Ontologies et Web Sémantique

Les Ontologies – Mise en Œuvre avec Protégé

Plan du cours

1. Protégé
2. Exemple: L'ontologie "African Wildlife »
3. Description de classes, propriétés, instances, en Protégé
4. Restrictions sur les propriétés en Protégé
5. Raisonnements sur les ontologies en Protégé

Présentation de Protégé

Protégé – IDE Ontologies OWL

- Protégé est un système permettant la création d'ontologies très populaire développé à l'Université de Stanford.
- Protégé est développé en Java.
- Protégé fonctionnant sur la machine virtuelle Java, est multiplateformes (linux, Windows, MacOS, …)
- Protégé est gratuit et son code source est publié sous une licence libre (Mozilla Public License).
- Protégé peut lire et sauvegarder des ontologies dans la plupart des formats d'ontologies : RDF, RDFS, OWL, Turtle, JSON-LD, etc.
- Lien de téléchargement : https://protege.stanford.edu/
- Web Protégé : https://webprotege.stanford.edu/

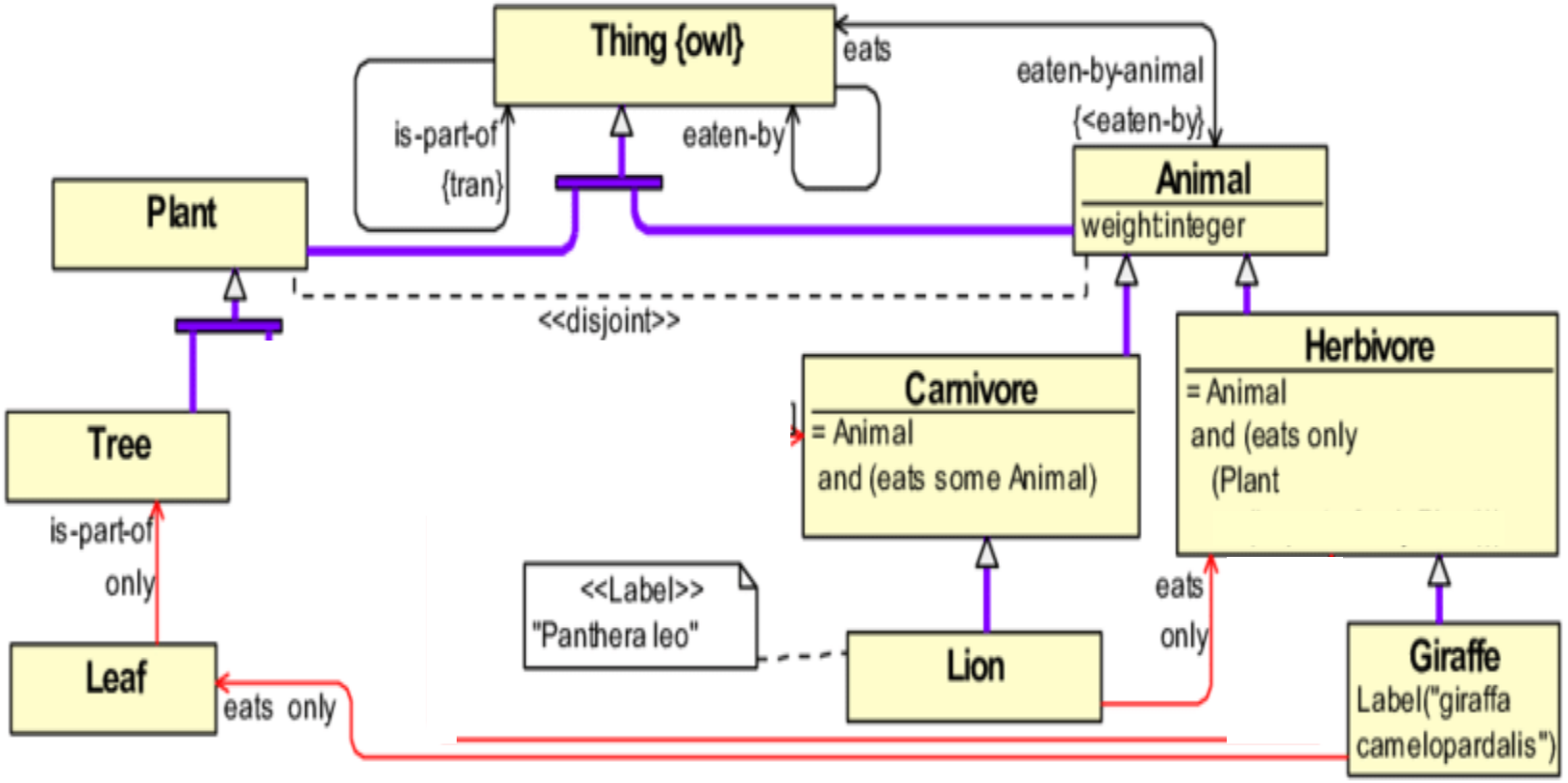
Présentation de Protégé

Protégé – IDE Ontologies OWL

- A télécharger depuis : https://protege.stanford.edu/
- Création classes / sous-classes.
- Création de propriétés.
- Restrictions classes / propriétés.
- Paramétrages .
- Création d'instances de classes.
- Vérifier l'ontologie. Reasoning.
- Visualisation en graphe. Interrogation SPARQL.
- Etc.

Exemple - African Wildlife Ontology

- **Source** : Antoniou, G, van Harmelen, F. A Semantic Web Primer. MIT Press, 2003.
- *Réadaptée*.

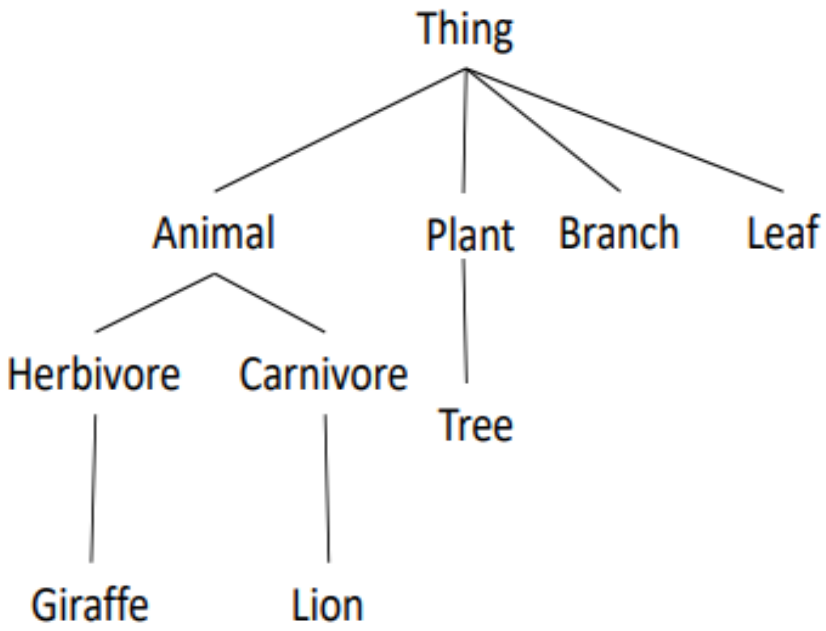


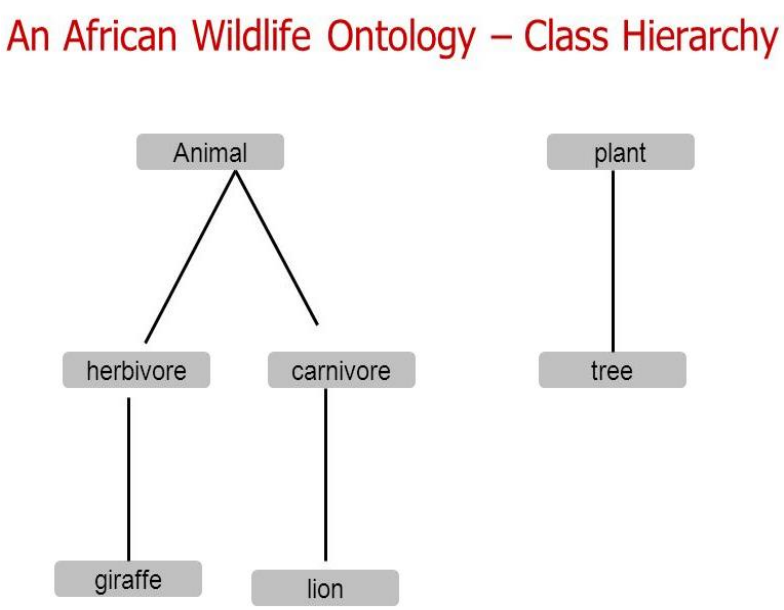
Exemple - African Wildlife Ontology

- **Source** : Antoniou, G, van Harmelen, F. A Semantic Web Primer. MIT Press, 2003. *Réadaptée*.

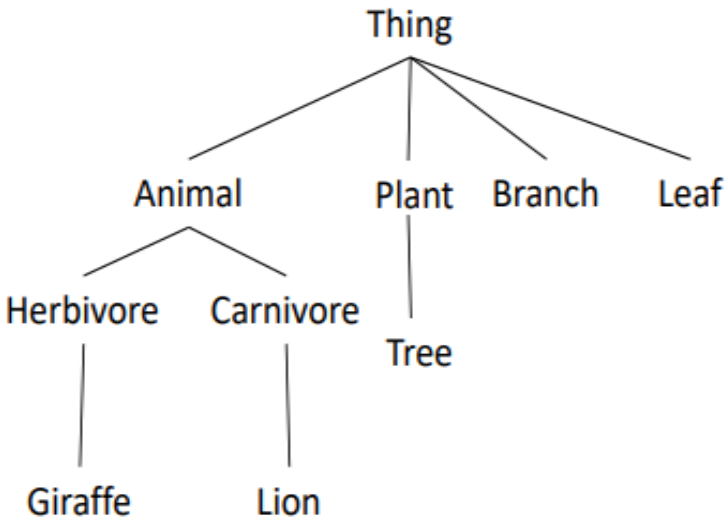
- **Objet** : Cette ontologie décrit la vie animale en Afrique avec :

  - des animaux: carnivores (des lions) et herbivores (des girafes)
  - des plantes telles que des arbres composés de branches et de feuilles





