CORNILLE Mathilde
DELOBEL Clément

16/12/2019

Rapport des TPs de Web Semantic

1. Introduction

Une ontologie est un ensemble de vocabulaire qui décrit une donnée. Ces ontologies permettent de décrire les données présentes sur le web afin de leur donner une valeur (unité, localisation, temporalité) les rendant ainsi interprétables. L'objectif de ces deux TPs de web sémantique est donc de créer et manipuler une ontologie.

2. Premier TP - Création d'une ontologie

Lors de la première séance de TP nous avons pu appréhender le concept d'ontologie à l'aide du logiciel Protégé. Nous avons travaillé sur la création et la manipulation d'une ontologie légère puis lourde.

2.1 - L'ontologie légère

Nous avons donc commencé par définir les différentes classes ainsi que leur propriétés (data et object properties). Les datas properties sont des propriétés qui relient un individu à une valeur. Les objects properties définissent un lien entre différents individus. Les trois figures ci-dessous présentent respectivement les classes définies (en jaune), les data properties (en vert) et les objects properties (en bleu). Ces figures présentent l'exemple de la classe Instant, pour laquelle nous avons défini la data property "a_timestamp" et l'object property "debute" (qui relie un Phénomène à un Instant).

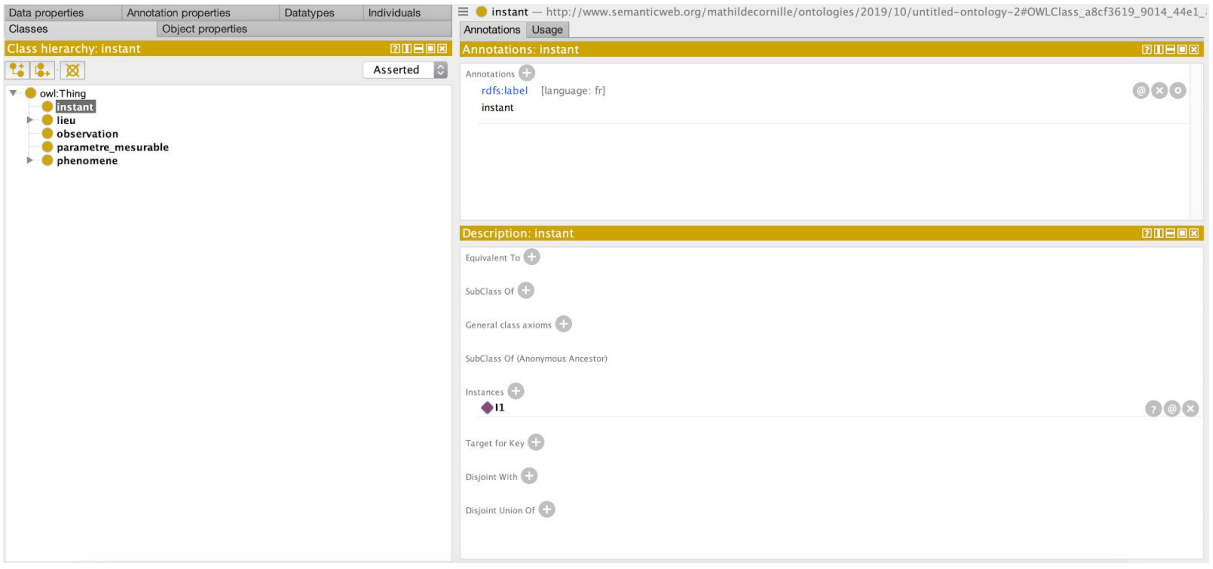

Figure n°1 : Menu "classes" sous Protégé

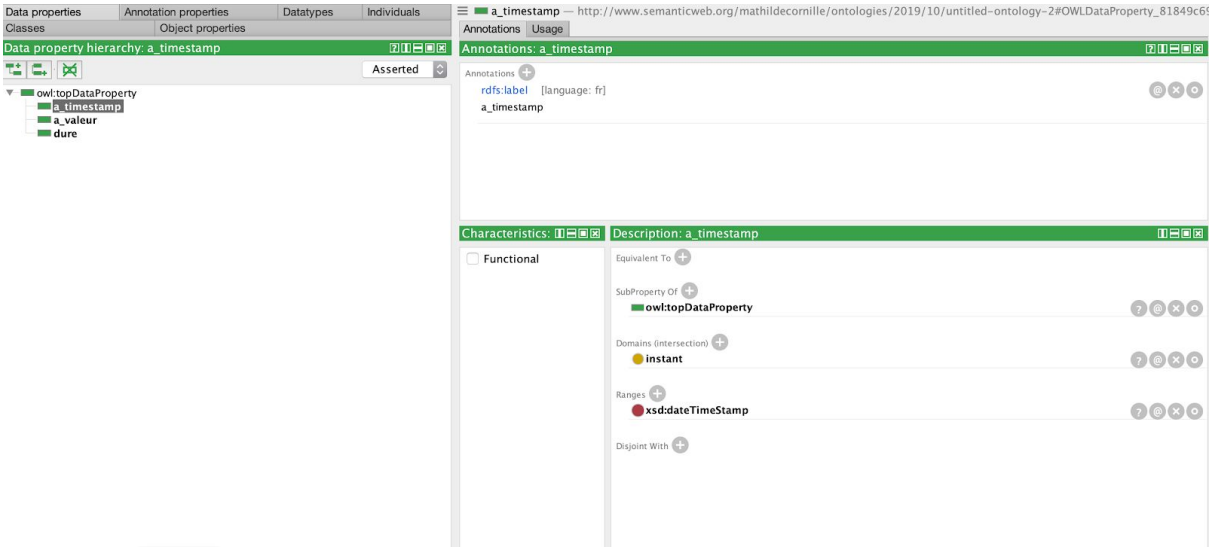

Figure n°2 : Menu "data property" sous Protégé

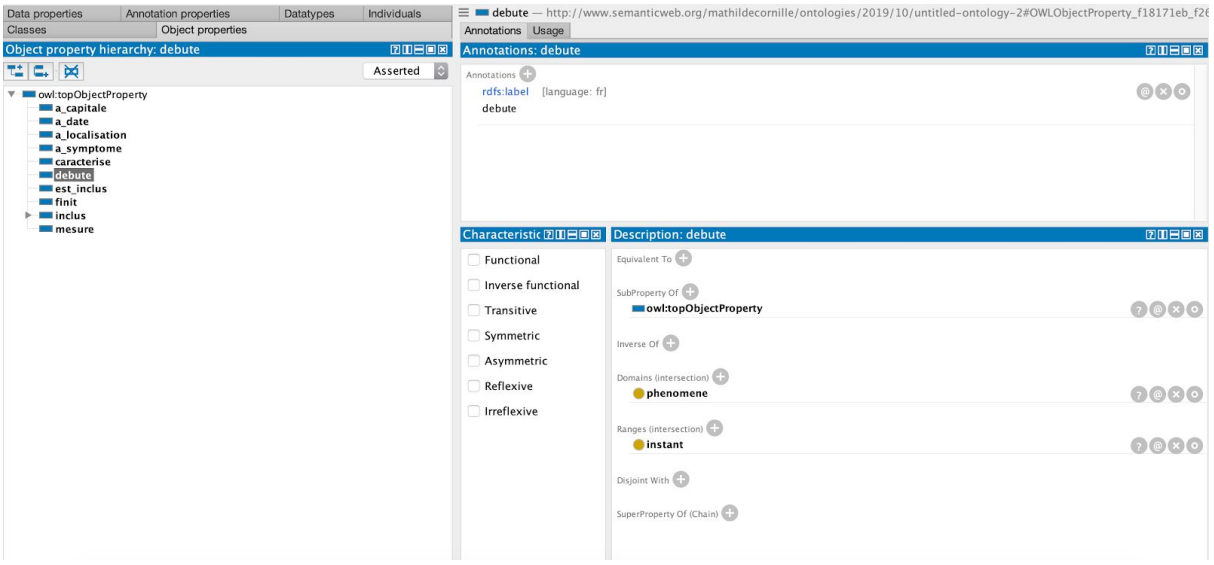

Figure n°3 : Menu "object property" sous Protégé

