Ontologies et Web Sémantique

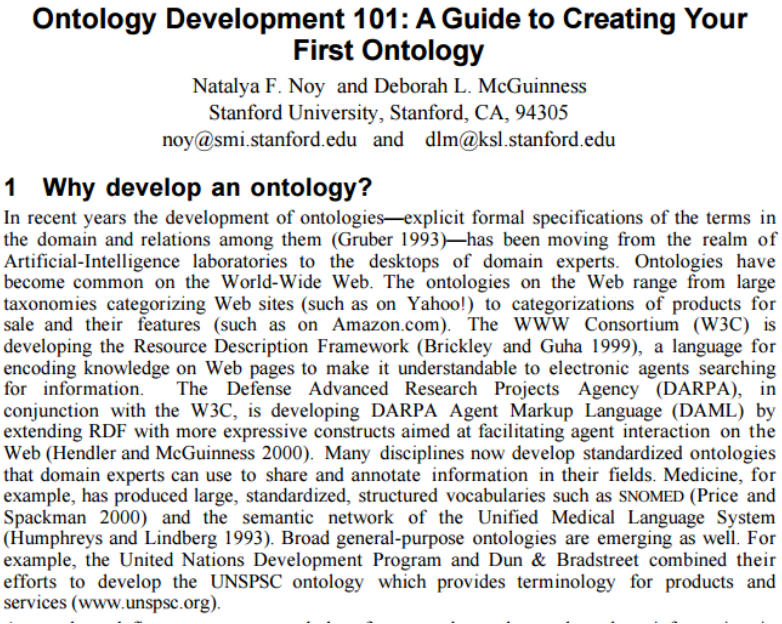
Les Ontologies – Exploitation et Inférence

Plan du cours

1. Guide de construction d'une ontologie – Noy & McGuinness
2. Inférence avec des règles - Reasoning
3. SWRL – Semantic Web Rule Language
4. Démo : Jena ARQ, Protégé, Pellet Reasoner

Guide de construction d'une ontologie

- Selon Natalya F. Noy et Deborah L. McGuinness.
- Ontology Development 101: A Guide to Creating Your First Ontology. 2001.
- « *There is no single correct ontology-design methodology and we did not attempt to define one.* »
- http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html



Ontology Development 101: A Guide to Creating Your First Ontology

Natalya F. Noy and Deborah L. McGuinness
Stanford University, Stanford, CA, 94305
noy@smi.stanford.edu and dlm@ksl.stanford.edu

1 Why develop an ontology?

In recent years the development of ontologies—explicit formal specifications of the terms in the domain and relations among them (Gruber 1993)—has been moving from the realm of Artificial-Intelligence laboratories to the desktops of domain experts. Ontologies have become common on the World-Wide Web. The ontologies on the Web range from large taxonomies categorizing Web sites (such as on Yahoo!) to categorizations of products for sale and their features (such as on Amazon.com). The WWW Consortium (W3C) is developing the Resource Description Framework (Brickley and Guha 1999), a language for encoding knowledge on Web pages to make it understandable to electronic agents searching for information. The Defense Advanced Research Projects Agency (DARPA), in conjunction with the W3C, is developing DARPA Agent Markup Language (DAML) by extending RDF with more expressive constructs aimed at facilitating agent interaction on the Web (Hendler and McGuinness 2000). Many disciplines now develop standardized ontologies that domain experts can use to share and annotate information in their fields. Medicine, for example, has produced large, standardized, structured vocabularies such as SNOMED (Price and Spackman 2000) and the semantic network of the Unified Medical Language System (Humphreys and Lindberg 1993). Broad general-purpose ontologies are emerging as well. For example, the United Nations Development Program and Dun & Bradstreet combined their efforts to develop the UNSPSC ontology which provides terminology for products and services (www.unspsc.org).

Guide de construction d'une ontologie

- Pour quelles raisons développer une ontologie ?

    1. Partager la compréhension commune de la structure de l'information entre les personnes ou les fabricants de logiciels.
    2. Permettre la réutilisation du savoir sur un domaine.
    3. Expliciter ce qui est considéré comme implicite sur un domaine.
    4. Distinguer le savoir sur un domaine du savoir opérationnel.
    5. Analyser le savoir sur un domaine.