Exemple - African Wildlife Ontology



- **Déclaration de Classes :**

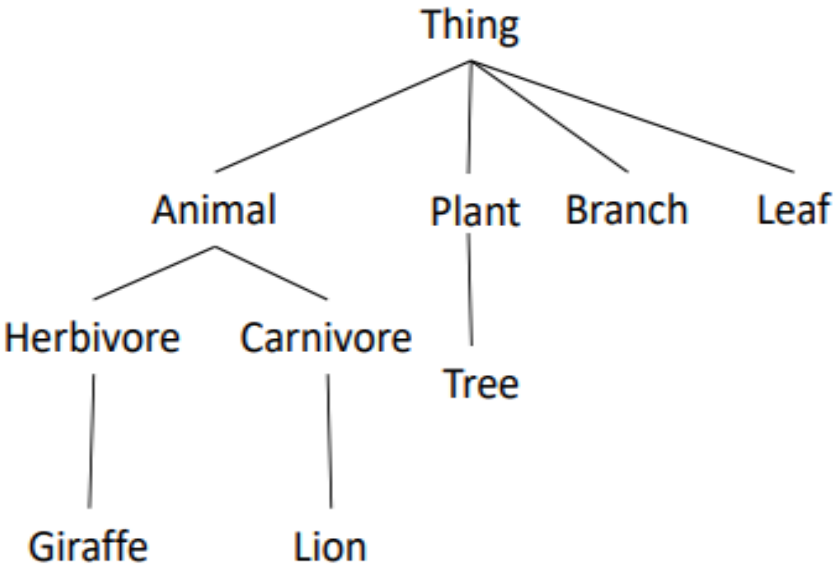
```
<owl:Class rdf:ID="Animal"/>

<owl:Class rdf:ID="Herbivore">
    <rdfs:subClassOf rdf:resource="#Animal"/>
</owl:Class>

<owl:Class rdf:ID="Carnivore">
    <rdfs:subClassOf rdf:resource="#Animal"/>
</owl:Class>

<owl:Class rdf:ID="Lion">
    <rdfs:subClassOf rdf:resource="#Carnivore"/>
</owl:Class>

<owl:Class rdf:ID="Giraffe">
    <rdfs:subClassOf rdf:resource="#Herbivore"/>
</owl:Class>
```

Exemple - African Wildlife Ontology

- **Déclaration de Classes** :

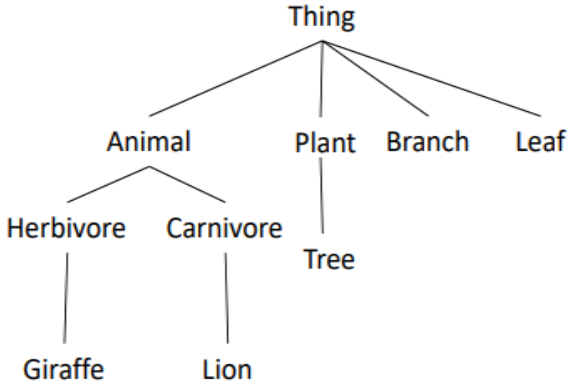
...

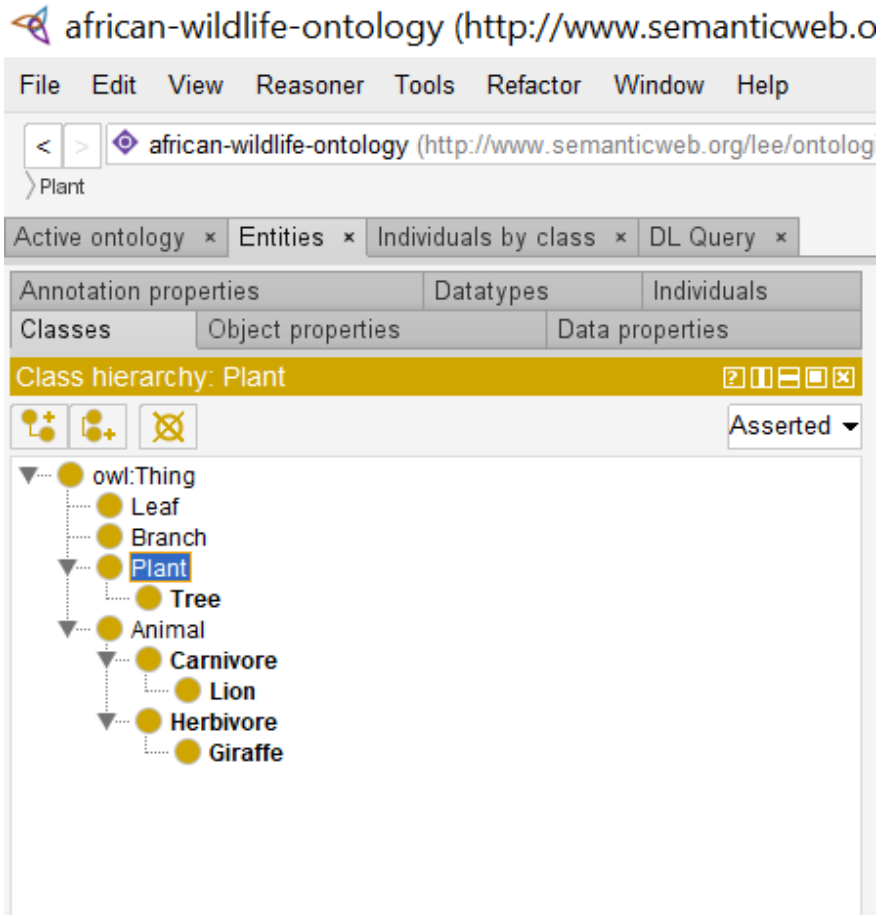
```
<owl:Class rdf:ID="Plant"/>

<owl:Class rdf:ID="Tree">
    <rdfs:subClassOf rdf:resource="#Plant"/>
</owl:Class>

<owl:Class rdf:ID="Branch"/>

<owl:Class rdf:ID="Leaf"/>
```

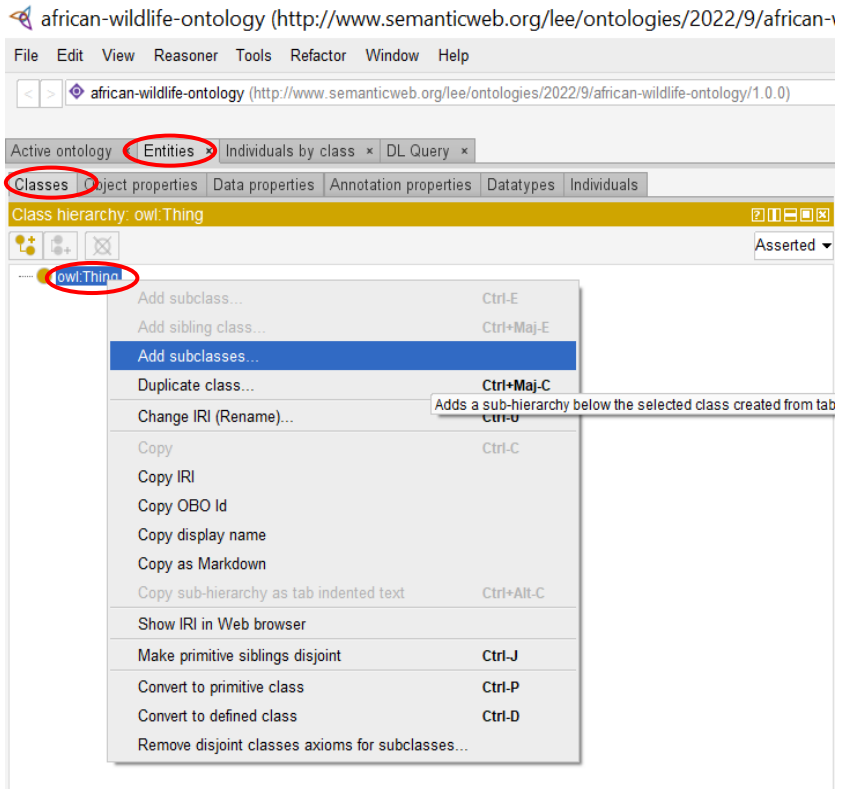


Exemple - African Wildlife Ontology

- **Déclaration de Classes** : dans Protégé







Exemple - African Wildlife Ontology

- **Déclaration de Propriétés** : ObjectProperty

```
<owl:ObjetcProperty rdf:ID="is-part-of" >
   <rdf:type rdf:resource="&owl;#TransitiveProperty"/>
   <rdf:type rdf:resource="&owl;#ReflexiveProperty"/>
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="eats">
    <rdfs:domain rdf:resource="#Animal"/>
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="eaten-by">
    <owl:inverseOf rdf:resource="#eats"/>
    <rdf:type rdf:resource="&owl;#ReflexiveProperty"/>
</owl:ObjectProperty>
```