Le but d'une ontologie est également d'apporter une intelligence aux différents éléments qui la constituent. Pour découvrir cela nous avons donc peuplé notre ontologie en créant des individus. Lors de la création il est possible de définir la classe et les propriétés de l'entité. Cependant, si ce n'est pas le cas lors de la création, l'ontologie pourra être analysée sur Protégé par le raisonneur Hermit afin de déduire automatiquement les informations induites par les relations déjà définies.

2.2 - L'ontologie lourde

Nous avons ensuite travaillé sur l'ébauche d'une ontologie lourde. Une ontologie lourde se différencie d'une ontologie légère par l'ajout de règles entre les entités. Nous

avons ainsi pu utiliser la syntaxe de Manchester afin de définir des relations comme l'inclusion, l'unicité ou la transitivité.

Nous avons ensuite peuplé cette ontologie lourde en ajoutant des individus aux nouvelles classes. Après chaque création nous avons lancé le raisonneur Hermit et pu observer ce qui était déduit pour chaque individu. Par exemple, lorsque nous définissons deux villes distinctes comme étant la capitale d'un même pays, il est alors déduit qu'elles sont une même ville puisque nous avons définit une règle selon laquelle une pays ne peut avoir qu'une seule capitale.

Une fois que tous les éléments sont décrits correctement, il est possible de créer des règles de vérification. Si l'on implémente des capteurs avec des éléments de leur datasheet, on pourra demander à Protégé d'analyser ces données afin de s'assurer que tous les capteurs puissent résister aux températures extrêmes d'une ville nordique, par exemple.

3. Deuxième TP - Implémentation du contrôleur et du modèle en JAVA

L'objectif de la deuxième séance de TP est d'implémenter un modèle et un contrôleur en JAVA, qui permettront d'agir sur l'ontologie créée lors de la première séance, en suivant le schéma de principe ci-dessous. Pour cela, nous devions compléter une structure de code pré-remplie.