- Terminologie - Ontologie :

    - ✓ Classes / Concepts.
    - ✓ Slots / Propriétés / Rôles / Attributs.
    - ✓ Restrictions sur les propriétés / Facets.
    - ✓ Instances / Individus / Base de connaissances.

Guide de construction d'une ontologie

- En termes pratiques, développer une ontologie inclut :
  - définir les classes dans l'ontologie.
  - arranger les classes en une hiérarchie taxinomique.
  - définir les propriétés et décrire leurs valeurs autorisées.
  - renseigner les valeurs pour les propriétés des instances.

- Puis, créer une base de connaissances en définissant les instances individuelles de ces classes.

Guide de construction d'une ontologie

- Règles fondamentales dans la conception des ontologies :

1. Il n'y a pas qu'une seule façon correcte pour modéliser un domaine - il y a toujours des alternatives viables.
2. La meilleure solution dépend presque toujours de l'application et du but que vous voulez mettre en place et des évolutions que vous anticipez.
3. Le développement d'une ontologie est nécessairement un processus itératif.
4. Les concepts dans une ontologie doivent être très proches des objets (physiques ou logiques) et des relations dans votre domaine d'intérêt.
5. Fort probablement ils sont des noms (objets) ou verbes (relations) dans des phrases qui décrivent votre domaine. Refléter la réalité.

Guide de construction d'une ontologie

- Les étapes :
- **Step 1** – Déterminer le domaine et la porté de l'ontologie
- Répondre à quelques questions de base :
  - Quel est le domaine que va couvrir l'ontologie ?
  - Dans quel but utiliserons nous l'ontologie ?
  - A quels types de questions l'ontologie devra-t-elle fournir des réponses ?
  - Qui va utiliser et maintenir l'ontologie ?
- Liste de questions auxquelles une base de connaissances fondée sur une ontologie devrait pouvoir répondre, appelées questions de compétence.

Guide de construction d'une ontologie

- Les étapes :
- **Step 2** – Envisager une éventuelle réutilisation des ontologies existantes

- Considérer ce que d'autres personnes ont fait.
- Examiner si nous pouvons élargir des sources existantes et les affiner pour répondre aux besoins de notre domaine ou de notre tâche particulière.
- Réutiliser des ontologies existantes peut même constituer une exigence si notre système a besoin d'interagir avec d'autres applications qui utilisent déjà des ontologies spécifiques ou des vocabulaires contrôlés.
- Il existe des bibliothèques d'ontologies réutilisables sur le Web et dans la littérature. Ontolingua, DAML, DMOZ, UNSPSC, RosettaNET, etc.

Guide de construction d'une ontologie

- Les étapes :

- **Step 3** – Enumérer les termes importants dans une ontologie

- **Step 4** – Définir les classes et les hiérarchies de classes

- Un procédé de développement de **haut en bas** commence par une définition des concepts les plus généraux du domaine et se poursuit par la spécialisation des concepts.

- Un procédé de développement **de bas en haut** commence par la définition des classes les plus spécifiques, les feuilles d'une hiérarchie, et se poursuit avec le regroupement de ces classes en concepts plus généraux.

- Une procédé **combiné.**

- La hiérarchie des classes représente une relation de type **est-un** (**is-a**).
- Si une classe A est super-classe d'une classe B, alors toute instance de B est également, une instance de A. B **kindOf** A / B **une sorte de** A.

Guide de construction d'une ontologie

- Les étapes :
- **Step 5** – Définir le propriétés des classes - Slots
- **Step 6** – Définir les facettes des propriétés
  - La cardinalité : nombre de valeurs possibles. Unique ou multiple.
  - Le Type de valeur. String, booléen, nombre, instance, énuméré, etc.
  - Les Domain et Range.
- **Step 7** – Créer les instances
  - Définir une instance individuelle d'une classe exige de :
    1. choisir une classe.
    2. créer une instance individuelle de cette classe.
    3. la renseigner avec les valeurs des attributs.

Guide de construction d'une ontologie

- Recommandations:

  - Utiliser toujours et exclusivement, soit le singulier, soit le pluriel dans la nomination des classes.

  - Il est important de distinguer entre une classe et son nom : *Les synonymes pour le même concept ne représentent pas de classes différentes*.

