Web Sémantique

RDF syntaxes suite
RDFa
RDFS / OWL
Jena

antoine@naturalpad.fr
Github : natoine

RDF

XML / RDF → Une syntaxe mais pas la seule

D'autres syntaxes :

- N-triples
- Turtle

XML RDF

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#">
  <rdf:Description rdf:about="http://somewhere/JohnSmith">
    <vcard:N rdf:parseType="Resource">
      <vcard:Family>Smith</vcard:Family>
      <vcard:Given>John</vcard:Given>
    </vcard:N>
    <vcard:FN>John Smith</vcard:FN>
  </rdf:Description>
</rdf:RDF>
```

FN :Formatted Name

N-triples

<http://somewhere/JohnSmith> <http://www.w3.org/2001/vcard-rdf/3.0#N> _:A4019455bX3aX1419ed7ea70X3aXX2dX7fff .

<http://somewhere/JohnSmith> <http://www.w3.org/2001/vcard-rdf/3.0#FN> "John Smith" .

_:A4019455bX3aX1419ed7ea70X3aXX2dX7fff <http://www.w3.org/2001/vcard-rdf/3.0#Family> "Smith" .

_:A4019455bX3aX1419ed7ea70X3aXX2dX7fff <http://www.w3.org/2001/vcard-rdf/3.0#Given> "John" .

Turtle

```
<http://somewhere/JohnSmith>
      <http://www.w3.org/2001/vcard-rdf/3.0#FN>
            "John Smith" ;
      <http://www.w3.org/2001/vcard-rdf/3.0#N>
            [ <http://www.w3.org/2001/vcard-rdf/3.0#Family>
                  "Smith" ;
              <http://www.w3.org/2001/vcard-rdf/3.0#Given>
                  "John"
            ] .
```

Blanknode

<http://somewhere/JohnSmith> <http://www.w3.org/2001/vcard-rdf/3.0#N> _:A4019455bX3aX1419ed7ea70X3aXX2dX7fff .

Ou ressource anonyme, ressource inconnue, non identifiée par une URI.

A éviter autant que possible... mais bon dès fois ça peut servir.

Par exemple pour déclarer une liste ou une collection.

RDFa

Injecter du RDF dans une page Web

Utiliser les attributs du HTML

Compatible XHTML et HTML 5

RDFa

Attributs XHTML réutilisés :

- ***rel***, liste de propriétés séparées par des espaces, utilisé pour exprimer les relations entre deux ressources
- ***rev***, liste de propriétés inversées séparées par des espaces, utilisé pour exprimer les relations entre deux ressources en inversant le sujet et l'objet
- ***content***, valeur littérale d'une propriété
- ***href*** et ***src***, indiquant l'URI d'une ressource utilisée comme objet d'une propriété

RDFa

Attributs ajoutés par RDFa :

- ***about***, indique le sujet d'une propriété
- ***property***, liste de propriétés ayant une valeur littérale
- ***resource***, indique l'URI d'une ressource utilisée comme objet d'une propriété
- ***datatype***, permet de spécifier le type de données littérales
- ***typeof***, liste de types pour une ressource séparés par des espaces

RDFa - HTML

```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:cal="http://www.w3.org/2002/12/cal/icaltzd#"
xmlns:xs="http://www.w3.org/2001/XMLSchema#" >
<head> <base href="http://www2012.org" /> </head>
<body>
<p about="#event1" typeof="cal:Vevent">
 <b property="cal:summary">Conférence WWW 2012</b>: du
 <span property="cal:dtstart" datatype="xs:date">2012-04-16</span>
 au <span property="cal:dtend" datatype="xs:date">2012-04-20</span>.
  Pour plus d'info voir <a rel="cal:url" href="http://www2012.org/">
  le site de la conférence WWW2012</a> à
 <span property="cal:location">Lyon, France</span>.
 </p>
 </body>
 </html>
```

RDFa - RDF