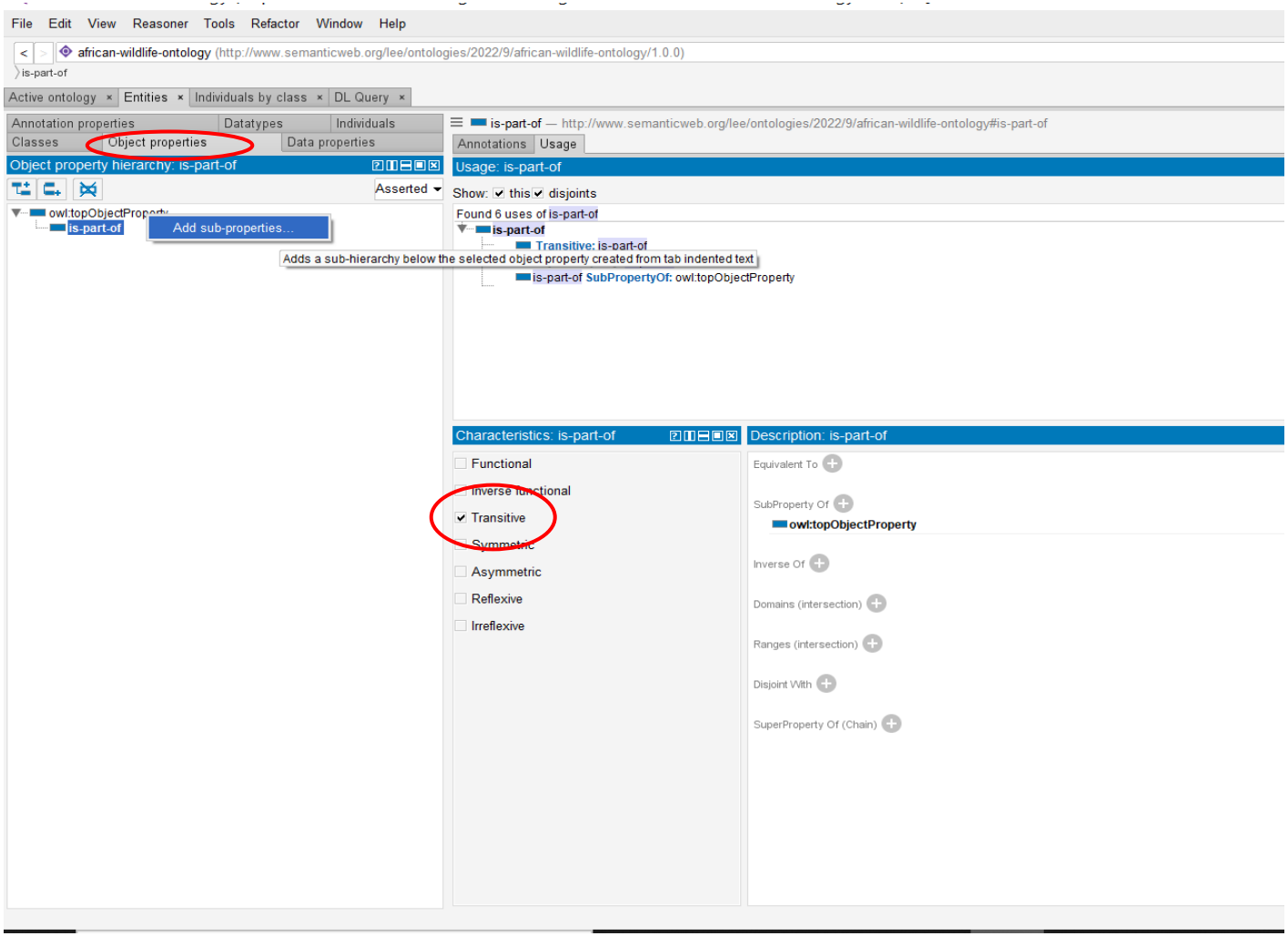
Exemple - African Wildlife Ontology

- **Déclaration de Propriétés** : DatatypeProperty

```
<owl:DatatypeProperty rdf:about="weight">

    <rdfs:domain rdf:resource="#Animal"/>

    <rdfs:range rdf:resource=
   "http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>

</owl:DatatypeProperty>
```