  - Eviter des boucles dans la hiérarchie d'une classe : une classe A a une sous-classe B et qu'en même temps B est une super-classe de A.

  - Les enfants de mêmes parents doivent être des concepts "appartenant à la même génération".

  - Il n'y a pas de règle rigide concernant le nombre de sous-classes directes qu'une classe doit avoir. Mais en moyenne entre 2 et une douzaine.

Guide de construction d'une ontologie

- <u>Recommandations:</u>

- L'héritage multiple est possible : une classe peut être une sous-classe de plusieurs classes.

- Instance ou classe ? Les instances individuelles sont les concepts les plus spécifiques représentés dans une base de connaissance.

- Instance ou classe ? Si les concepts forment une hiérarchie naturelle, alors ils doivent être représentés comme des classes.

- Limiter la portée. L'ontologie ne doit pas contenir toute l'information possible sur le domaine : vous ne devez pas spécialiser (ou généraliser) plus que de besoin pour votre application.

- De même, L'ontologie ne doit pas contenir toutes les propriétés possibles des classes et toutes les distinctions entre les classes dans la hiérarchie.

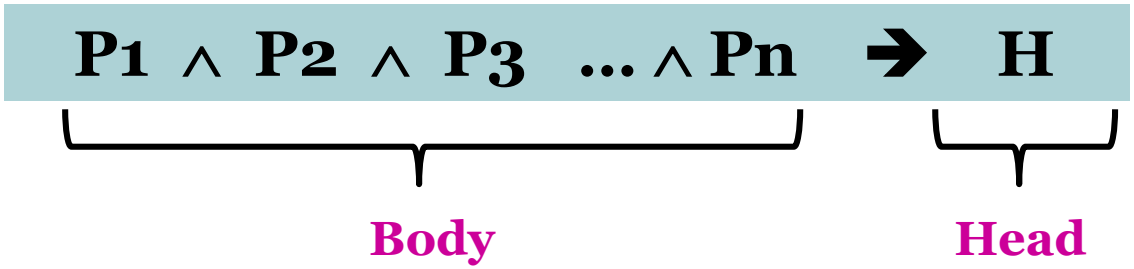
Inférence avec règles - Reasoning

- Comment inférer de nouveaux faits qui ne sont pas représentées explicitement dans une ontologie ?
- Il y a des déclarations qui ne peuvent pas être exprimées avec OWL.
- L'expressivité de OWL peut être étendu en ajoutant des **règles** à l'ontologie.
- Les règles sont exprimées sous forme d'implications logiques – Clauses de Horn.

Si prémisses alors conclusion

$$P1 \wedge P2 \wedge P3 \ldots \wedge Pn \rightarrow H$$

Inférence avec règles - Reasoning



- Constantes, variables, fonctions, etc. mais pas de négation.
- Les quantificateurs (il existe, pour tout) sont implicites.

- Les variables dans Body sont universellement quantifiées.
- Les variables uniquement dans Head sont quantifiées existentiellement.

- Exemple : isParent(X) ➔ hasChild(X, Y)
- Est équivalente :

Pour tout X : isParent(X), il existe Y tq : hasChild(X, Y)

Inférence avec règles - Reasoning

- Raisonner sur les ontologies :
  - Reasoning terminologique :

✓ Contrôle de cohérence / Consistency Check : recherche de définitions de classes incohérentes.

✓ Classification : détermine les concepts qui subsume immédiatement ou sont subsumés par un concept donné.

✓ Construction de taxonomie : calcule toutes les relations de sous-classe (y compris ceux qui ne sont pas explicitement énoncés, mais qui sont impliqués par les définitions données).