```
<http://www2012.org#conf>
        <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
                <http://www.w3.org/2002/12/cal/icaltzd#Vevent> ;
        <http://www.w3.org/2002/12/cal/icaltzd#summary>
                "Conférence WWW 2012" ;
        <http://www.w3.org/2002/12/cal/icaltzd#dtstart>
                "2012-04-16"^^<http://www.w3.org/2001/XMLSchema#date> ;
        <http://www.w3.org/2002/12/cal/icaltzd#dtend>
                "2012-04-20"^^<http://www.w3.org/2001/XMLSchema#date> ;
        <http://www.w3.org/2002/12/cal/icaltzd#url>
                <http://www2012.org/> ;
        <http://www.w3.org/2002/12/cal/icaltzd#location>
                "Lyon, France" .
```

Modèle RDF

- Définit le concept de Ressources.
- Définit le concept de Litéral
- Définit le concept de Propriété (comme étant notamment un sous concept de Ressources)

Modèle RDF

- Définit le concept de triplet, ou sentence (statement). Un triplet est de la forme : {prédicat, sujet, objet}
- Le prédicat est une propriété, le sujet est une ressource, l'objet est soit une ressource soit un litéral

Modèle RDF

- Il existe une propriété RDF:type
- Les triplets de la forme {RDF:type, sujet, objet} doivent satisfaire le fait que sujet et objet sont des ressources.
- Il existe trois propriétés RDF:predicate, RDF :subject, RDF:object
- Il existe un ensemble RDF:Statement de Ressource qui n'est pas un ensemble de Propriétés

Modèle RDF

- La réification d'un triplet permet de décrire explicitement la structure des triplets et de les décrire comme des ressources.