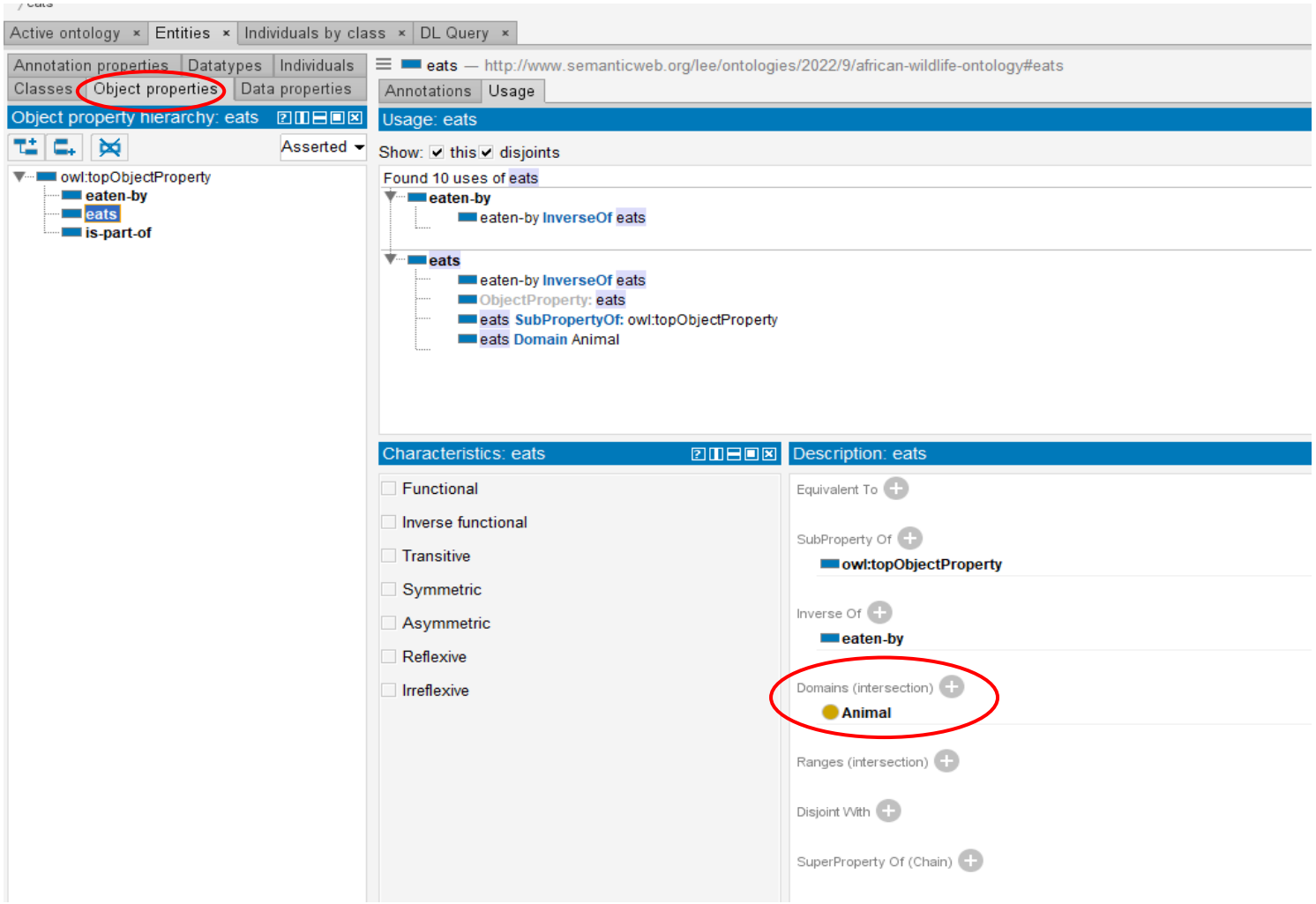
Exemple - African Wildlife Ontology

- **Déclaration de Propriétés** : dans Protégé



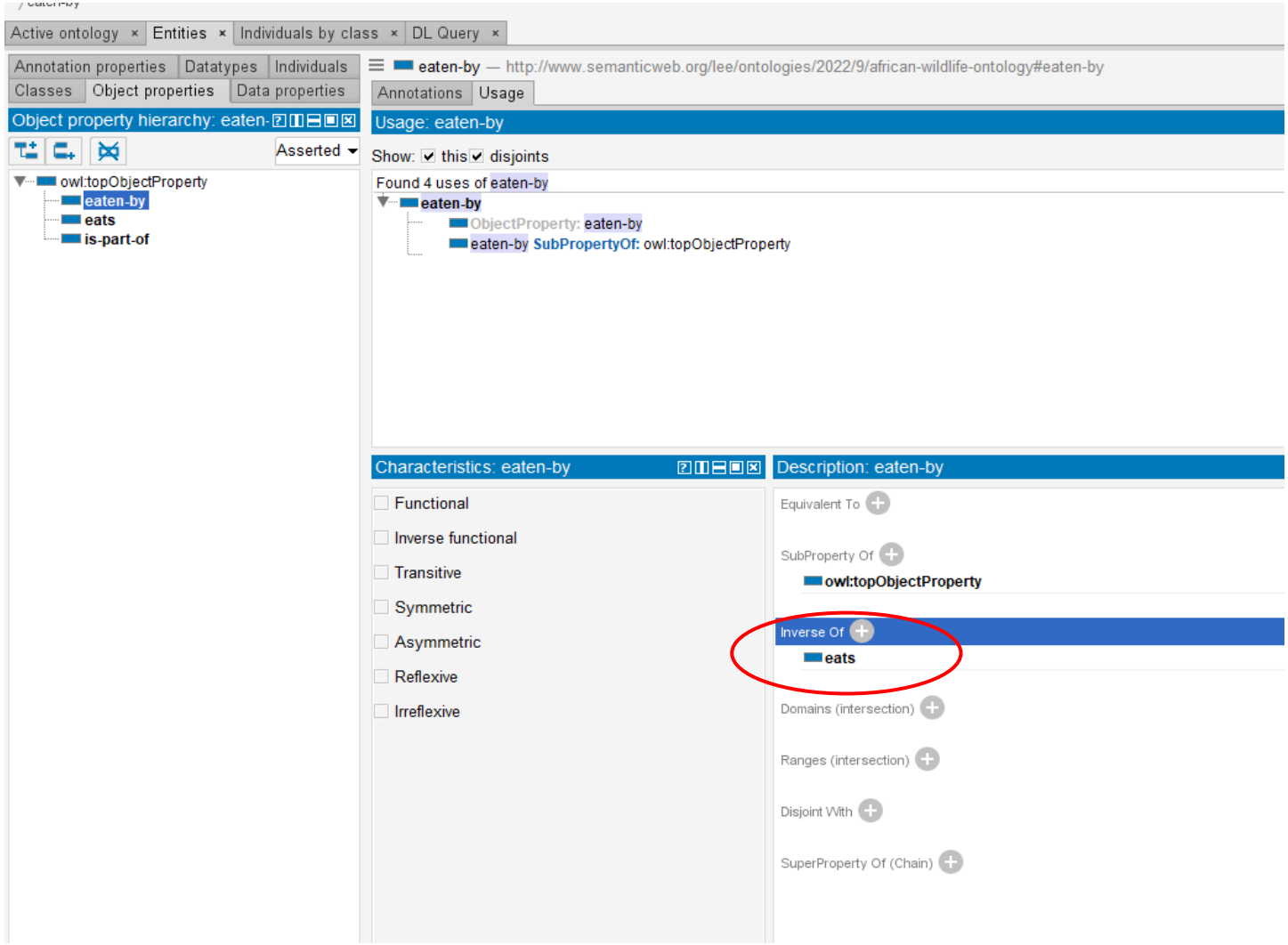
Exemple - African Wildlife Ontology

- **Déclaration de Propriétés** : dans Protégé



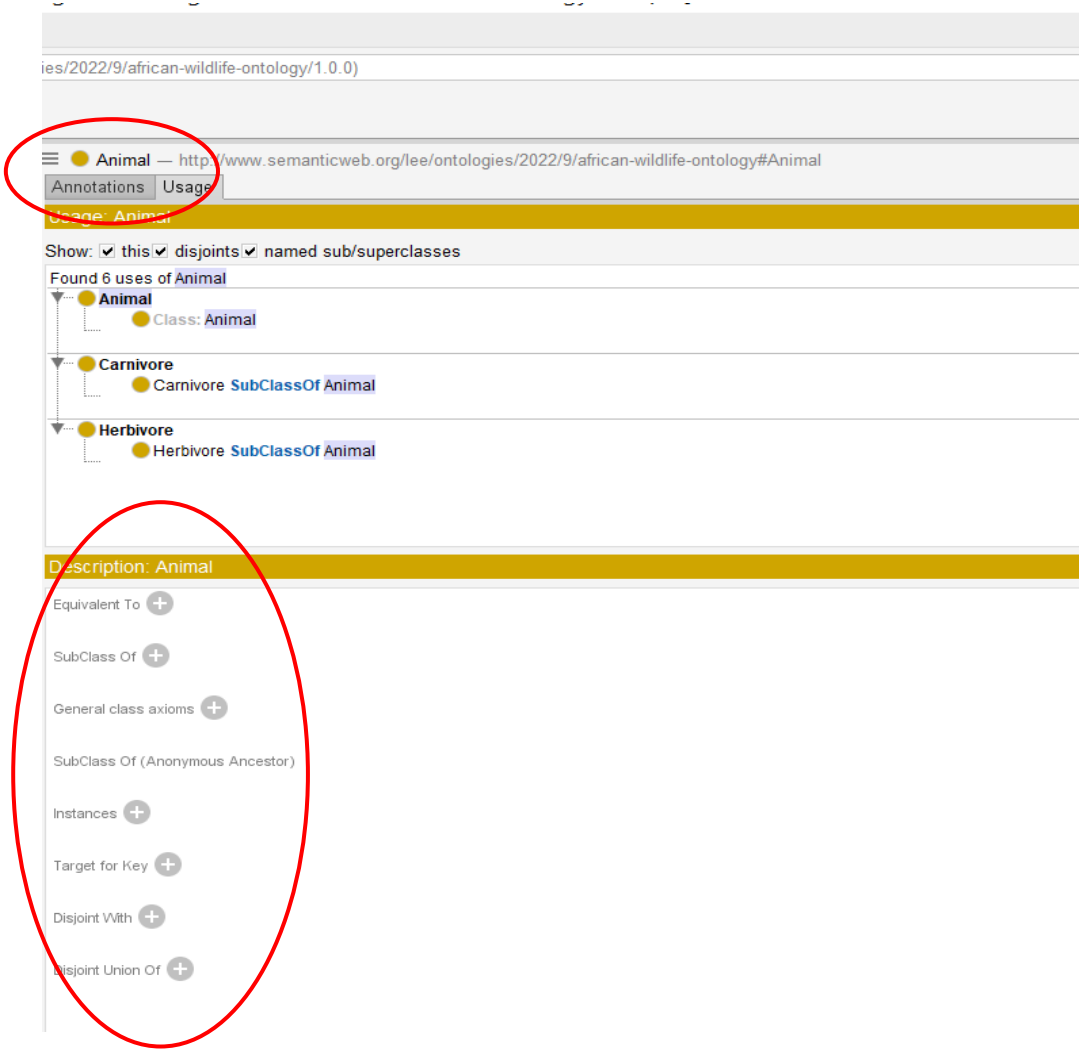
Exemple - African Wildlife Ontology

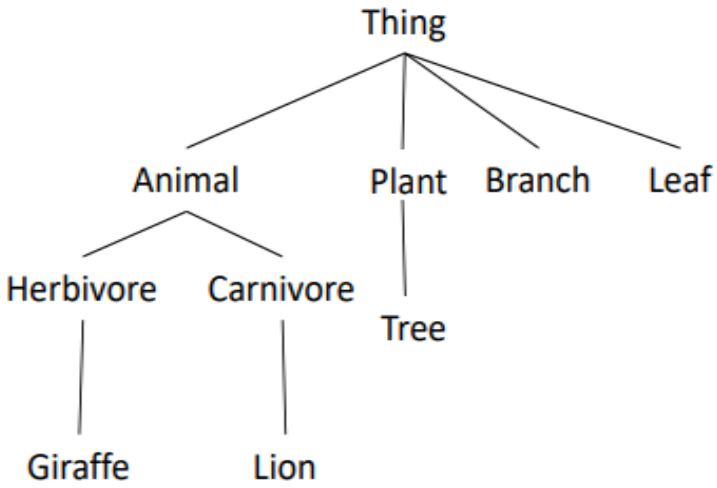
- **Déclaration de Propriétés** : dans Protégé



Exemple - African Wildlife Ontology

- **Déclaration de Classes :**

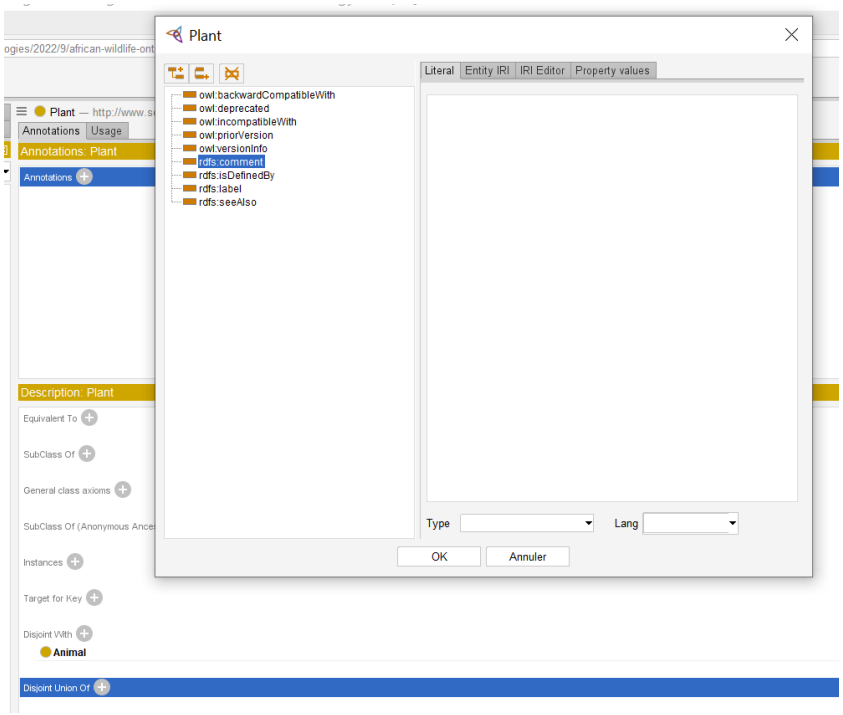


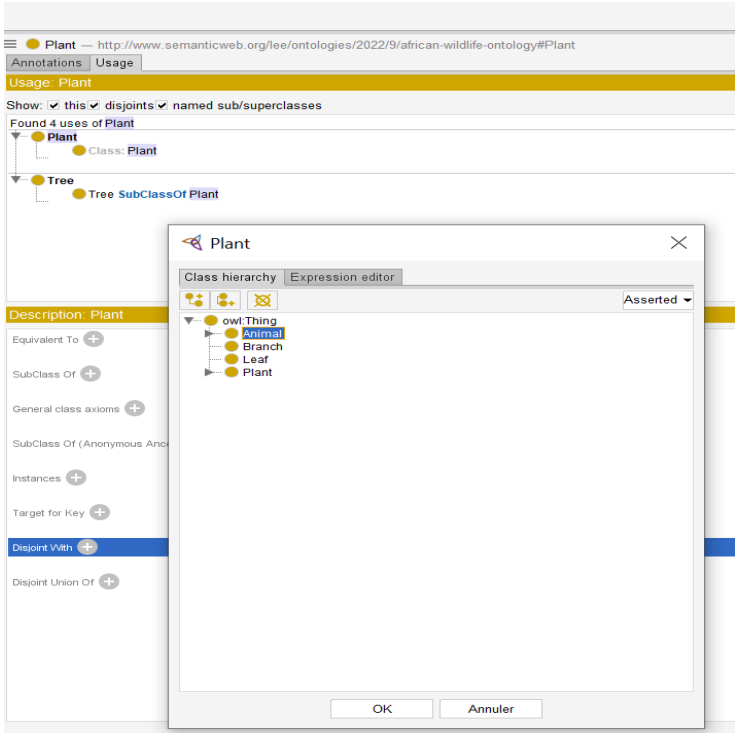


Exemple - African Wildlife Ontology

- **Déclaration de Classes** : sur Protégé

```
<owl:Class rdf:ID="Plant">
    <rdfs:comment>Plants are disjoints from animals
    </rdfs:comment>
    <owl:disjointWith rdf:resource="#Animal"/>
</owl:Class>
```

