



Ministère de l'Éducation Nationale  
Université de Montpellier II  
Place Eugène Bataillon  
34095 Montpellier Cedex 5



---

# Manipulation de schémas de concepts dédiés

## Partie 1

---

PROJET GESTION DE DONNÉES COMPLEXES GMIN332  
RAPPORT (NOVEMBRE 2012)

Travail réalisé par :

Thibaut MARMIN  
Namrata PATEL

<https://github.com/marminthibaut/gmin332/>

## 1 Développement

*L'outil développé permet, à partir d'une url pointant une ontologie, de la représenter sous la forme d'un graphe*

### 1.1 Description

Devant utiliser Jena, nous avons fait le choix de développer une application javaWeb. Celle-ci est composée de deux pages :

- La page d'accueil contenant le champ url,
- La page d'affichage du graphe.

### 1.2 J2EE

L'application est composée d'une unique servlet permettant l'accès au graph (/explore). Il est également possible de récupérer le json décrivant le graphe à l'adresse /explore/json. Deux pages JSP ont été définies permettant la définition du formulaire (default.jsp) et de l'explorer (explorer.jsp).

### 1.3 Jena

Lors de la soumission du formulaire, un OntModel de Jena est chargé à partir de l'url soumise. Un générateur de code json ajoute une classe 'Thing' (classe root) puis parcourt le modèle de l'ontologie chargé avec Jena.

### 1.4 Graphe

L'affichage du graphe se fait à l'aide de la librairie JavaScript N3 (<https://github.com/RubenVerborgh/node-n3/>). Le graphe est en faite un arbre, qui est généré à partir d'une ressource json externe (ici /explore/json).

### 1.5 Difficultés

Le OntModel de Jena peut être paramétré afin d'accepter un certain type d'ontologie. Le problème est que nous voulions faire un produit générique. Nous n'avons pas trouvé de manière pour charger correctement toutes les ontologies. Du coup plusieurs problèmes peuvent apparaître : L'ontologie ne se charge pas : c'est le pire des cas, seulement le nœud 'Thing' est présent, L'ontologie possède des classes en double. Notre application ne gère pas les classes définies avec des owl:Restrictions. Cela affecte le graphe par l'ajout de nœuds vides, qui le rendent inexploitable. Ajoutons qu'ayant très peu développé d'application java web, la mise en route à été particulièrement longue pour ce projet.

## 2 TD 6

### 2.1 Question 1

Type de jointure : S-S Patron de graphe : Étoile

### 2.2 Question 2

Type de jointure : S-S Patron de graphe : Étoile

### 2.3 Question 3

### 2.4 Question 4

### 2.5 Question 5

Retourne les voisins d'une feature (ici <http://sws.geonames.org/2985244/>) en sélectionnant le père et en récupérant les fils. Sélection du nom de chaque feature et son url (isDefinedBy).

### 2.6 Question 6

Sélection de toutes les entités qui ont au moins une relation avec une feature (relation ayant comme range Feature). Questions sur le modèle

Une feature au sens géonames peut admettre plusieurs codes, étant donné qu'aucune contrainte de cardinalité n'est définie dans l'ontologie. Geonames utilise qu'une faible partie de skos pour définir ses concepts. On y retrouve l'utilisation des classes de scheme et de concept (et donc la relation inScheme), ainsi que les comments et prefLabel. Mais aucune hiérarchie ni relations entre concepts ne sont définies (broader / narrower / related / etc...). La classe la moins décrite dans l'ontologie géonames est la classe Country. En effet elle n'hérite d'aucune classe et n'est donc définie que par un commentaire et un label.