- Exemple, la réification du triplet {predicat : pred, sujet : suj, objet : ob} est une Ressource r telle que l'on peut écrire les triplets :

  s1: {RDF:predicat, r, pred}

  s2: {RDF:sujet, r, suj}

  s3: {RDF:objet, r, obj}

  s4: {RDF:type, r, [RDF:Statement]}

  Permet d'identifier un triplet. On va pouvoir dire des choses au sujet de ce triplet puisque c'est une ressource. Comme par exemple « qui a écrit ce triplet ».

Modèle RDF

- Il existe 3 ressources qui ne sont pas des propriétés : RDF:Seq (liste ordonnée), RDF:Bag (liste non ordonnée), and RDF:Alt (liste de valeurs alternatives).
- Il existe un ensemble de propriétés qui correspondent à des ordinaux (1, 2, 3 ...) appelés Ord (permettent de décrire un ordre). On les utilise en écrivant : RDF:_1, RDF:_2, RDF:_3, ...

RDFS

RDF Schema

1998 working draft

2004 recommandation

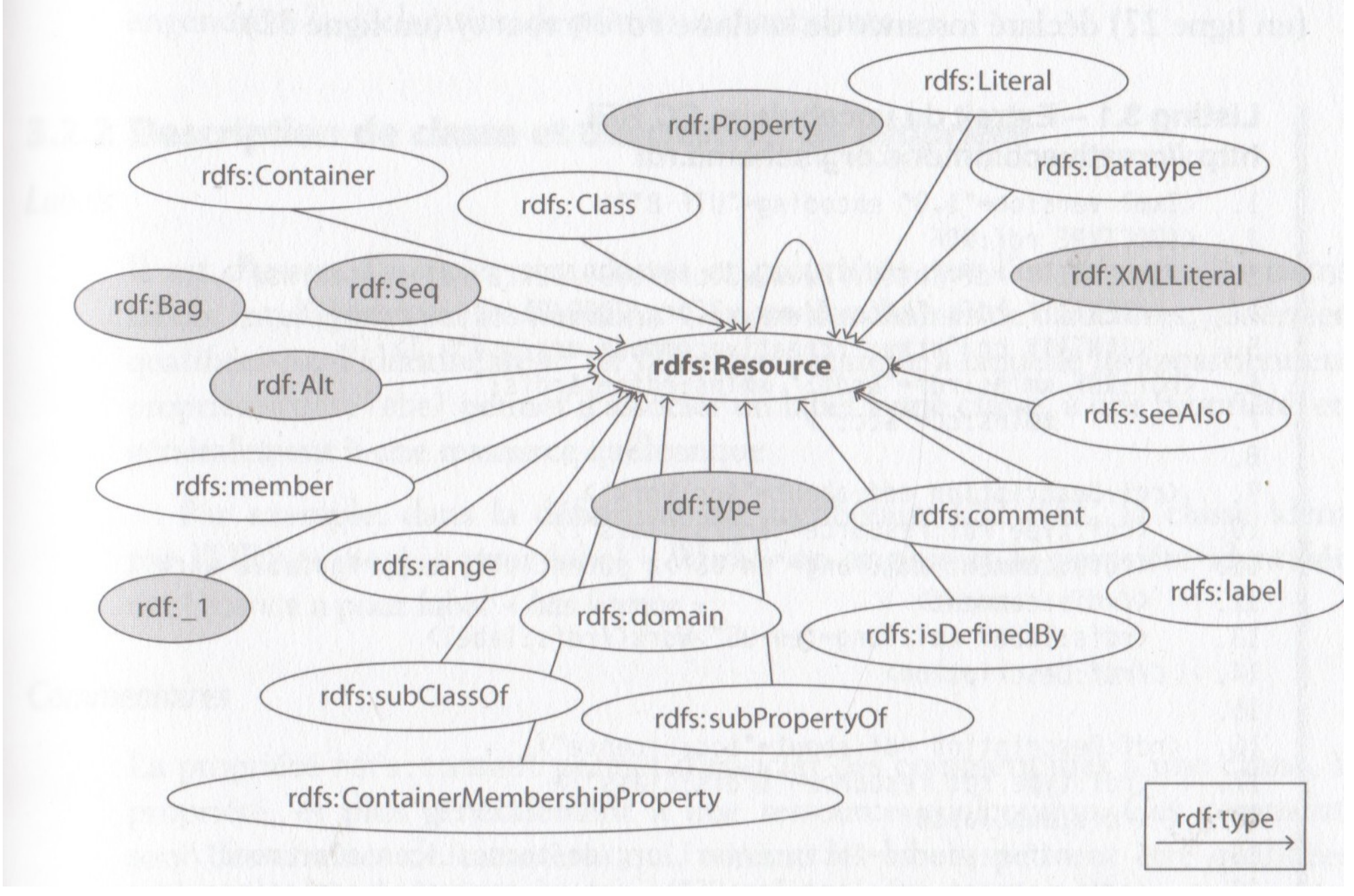
Nommer des classes et des propriétés

Signer les propriétés (domain, range)

Définir une organisation hierarchique entre les classes, entre les propriétés

Premier langage de description de vocabulaire créé

RDFS


rdfs:Literal
rdf:Property
rdfs:Container
rdfs:Class
rdfs:Datatype
rdf:XMLLiteral
rdf:Seq
rdf:Bag
rdf:Alt
rdfs:Resource
rdfs:seeAlso
rdfs:member
rdf:type
rdfs:comment
rdfs:range
rdfs:label
rdf:_1
rdfs:domain
rdfs:isDefinedBy
rdfs:subClassOf
rdfs:subPropertyOf
rdfs:ContainerMembershipProperty
rdf:type

Ecrire un vocabulaire avec RDFS

Définition d'une classe :