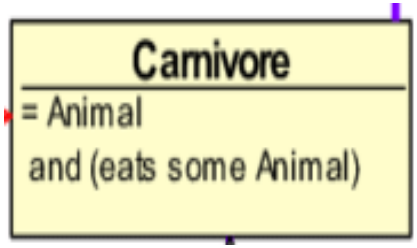




Exemple - African Wildlife Ontology

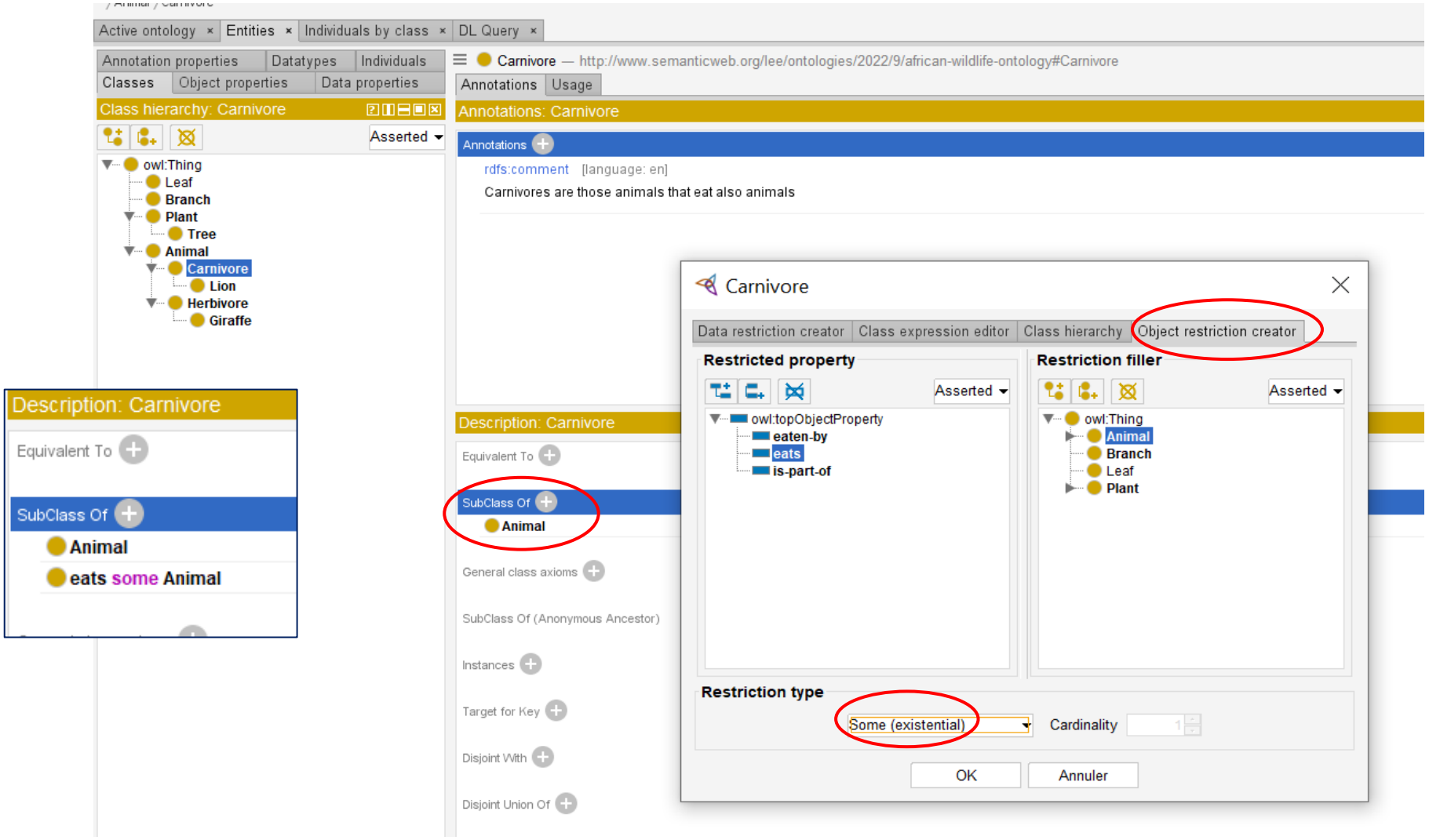
- **Déclaration de Classes** : restrictions sur les propriétés

```
<owl:Class rdf:ID="Carnivore">

  <rdfs:subClassOf rdf:resource="#Animal"/>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#eats"/>
      <owl:someValuesFrom rdf:resource="#Animal"/>
    </owl:Restriction>
  </rdfs:subClassOf>

  <rdfs:comment>
       Carnivores are those animals that eat also animals.
  </rdfs:comment>

</owl:Class>
```



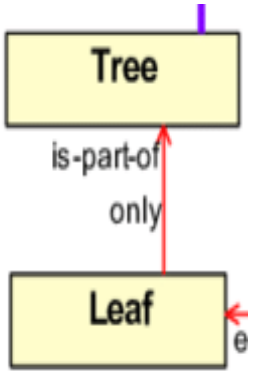
Exemple - African Wildlife Ontology

- **Déclaration de Classes** : restrictions sur les propriétés – dans Protégé



Exemple - African Wildlife Ontology

- **Déclaration de Classes** : restrictions sur les propriétés




```
<owl:Class rdf:ID="Leaf">

    <rdfs:comment>
        Leafs are parts of trees.
    </rdfs:comment>

    <rdfs:subClassOf>
      <owl:Restriction>
         <owl:onProperty rdf:resource="#is-part-of"/>
         <owl:allValuesFrom rdf:resource="#Tree"/>
      </owl:Restriction>
     </rdfs:subClassOf>

</owl:Class>
```

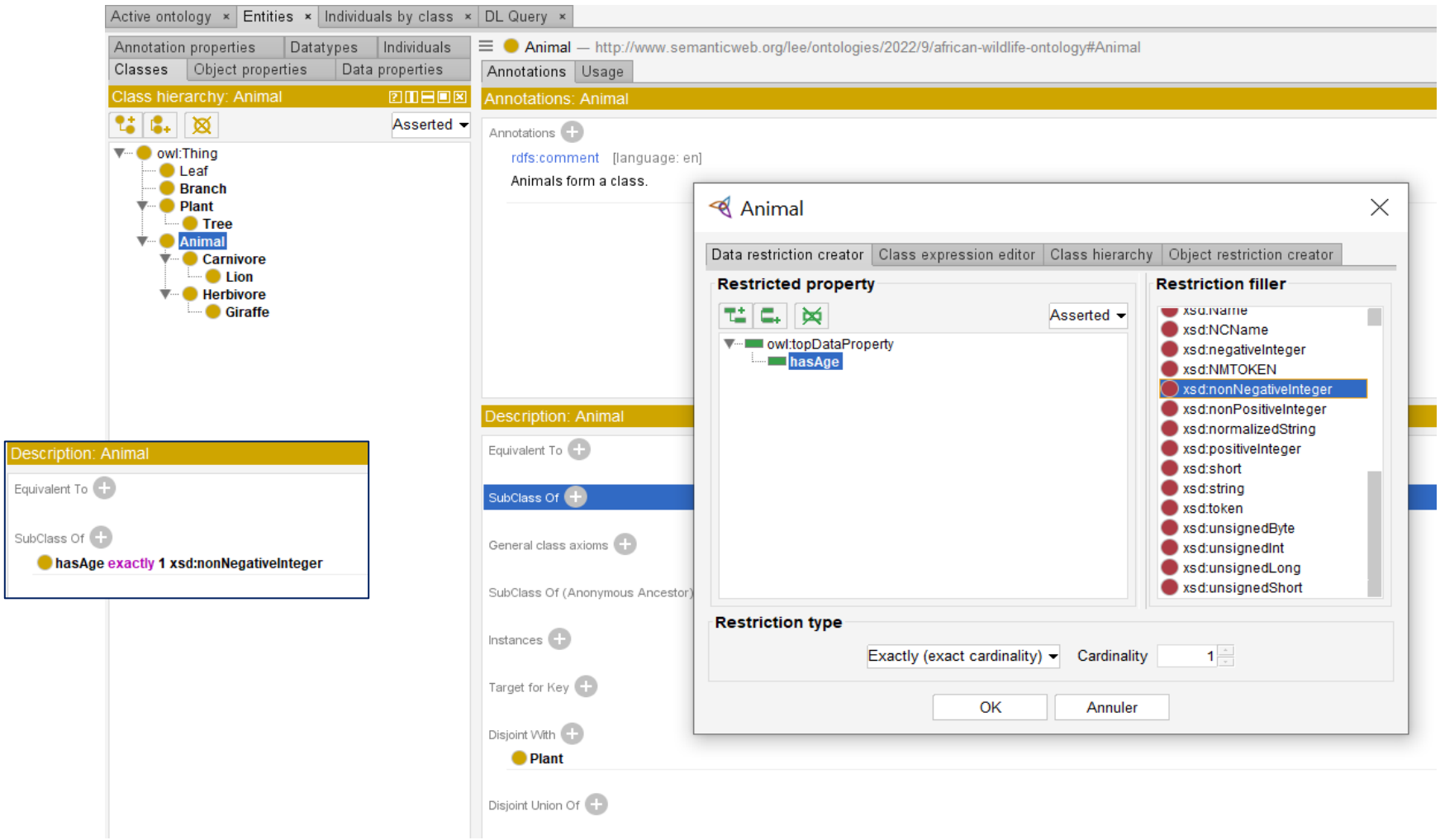
Exemple - African Wildlife Ontology

- **Déclaration de Classes** : restrictions sur les propriétés

```
<owl:Class rdf:about="#Animal">

      <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="hasAge"/>
            <owl:cardinality
                        rdf:datatype=&xsd;nonNegativeInteger>
                    1
            </owl:cardinality>
        </owl:Restriction>
     </rdfs:subClassOf>

</owl:Class>
```

Exemple - African Wildlife Ontology

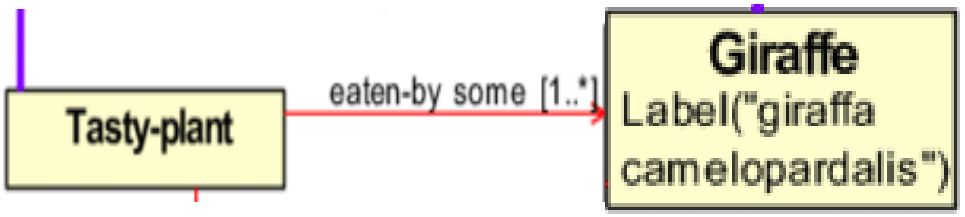
- **Déclaration de Classes** : restrictions sur les propriétés – dans Protégé



Exemple - African Wildlife Ontology

- **Déclaration de Classes** : restrictions sur les propriétés

```
<owl:Class rdf:about="#Giraffe">

      <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="eaten-by"/>
            <owl:minCardinality
                        rdf:datatype=&xsd;nonNegativeInteger>
                    1
            </owl:minCardinality>
        </owl:Restriction>
     </rdfs:subClassOf>

</owl:Class>
```