Inférence avec règles - Reasoning

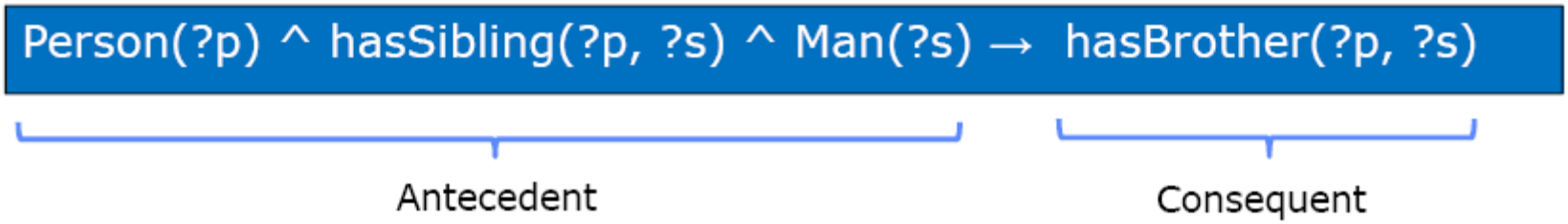
- Raisonner sur les ontologies :
  - Reasoning sur les instances :

✓ Vérification d'instance - étant donné une description partielle d'une instance et une description de classe, rechercher si la classe décrit l'instance.

✓ Recherche d'instances - trouve toutes les instances qui sont décrites par un concept donné.

SWRL - Semantic Web Rule Language

- SWRL – Semantic Web Rule Language.
- W3C member submission. http ://www.w3.org/Submission/SWRL/
- Langage de règles basé sur OWL.
- Fournit des capacités d'inférences supérieures à OWL seul.
- Combinaison de OWL DL (ontologies) et de RuleML (règles).
- Si antécédent alors conséquent - if A then B
- Exemple règle SWRL : (Syntaxe Abstraite Human Readable)



SWRL - Semantic Web Rule Language

- Règle SWRL :

  Si antécédent alors conséquent

- Antécédent : conjonctions d'atomes.
- Conséquent : conjonctions d'atomes.
- Atome :
  - C(x) où C est un concept OWL.
  - P(x, y) où P est propriété OWL.
  - sameAs(x, y) et differentFrom(x, y).
  - builtIn(r, x, ...) : atome avec prédicat prédéfini "built-in"
  - x, y : variables, individus OWL, valeurs de données OWL
  - R : relation built-in.
  - noms : RDF URI

SWRL - Semantic Web Rule Language

- Règle SWRL :

  Si antécédent alors conséquent

- Exemples d'atomes :

  - Personne(?p)
  - xsd :int(?x)
  - aEnfant(?x, ?y)
  - aAge(?x, ?age), aAge(?x, 100)
  - differentFrom(?x, ?y), sameAs(?x, ?y),
  - swrlb:cos(?x), swrlb:add(?x,2,3).

SWRL - Semantic Web Rule Language

- SWRL builts-ins fournissent des fonctions pour : comparaison, mathématiques, manipulation des chaines de caractères, etc.
- Sont identifiés en utilisant l'espace de nommage :

  http ://www.w3.org/2003/11/swrlb

- Exemples :

hasBrother(?x1,?x2) ^ hasAge(?x1,?age1) ^ hasAge(?x2,?age2) ^
swrlb:greaterThan(?age2,?age1)
→ hasOlderBrother(?x1,?x2)

SWRL - Semantic Web Rule Language

Syntaxe XML concrète

- Combinaison de OWL XML et RuleML XML.
- Utilise les espaces de nommage suivants :

```
swrlx :     http ://www.w3.org/2003/11/swrlx
ruleml :    http ://www.w3.org/2003/11/ruleml
owlx :      http ://www.w3.org/2003/105/owl-xml
xsd :       http ://www.w3.org/2001/XMLSchema
swrlb :     http ://www.w3.org/2003/11/swrlb
```

- Ontology root :

Syntaxe OWL XML et non pas OWL en RDF/XML

```
<swrlx:Ontology swrlx:name = xsd:baseURI
                    namespaces … >
       // Content

</swrlx:Ontology>
```

SWRL - Semantic Web Rule Language

Syntaxe XML concrète

- Contenu :

```
<swrlx:Ontology
  swrlx:name = xsd:anyURI
>
  Content: (owlx:VersionInfo | owlx:PriorVersion | owlx:BackwardCompatibleWith |
            owlx:IncompatibleWith | owlx:Imports | owlx:Annotation |
            owlx:Class[axiom] | owlx:EnumeratedClass(D,F) |
            owlx:SubClassOf(D,F) | owlx:EquivalentClasses | owlx:DisjointClasses(D,F) |
            owlx:DatatypeProperty | owlx:ObjectProperty |
            owlx:SubPropertyOf | owlx:EquivalentProperties |
            owlx:Individual[axiom] | owlx:SameIndividual | owlx:DifferentIndividuals |
            ruleml:imp[axiom] | ruleml:var[axiom])*
</swrlx:Ontology>
```

```
<ruleml:var>var</ruleml:var>
```

```
<ruleml:imp>
        Content: (_rlab, _head, _body)
</ruleml:imp>
```