```
<rdf:Description rdf:about=ns :[NomDeLaClasse]>
        <rdf :type rdf:resource="rdfs:Class"/>
*       <rdfs:subClassOf rdf:resource="http://..../NomSuperClasse" />
        <rdfs:comment>[un commentaire au sujet de la classe]</rdfs:comment>
        <rdfs:label>[NomDeLaClasse]</rdfs:label>
</rdf>
```

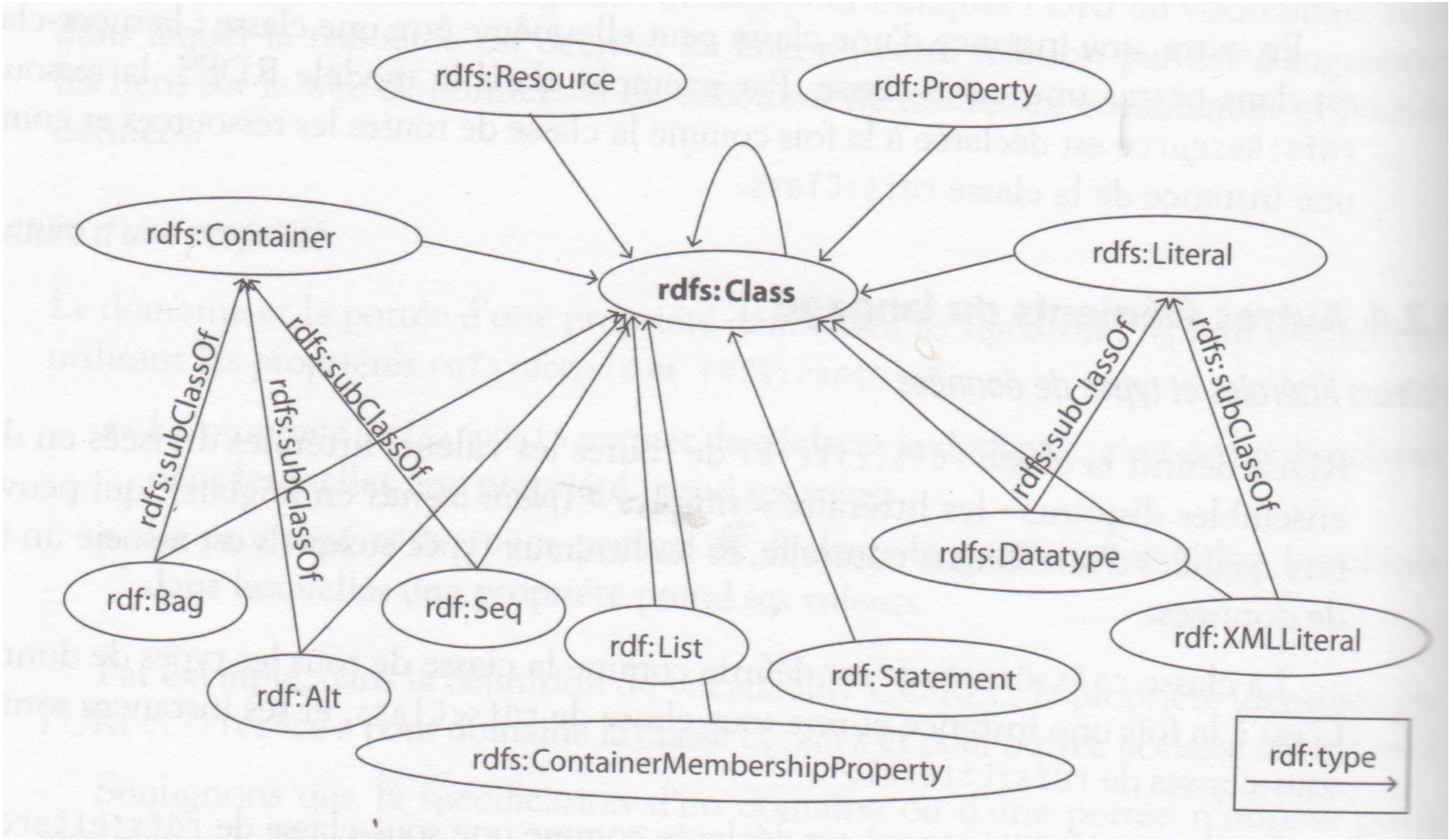
Définition d'une propriété :