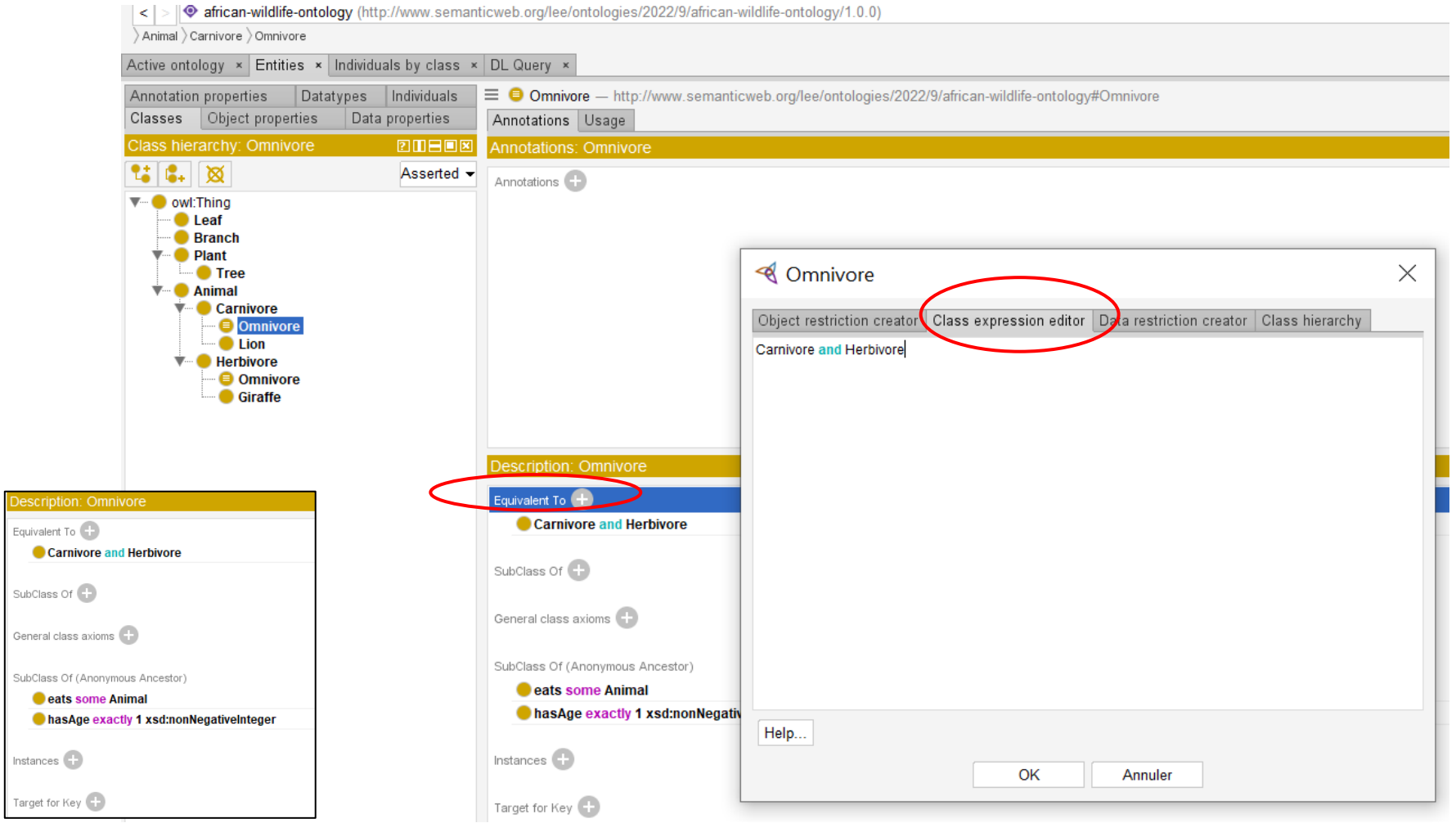
Exemple - African Wildlife Ontology

- **Déclaration de Classes** : intersectionOf

```
<owl:Class rdf:ID="Omnivore">

   <owl:equivalentClass>
      <owl:Class>
         <owl:intersectionOf rdf:parseType="Collection">
            <owl:Class rdf:about="#Herbivore"/>
            <owl:Class rdf:about="#Carnivore"/>
         </owl:intersectionOf>
      </owl:Class>
  </owl:equivalentClass>

</owl:Class>
```

Exemple - African Wildlife Ontology

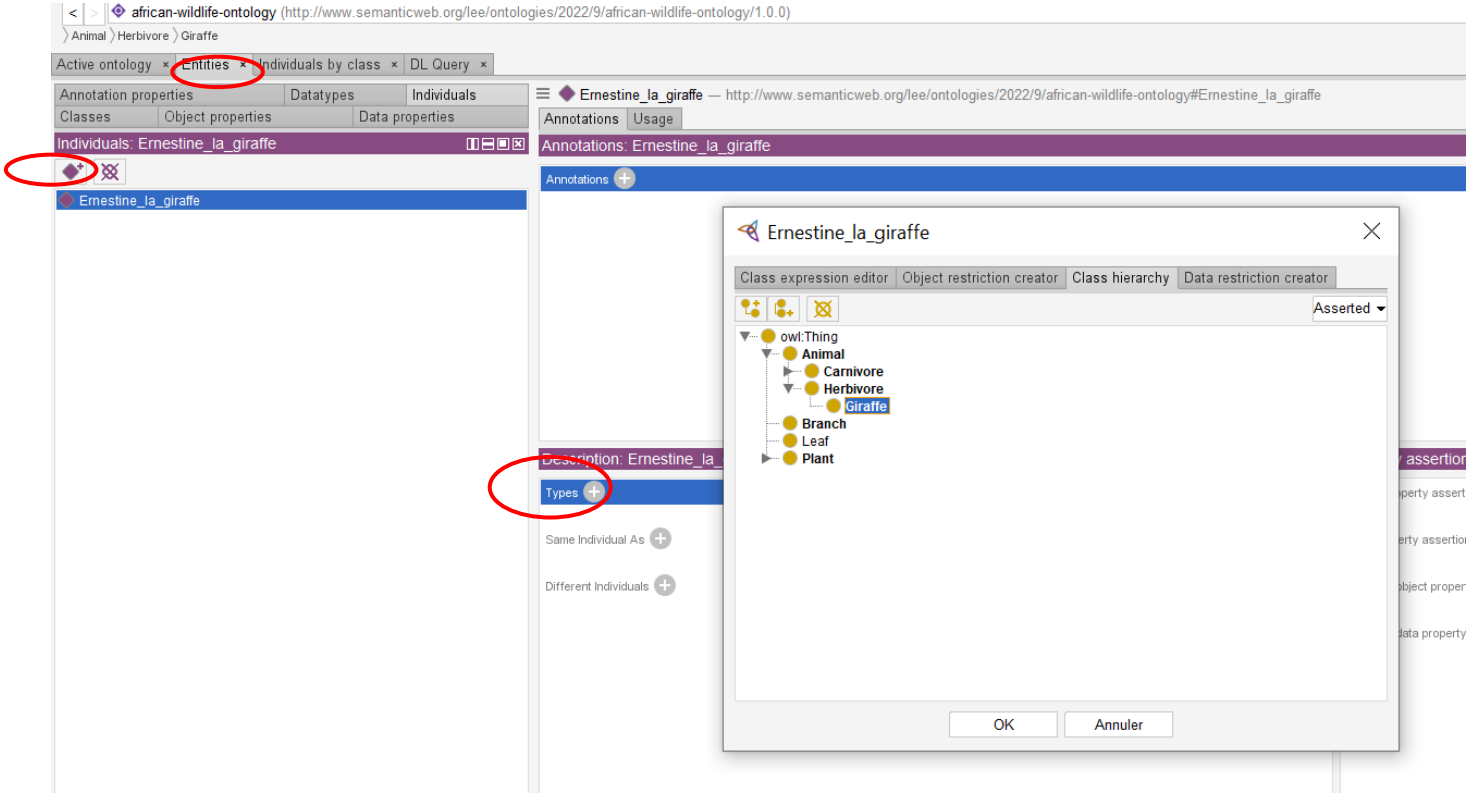
- **Déclaration de Classes** : intersectionOf – dans Protégé



Exemple - African Wildlife Ontology

- **Déclaration des instances (individuals)** : dans Protégé

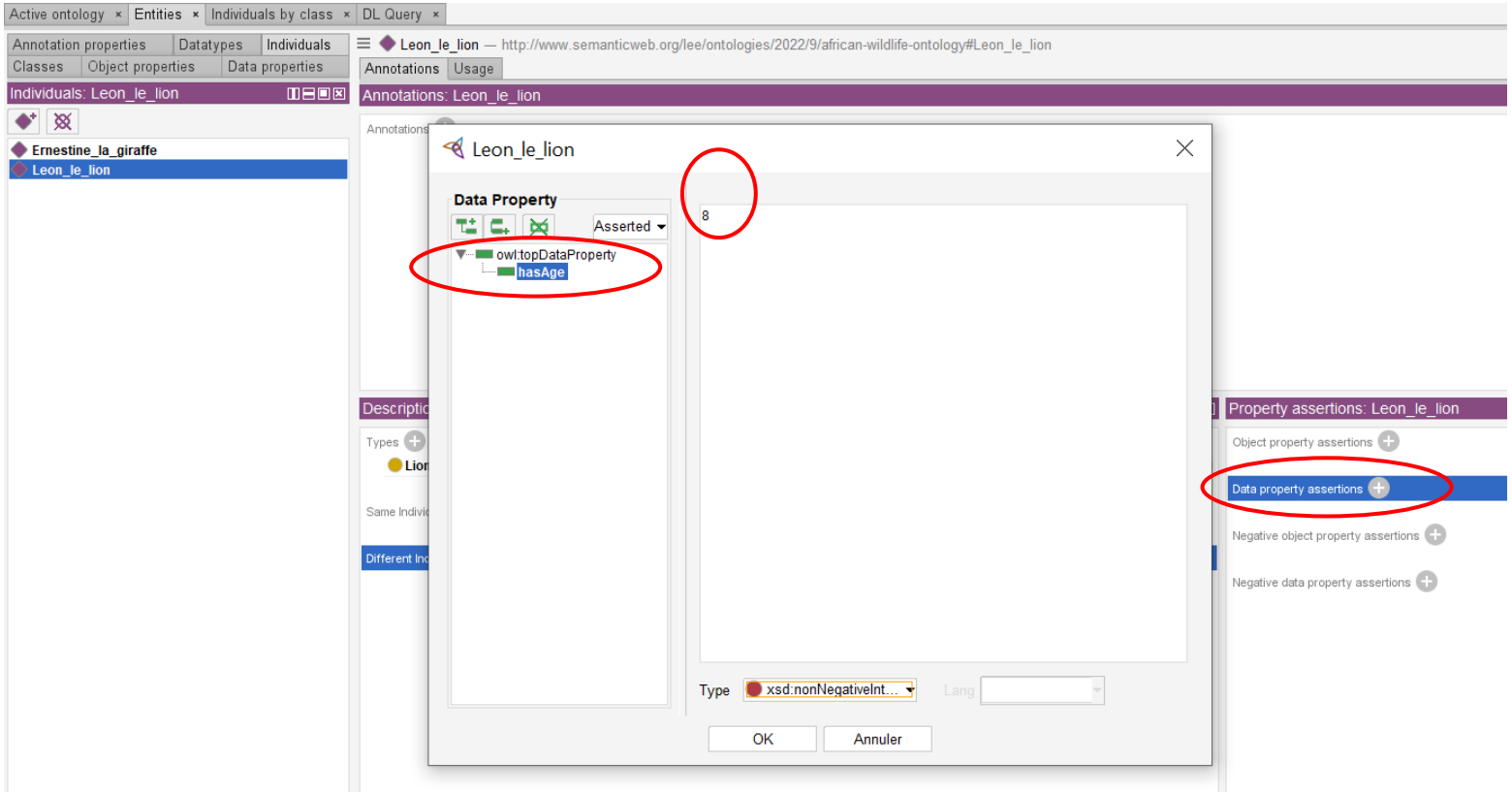
```
<Giraffe     rdf:ID="Ernestine_la_girafe"/>

<Lion     rdf:ID="Leon_le_lion"/>
```



