Exemple : XML et Turtle

Définition des propriétés d'une ressource existante

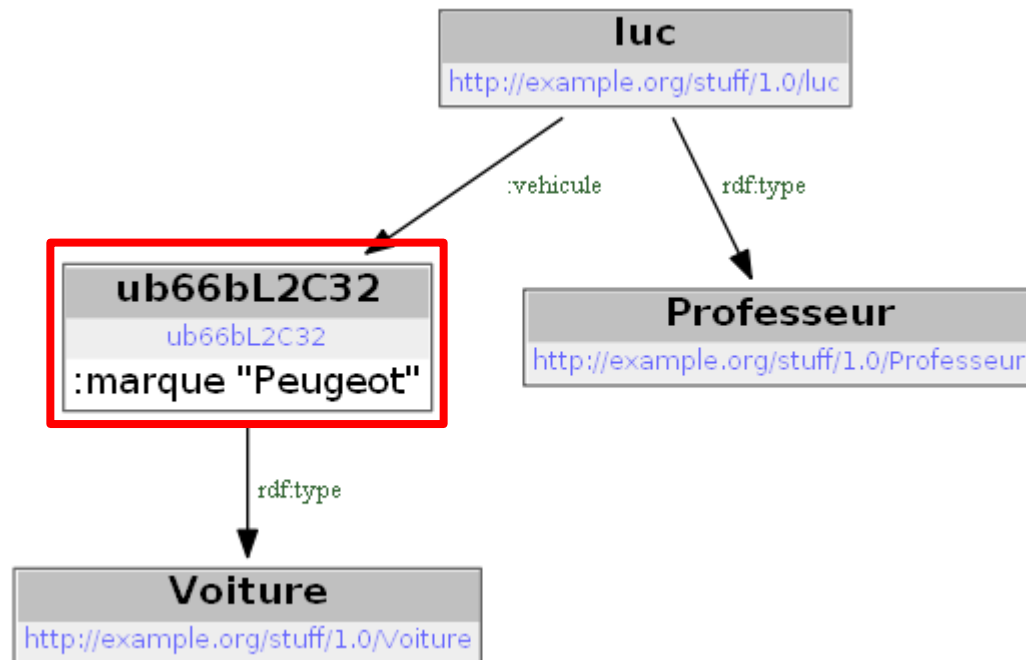
```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns="http://example.org/stuff/1.0/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <pageWeb rdf:about="http://www.lip6.fr">
    <dc:author rdf:resource="http://example.org/stuff/1.0/jean"/>
  </pageWeb>
</rdf:RDF>
```



Turtle :

```
@prefix : <http://example.org/stuff/1.0/> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
<http://www.lip6.fr> a :pageWeb ; dc:author :jean .
```

Exemple : nœuds blanc



Description de nœuds blancs

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

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<rdf:RDF
```

```
  xmlns="http://example.org/stuff/1.0/"
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>
```

```
<rdf:Description rdf:about="http://example.org/stuff/1.0/luc">
```

```
  <vehicule>
```

```
    <rdf:Description>
```

```
      <marque>Peugeot</marque>
```

```
      <rdf:type rdf:resource="http://example.org/stuff/1.0/Voiture"/>
```

```
    </rdf:Description>
```

```
  </vehicule>
```

```
  <rdf:type rdf:resource="http://example.org/stuff/1.0/Professeur"/>
```

```
</rdf:Description>
```

```
</rdf:RDF>
```

Turtle :

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

```
:luc a :Professeur ;
```

```
:vehicule [ a :Voiture ;
```

```
:marque "Peugeot" ] .
```

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

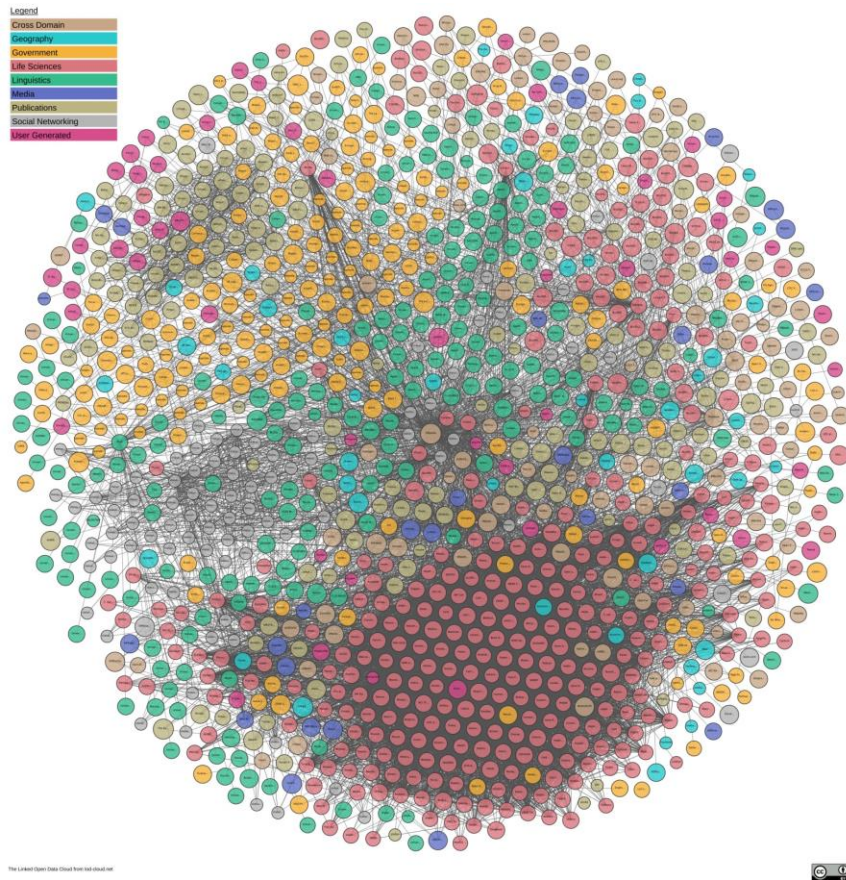
Linked Data (Web des données) : mise en relation des données pour constituer un réseau global (initiative W3C)

- 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

Linked Open Data cloud



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`.

Interroger les schémas RDF : **SPARQL**