```
<rdf:Description rdf:about=ns:[nomDeLaPropriété]>
        <rdf:type rdf:resource="rdf:Property" />
*       <rdfs:subPropertyOf rdf:resource="http://..../nomSuperPropriété />
        <rdfs:range rdf:resource=[ClasseObjet] />
        <rdfs:domain rdf:resource=[ClasseSujet] />
        <rdfs:label>[nomDeLaPropriété]</rdfs:label>
</rdf>
```

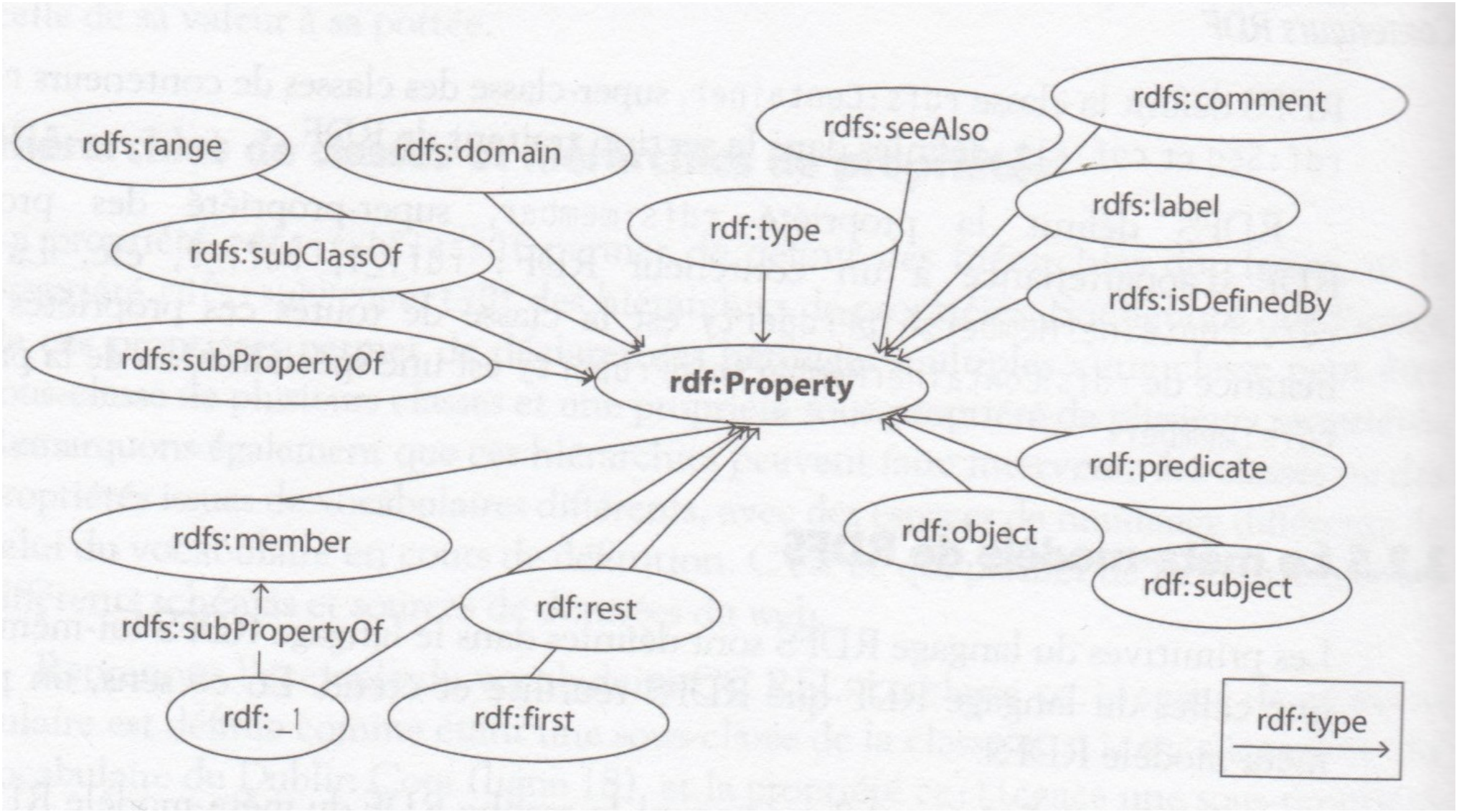
Méta-modèle RDFS

- Les primitives de RDFS sont définies dans RDFS.
- RDFS définies également les primitives de RDF qu'il réutilise.

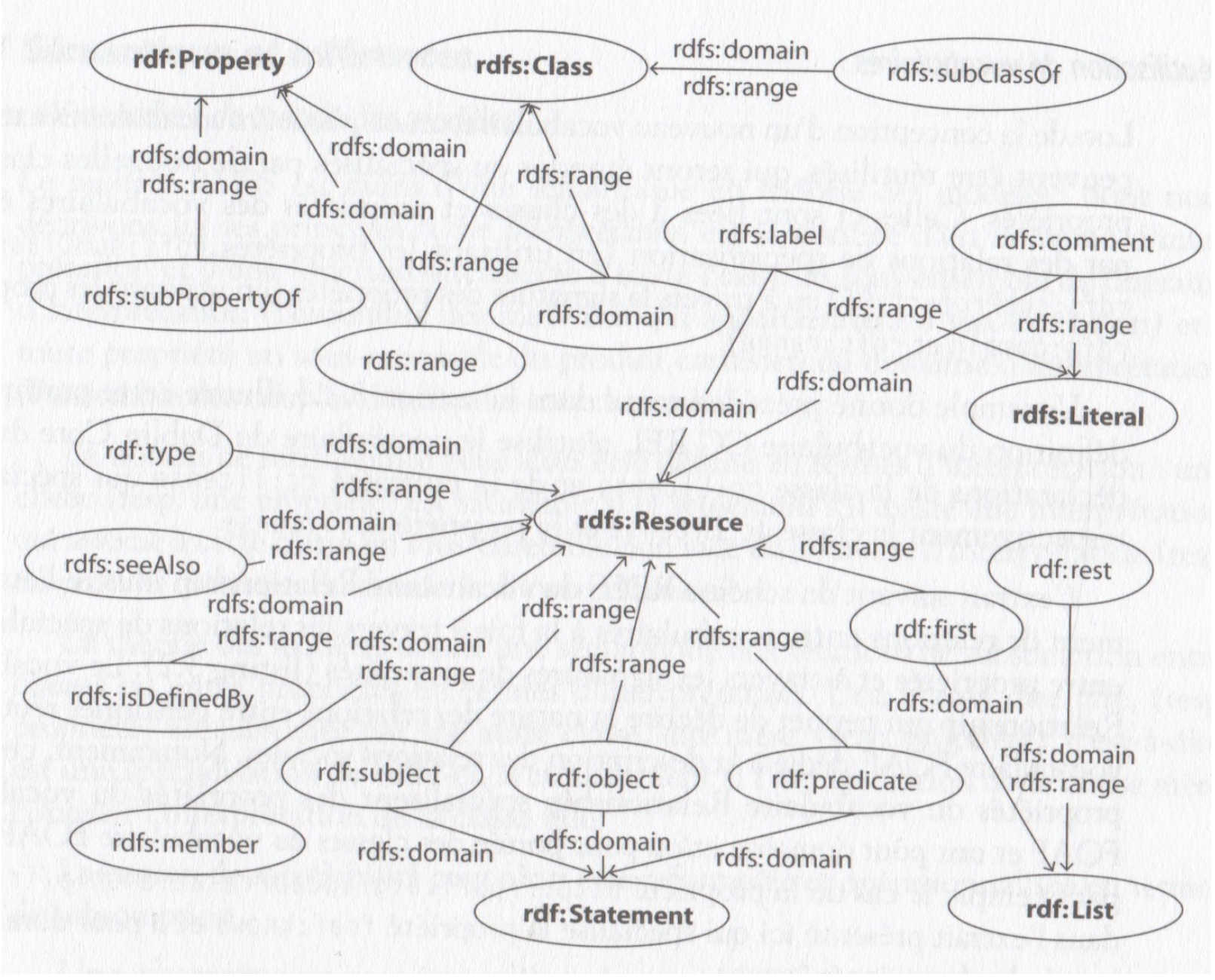
Méta-modèle RDFS



Méta-modèle RDFS



Méta-modèle RDFS