Exemple - African Wildlife Ontology

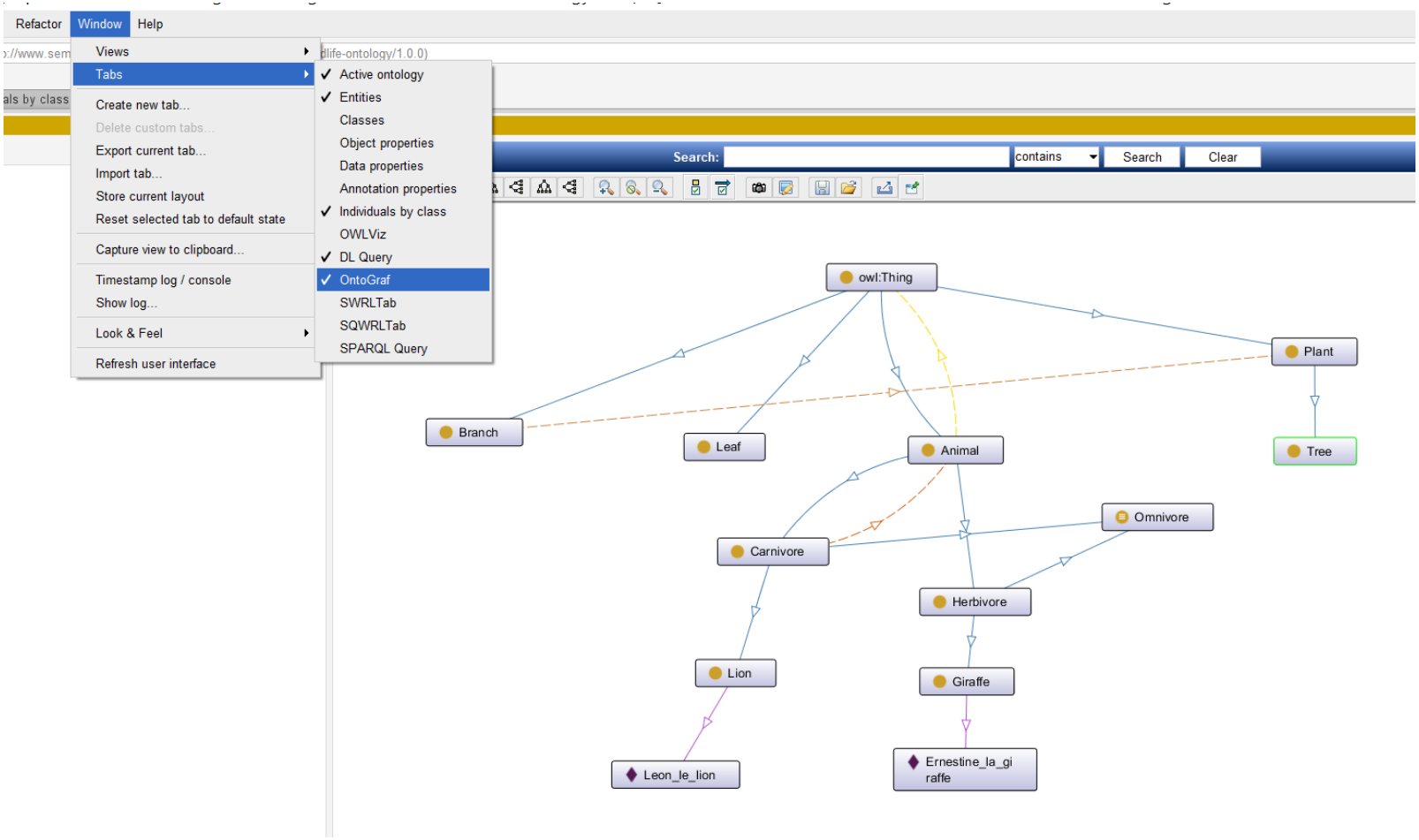
- **Déclaration des instances** : hasAge - dans Protégé

```
<Lion     rdf:about="Leon_le_lion"/>
  <hasAge rdf:datatype="&xsd;interger">8</hasAge>
</Lion>
```



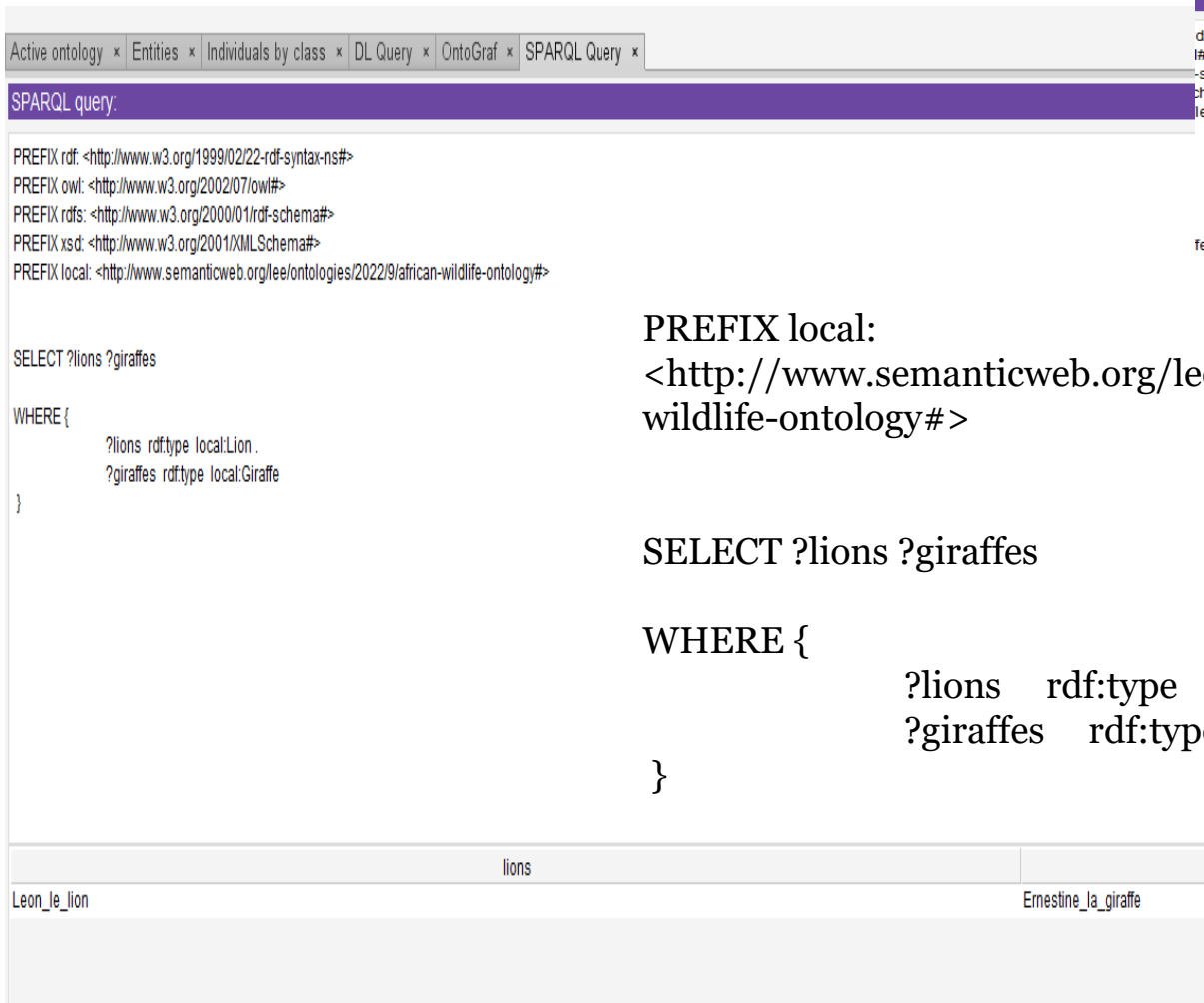
Exemple - African Wildlife Ontology

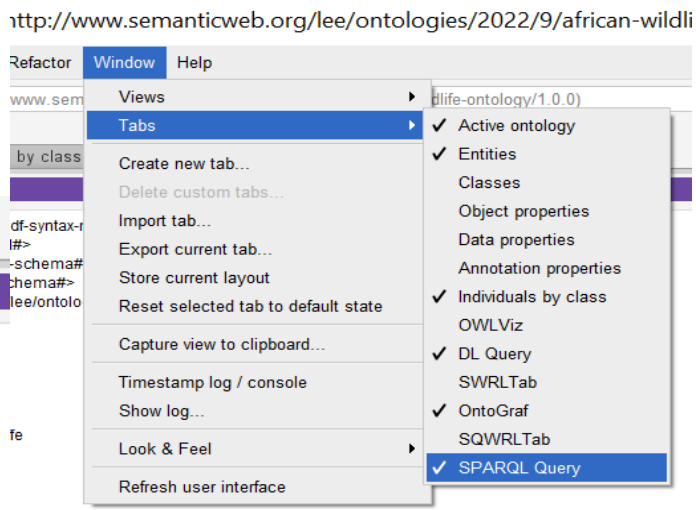
- **Visualisation de l'ontologie : OntoGraph** - dans Protégé



Exemple - African Wildlife Ontology

- **Requêtes: SPARQL Query** - dans Protégé





PREFIX local:
<http://www.semanticweb.org/lee/ontologies/2022/9/african-wildlife-ontology#>

SELECT ?lions ?giraffes

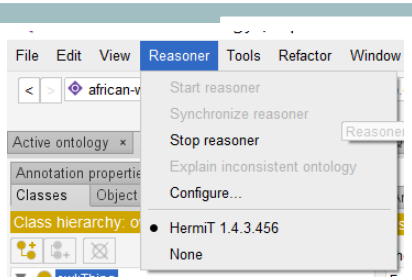
WHERE {
?lions rdf:type local:Lion .
?giraffes rdf:type local:Giraffe
}