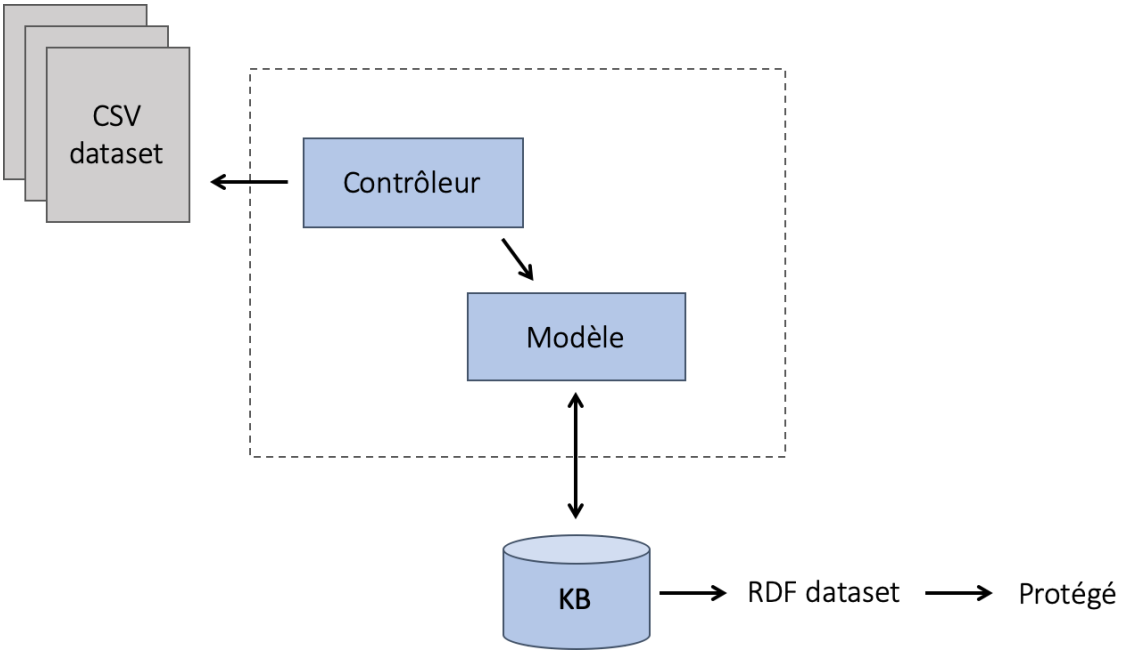

Figure n°4 : Présentation de l'architecture du TP2

Tout d'abord, nous avons implémenté les fonctions du modèle permettant d'ajouter des instances d'objet (place, instant). Ces fonctions, sont basées sur des requêtes SPARQL

qui permettent d'interagir avec la base de données. D'autres fonctions permettent, par exemple, de récupérer les URI à partir d'une instance de *Timestamp* et inversement. Enfin, une dernière fonction implémente la création d'une observation.

Afin d'assurer l'unicité des instances d'Instant nous vérifions la liste des Instants dans la fonction de création. Si un des instants de la liste possède le même TimeStamp que celui que nous souhaitons créer il n'est pas nécessaire de créer une nouvelle instance. En assurant ainsi l'unicité des instances nous pouvons utiliser un Instant comme identifiant unique. Cette opération peut se révéler coûteuse lors du peuplement de l'ontologie mais pourra par la suite grandement simplifer et rendre plus rapide son utilisation.

Ensuite, nous avons implémenté la fonction du contrôleur permettant d'instancier des observations en utilisant les fonctions du modèle, enrichissant ainsi la base de données. Nous avons enfin exporté cette nouvelle base de données afin de pouvoir l'examiner avec Protégé. Cette opération est uniquement dans un but de vérification du travail effectué, il serait, en effet, possible d'automatiser les vérifications de la base de données directement en JAVA.

Ces fonctions simplifient grandement l'accès à l'ontologie et permettraient, par exemple, d'automatiser la création d'*individus* ou de *DataProperties.* Néanmoins, l'implémentation actuelle est basée sur de nombreuses petites requêtes SPARQL ce qui n'est pas optimal. En effet, afin de réduire le temps de requête ou sa consommation de ressources, il serait plus judicieux de faire une grosse requête permettant de récupérer de nombreuses instances de la base de données ou à minima de garder en mémoire les requêtes précédentes.

4. Conclusion

Lors de ces deux séance de TP, nous avons pu manipuler les principes de base de création et d'utilisation d'une ontologie. Nous avons compris comment les objets la composant sont instanciés et reliés entre eux ou avec des données. Nous avons également appris à mettre en place des règles conditionnelles afin de vérifier la cohérence de l'ontologie. Enfin, nous avons appris à utiliser une ontologie à travers une interface JAVA utilisant des requêtes SPARQL afin d'instancier des objets dans la base de donnée.