Lier des données

Réutiliser un vocabulaire existant permet de simplifier la mise en lien de données.

Quand on définit un nouveau vocabulaire on essaye autant que possible d'étendre des vocabulaires existants en définissant des sous classes, des sous propriétés et en utilisant des classes existantes pour les *domain* et *range*

On utilise également la propriété *rdfs:seeAlso* qui permet de signaler ou trouver d'autres informations au sujet d'une ressource.

OWL

- Avec RDFS on écrit des ontologies dites légères : des hiérarchies de classes et de propriétés.

  Recommandation depuis 2004
- OWL permet d'exprimer :
  - Équivalence de classes
  - Équivalence de propriétés
  - Égalité de ressources
  - Différence
  - Contraintes
  - Symétrie
  - Cardinalité
  - ...

OWL – déclarer une classe

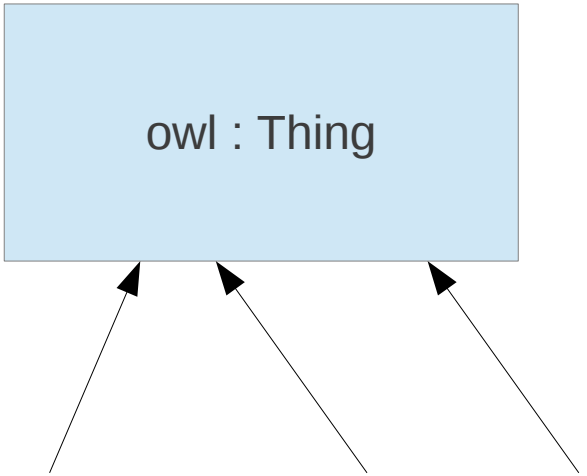
- 6 façons de faire :
  - Nommage d'une classe
  - Énumération des instances
  - Intersection de descriptions de classe
  - Union de descriptions de classe
  - Complémentaire d'une description de classe
  - Restriction de propriétés pour la classe décrite

Exemples : http://www.w3.org/TR/owl-guide/wine.rdf

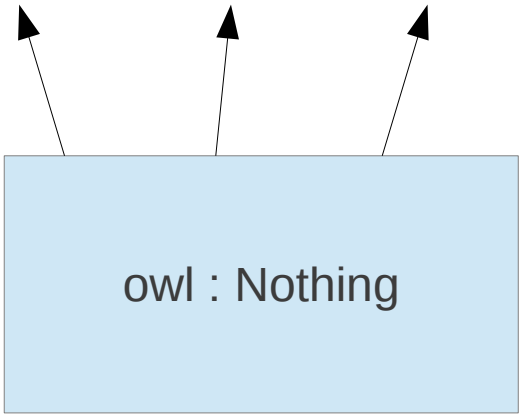
La classe vendange tardive