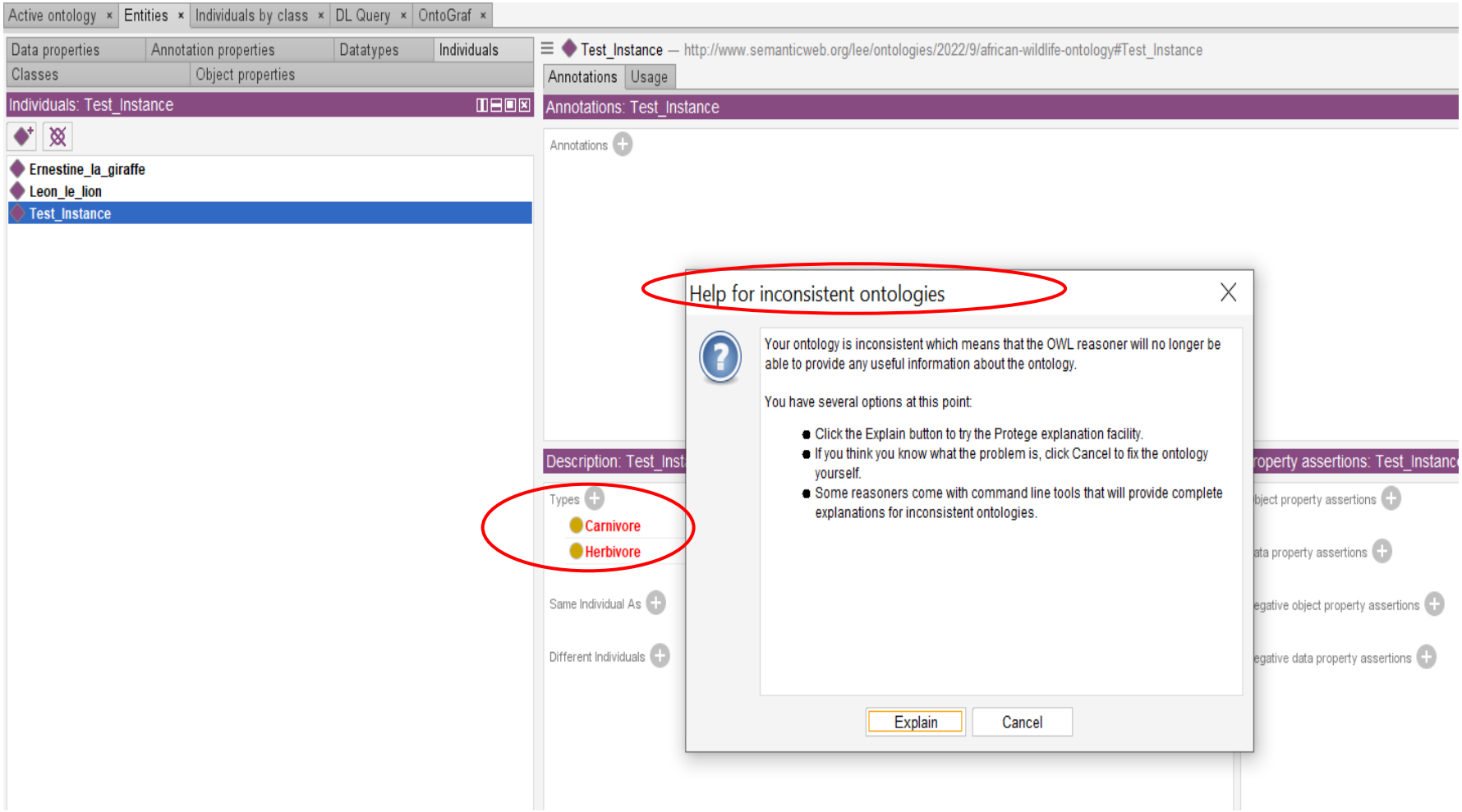
Raisonnement sur les ontologies

- Pourquoi raisonner sur les ontologies :
  - Pour vérifier la cohérence d'une ontologie.
  - Pour inférer de nouvelles connaissances, on peut utiliser un raisonneur.
- Exemple : *Vérifier*

- Supposons qu'on crée une nouvelle instance *Test_Instance*, de type Carnivore et Herbivore, qui sont déclarées disjointes.
- Si on lance le « raisonneur » de Protégé, on obtient:

Raisonnement sur les ontologies

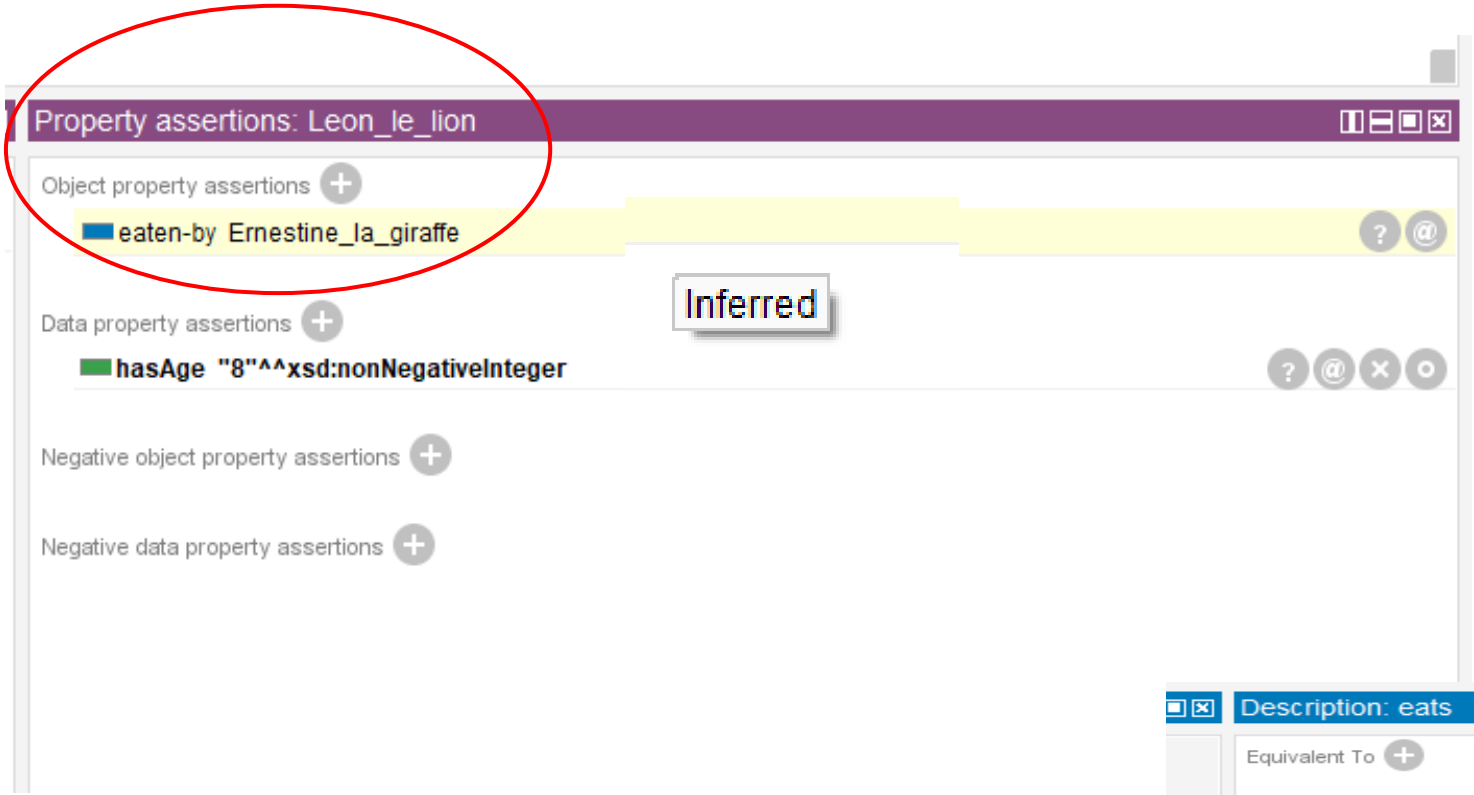




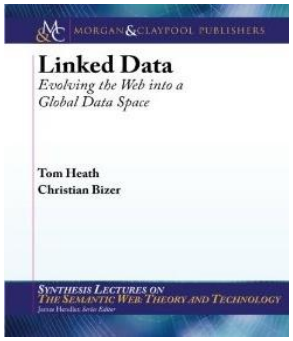
Raisonnement sur les ontologies

- Pourquoi raisonner sur les ontologies :
  - Pour vérifier la cohérence d'une ontologie.
  - Pour inférer de nouvelles connaissances, on peut utiliser un raisonneur.
- Exemple : *inférer*
- Supposons qu'*Ernestine_la_girafe* mange (*eats*) *Leon_le_lion*.
- Si on lance le « raisonneur » de Protégé, on obtient:

Raisonnement sur les ontologies

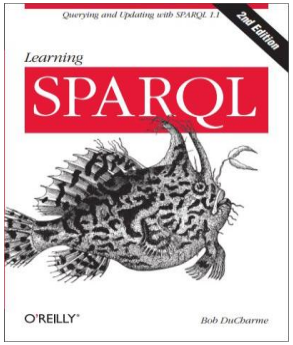


Références



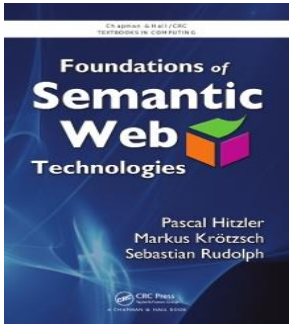
Linked Data: Evolving the Web into a Global Data Space

- ✓ Auteur : Christian Bizer, Tom Heath
- ✓ Éditeur : Morgan & Claypool Publishers
- ✓ Edition : Février 2011 - 136 pages - ISBN 9781608454310



Learning SPARQL : Querying and Updating with SPARQL

- ✓ Auteur : Bob DuCharme
- ✓ Éditeur : O'Reilly Media
- ✓ Edition: Juillet 2013– 386pages -ISBN : *9781449306595*



Foundations of Semantic Web Technologies

- ✓ Auteur : Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph
- ✓ Éditeur : CRC Press/Chapman and Hall
- ✓ Edition : 2009 - 455 pages - ISBN : 9781420090505

Références

- W3C – OWL Web Ontology Language
  - ✓ https://www.w3.org/Submission/SWRL/
- INRIA MOOC - Fabien Gandon– Web Sémantique et Web de Données
  - ✓ https://www.canal-u.tv/producteurs/inria/cours_en_ligne/web_semantique_et_web_de_donnees
- Noy et McGuinness - Ontology Development 101: A Guide to Creating Your First Ontology.
  - ✓ https://protege.stanford.edu/publications/ontology_development/ontology101.pdf/
- Cours – Bernard ESPINASSE
  - ✓ https://pageperso.lis-lab.fr/bernard.espinasse/wp-content/uploads/2021/12/8-Cours-OWL-Protege-4P-1.pdf /