SWRL - Semantic Web Rule Language

Syntaxe XML concrète

- Contenu ruleml:imp :

```
<ruleml:_rlab ruleml:href = xsd:anyURI (required) />
```

```
<ruleml:_body>
        Content: ( swrlx:atom )
</ruleml:_body>
```

```
<ruleml:_head>
        Content: ( swrlx:atom )
</ruleml:_head>
```

Model group ***swrlx:atom***

Content: (swrlx:classAtom | swrlx:datarangeAtom |
swrlx:individualPropertyAtom | swrlx:datavaluedPropertyAtom |
swrlx:sameIndividualAtom | swrlx:differentIndividualsAtom |
swrlx:builtinAtom)

SWRL - Semantic Web Rule Language

Syntaxe XML concrète

- Exemples : **classAtom**

```
<swrlx:classAtom>
       <owlx:Class owlx:name="Person" />
       <ruleml:var>x1</ruleml:var>
</swrlx:classAtom>

<swrlx:classAtom>
   <owlx:IntersectionOf>
     <owlx:Class owlx:name="Person" />
     <owlx:ObjectRestriction owlx:property="hasParent">
         <owlx:someValuesFrom owlx:class="Physician" />
     </owlx:ObjectRestriction>
   </owlx:IntersectionOf>
  <ruleml:var>x2</ruleml:var>
</swrlx:classAtom>
```

SWRL - Semantic Web Rule Language

Syntaxe XML concrète

- Exemples : **datarangeAtom**

```
<swrlx:datarangeAtom>
        <owlx:Datatype owlx:name="&xsd;int" />
        <ruleml:var>x1</ruleml:var>
</swrlx:datarangeAtom>

<swrlx:datarangeAtom>
    <owlx:OneOf>
        <owlx:DataValue owlx:datatype="&xsd;int">   5
        </owlx:DataValue>
        <owlx:DataValue owlx:datatype="&xsd;int">   10
        </owlx:DataValue>
    </owlx:OneOf>
    <ruleml:var>x2</ruleml:var>
</swrlx:datarangeAtom>
```

SWRL - Semantic Web Rule Language

Syntaxe XML concrète

- Exemples : **individualPropertyAtom /datavaluedPropertyAtom**

```
<swrlx:individualPropertyAtom swrlx:property="hasParent">
    <ruleml:var>x1</ruleml:var>
    <owlx:Individual owlx:name="John" />
</swrlx:individualPropertyAtom>


<swrlx:datavaluedPropertyAtom swrlx:property="grade">
   <ruleml:var>x1</ruleml:var>
   <owlx:DataValue owlx:datatype="&xsd;int">   12
   </owlx:DataValue>
</swrlx:datavaluedPropertyAtom>
```

SWRL - Semantic Web Rule Language

Syntaxe XML concrète

- Exemples : **builtinAtom**

```
<swrlx:builtinAtom
            swrlx:builtin="&swrlb;#greaterThanOrEqual">
    <ruleml:var>val</ruleml:var>
    <owlx:DataValue owlx:datatype="&xsd;#int">  10
    </owlx:DataValue>
</swrlx:builtinAtom>
```

SWRL - Semantic Web Rule Language

- Exemple Règle SWRL :

```
<ruleml:imp>
   <ruleml:_rlab ruleml:href="#example"/>
   <ruleml:_body>
      <swrlx:individualPropertyAtom swrlx:property="hasParent">
          <ruleml:var>x1</ruleml:var>
          <ruleml:var>x2</ruleml:var>
      </swrlx:individualPropertyAtom>
      <swrlx:individualPropertyAtom swrlx:property="hasBrother">
          <ruleml:var>x2</ruleml:var>
          <ruleml:var>x3</ruleml:var>
      </swrlx:individualPropertyAtom>
   </ruleml:_body>
   <ruleml:_head>
      <swrlx:individualPropertyAtom swrlx:property="hasUncle">
          <ruleml:var>x1</ruleml:var>
          <ruleml:var>x3</ruleml:var>
      </swrlx:individualPropertyAtom>
   </ruleml:_head>
</ruleml:imp>
```