```
<owl:Class rdf:ID="LateHarvest">
    <rdfs:subClassOf rdf:resource="#Wine" />
    <owl:disjointWith rdf:resource="#EarlyHarvest" />
</owl>
```

Le treillis OWL



Toutes les classes sont des sous-classes de Thing.
Nothing est une sous classe de toutes les classes.



Énumération des instances

```
<owl:Class rdf:ID="WineBody">
    <rdfs:subClassOf rdf:resource="#WineTaste" />
    <owl:oneOf rdf:parseType="Collection">
      <owl:Thing rdf:about="#Light" />
      <owl:Thing rdf:about="#Medium" />
      <owl:Thing rdf:about="#Full" />
    </owl:oneOf>
  </owl:Class>
```

Intersection de classes

```
<owl:Class rdf:ID="WhiteLoire">
    <owl:intersectionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Loire" />
      <owl:Class rdf:about="#WhiteWine" />
    </owl:intersectionOf>
  </owl:Class>
```

Conjonction (et) logique

Union de classes

```
<owl:Class rdf:ID="WineDescriptor">
    <rdfs:comment>Made WineDescriptor unionType of tastes
and color</rdfs:comment>
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#WineTaste" />
      <owl:Class rdf:about="#WineColor" />
    </owl:unionOf>
  </owl:Class>
```

Disjonction (ou) logique

Complémentaire d'une classe

```
<owl:Class rdf:ID="NonConsumableThing">
    <owl:complementOf rdf:resource="#ConsumableThing" />
  </owl:Class>
```

négation logique

Restriction de propriété - hasValue

```
<owl:Class rdf:ID="WhiteWine">
    <owl:intersectionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Wine" />
      <owl:Restriction>
        <owl:onProperty rdf:resource="#hasColor" />
        <owl:hasValue rdf:resource="#White" />
      </owl:Restriction>
    </owl:intersectionOf>
  </owl:Class>
```

Restriction de propriété - allValuesFrom

```
<owl:Class rdf:about="#WhiteLoire">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#madeFromGrape" />
        <owl:allValuesFrom>
          <owl:Class>
            <owl:oneOf rdf:parseType="Collection">
              <owl:Thing rdf:about="#CheninBlancGrape" />
              <owl:Thing rdf:about="#PinotBlancGrape" />
              <owl:Thing
rdf:about="#SauvignonBlancGrape" />
            </owl:oneOf>
          </owl:Class>
        </owl:allValuesFrom>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
```

Doit avoir toutes les propriétés

Relations entre propriétés - inverseOf

```
<owl:ObjectProperty rdf:ID="producesWine">
    <owl:inverseOf rdf:resource="#hasMaker" />
  </owl:ObjectProperty>
```

Relations entre propriétés – transitivité - propertyChainAxiom

```
<owl:ObjectProperty rdf:ID="uncle">
    <owl:propertyChainAxiom rdf:parseType="Collection">
      <owl:ObjectProperty rdf:about="#parent">
      <owl:ObjectProperty rdf:about="#brother">
    </owl:propertyChainAxiom>
  </owl:ObjectProperty>
```

Réification de l'ontologie

```
<owl:Ontology rdf:about="">
    <rdfs:comment>An example OWL ontology</rdfs:comment>
    <owl:priorVersion>
      <owl:Ontology
rdf:about="http://www.example.org/wine-020303"/>
    </owl:priorVersion>
    <owl:imports
rdf:resource="http://www.w3.org/2002/03owlt/miscellaneous/
consistent002"/>
    <rdfs:comment>Derived from the DAML Wine ontology at

http://ontolingua.stanford.edu/doc/chimaera/ontologies/win
es.daml
      Substantially changed, in particular the Region
based relations.
    </rdfs:comment>
    <rdfs:label>Wine Ontology</rdfs:label>
  </owl:Ontology>
```

Introduction de la classe Ontology, on peut maintenant dire des choses au sujet de l'ontologie. L'ontologie est une Ressource.

Jena

- jena.apache.org
- Sources écriture / lecture de RDF :

  https://github.com/natoine/jenaTutorials