Quelques Outils - Démo

1. **Jena ARQ** – SPARQL Processor for Jena

- Pré-requis : Java JDK + Variables d'environnement *PATH*
- Télécharger Apache Jena : https://jena.apache.org/download/index.cgi
- Dé zipper et extraire le dossier.
- Ajouter la variables d'environnement système *JENAROOT* avec comme valeur le chemin du dossier extrait Jena.
- Ajouter à la variable d'environnement *PATH* le chemin du dossier ***bat*** de Jena.
- Pour tester la bonne installation : cmd.exe ➔ entrer la commande : sparql --help

Quelques Outils - Démo

1. **Jena ARQ** – SPARQL Processor for Jena

- Interroger un fichier RDF : Web ou Local

```
> sparql --data rdfFile --query sparqlFile
```

- Exemples : (Exo 1 - TD 3)

```
> sparql --data D:\localFile.ttl --query D:\query.sparql
```

```
> sparql --data http://fr.dbpedia.org/data/Freeklane.rdf --
query D:\query.sparql
```

```
> sparql --data D:\localFile.ttl --data D:\Freeklane.rdf --
query D:\query.sparql
```

Quelques Outils - Démo

1. **Jena ARQ** – SPARQL Processor for Jena

- Interroger différents fichiers RDF avec des graphes nommés

```
> rsparql --service endpointURL --query sparqlFile
```

- Exemple : DBpedia Endpoint

```
> rsparql --service http://fr.dbpedia.org/sparql --query
D:\query.sparql
```

Quelques Outils - Démo

1. **Jena ARQ** – SPARQL Processor for Jena

- Interroger différents fichiers RDF avec des graphes nommés

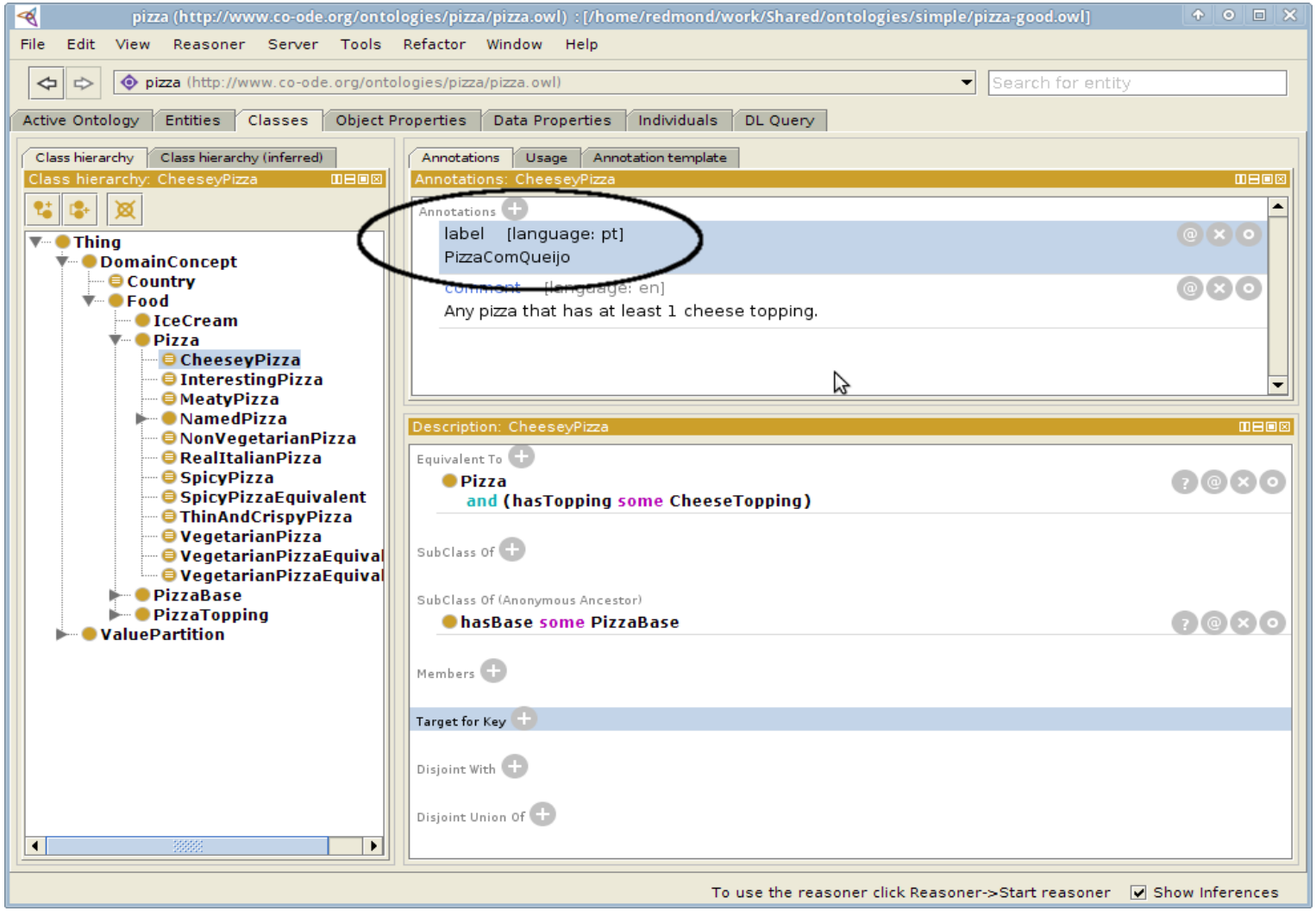
```
> sparql --graph loadFileAsDefaultGraph
         --namedGraph loadFileAsNamedGraph --query …
```

- Exemple : (TD 3 – Exo 3)

```
> sparql --graph fileexo1.ttl --namedGraph newfile.ttl --
query D:\query.sparql
```

Quelques Outils - Démo

2. **Protégé** – IDE Ontologies OWL



Quelques Outils - Démo

2. **Protégé** – IDE Ontologies OWL

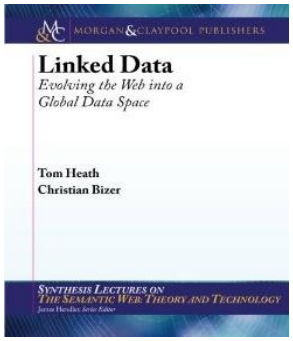
- A télécharger depuis : https://protege.stanford.edu/
- Création classes / sous-classes.
- Création de propriétés.
- Restrictions classes / propriétés.
- Paramétrages .
- Création d'instances de classes.
- Vérifier l'ontologie. Reasoning.
- Visualisation en graphe. Interrogation SPARQL.
- Etc.

Quelques Outils - Démo

3. **Pellet** – Reasoner

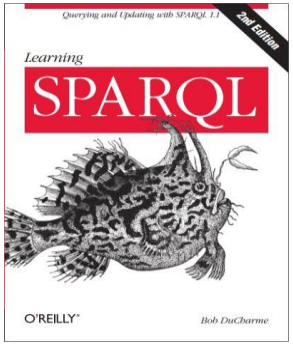
- Protégé Plugin
- Hermit, Fact++, etc.
- Classification d'une ontologie.
- Consistency Check.

Références



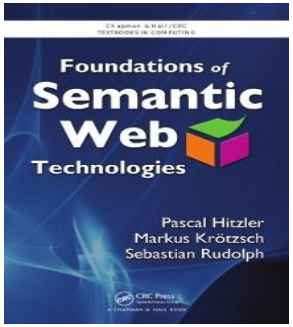
Linked Data: Evolving the Web into a Global Data Space

- ✓ 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
- Odile Papini – Cours - Ingénierie du Web Sémantique
  - ✓ http://odile.papini.perso.luminy.univ-amu.fr/sources/WEBSEM/cours-ing-web-5.pdf
- Noy et McGuinness - Ontology Development 101: A Guide to Creating Your First Ontology.
  - ✓ https://protege.stanford.edu/publications/ontology_development/ontology101.pdf/
- Cours - Knowledege Management
  - ✓ http://www-inf.it-sudparis.eu/~gaaloulw/KM/