

## Rapport de Web sémantique

Poonkuzhali Pajanissamy

Johan Alberti

Romain Saboret

January 23, 2020

# 1. Introduction

Ce projet est une introduction à la sémantique Web. Par l'utilisation de l'outil Protégé, nous avons pu créer une ontologie qui correspond à la manière de structurer nos données. Nous avons pu ensuite peupler cette ontologie avec divers exemples et grâce au raisonneur du logiciel, nous avons pu mettre au jour les incohérences entre nos exemples et la structure choisie. Cette ontologie a ensuite été mise à l'épreuve avec un dataset de température sur une ville du Danemark.

## 2. Création de l'ontologie

### 2.1 L'ontologie légère

#### Conception

Nous avons représenté la connaissance décrite à la question 2.2.1 par la création des classes :

1. *Phénomène* contenant deux sous-classes *Beau temps* et *Mauvais temps*.
2. Deux sous-classes *Pluie* et *Brouillard* dans la classe *Mauvais temps* et la création de la sous-classe *Ensoleillement* dans la classe *Beau temps*.
3. Trois classes *Paramètres mesurables*, *Instants* et *Observations*.
4. *Lieu* et de ses sous-classes *Ville*, *Pays* et *Continent*.

Et par l'ajout des propriétés :

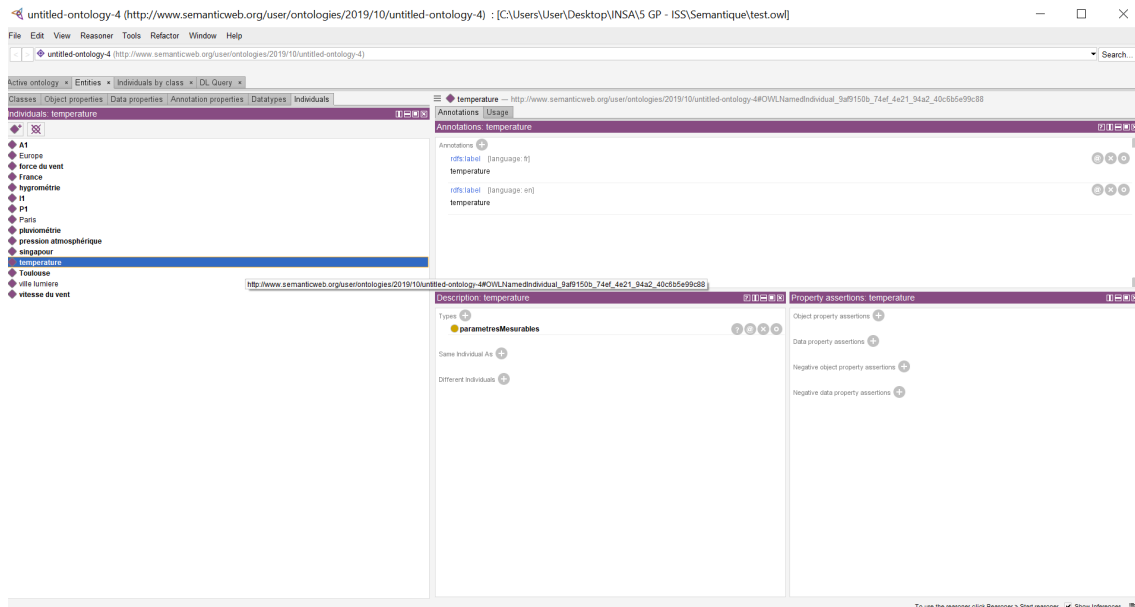
1. *est caractérisé par*(*Phénomène* - *Paramètres mesurables*) dans Object Properties
2. *a une durée* (*Phénomène*) de type xsd:float dans Data Properties
3. *début* (*Phénomène* - *Instant*) dans Object Properties
4. *fin* (*Phénomène* - *Instant*) dans Object Properties
5. *timestamp* de type xsd:dateTimeStamp dans Data Properties
6. *a pour symptôme* (*Phénomène* - *Observation*) dans Object Properties
7. *mesure* (*Observation* - *Paramètre mesurable*) dans Object Properties
8. *a une valeur* (*Observation*) dans Data Properties
9. *a pour localisation* (*Observation* - *Lieu*) dans Object Properties
10. *a pour date* (*Observation* - *Instant*) dans Object Properties
11. *est inclus dans* (*Lieu* - *Lieu*) dans Object Properties
12. *inclut* (*Lieu* - *Lieu*) dans Object Properties
13. *a pour capitale* (*Pays* - *Ville*) dans Object Properties

#### Peuplement

Après avoir représenté la connaissance de la question 2.2.2 par des *Individus* au sein de la table des *Individuals*, on constate que le raisonneur déduit les éléments suivants :

- A1 est un Phénomène.
- Paris est une ville.
- Paris est inclus dans France.

- La France inclut Paris.
- La France inclut Toulouse.
- La France est un pays.
- I1 est un instant.

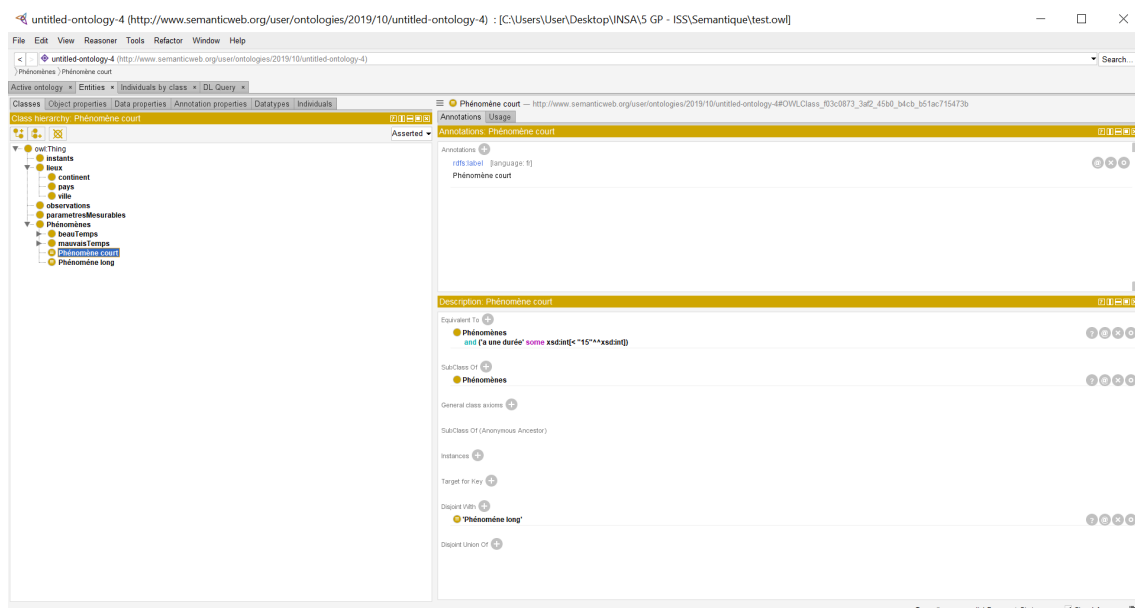


En utilisant les domain/range d'une data property, le raisonneur peut déterminer la classe d'un individu grâce aux données qu'il contient, alors que pour une object property il peut déduire les relations entre les différents individus et déterminer leur classe.

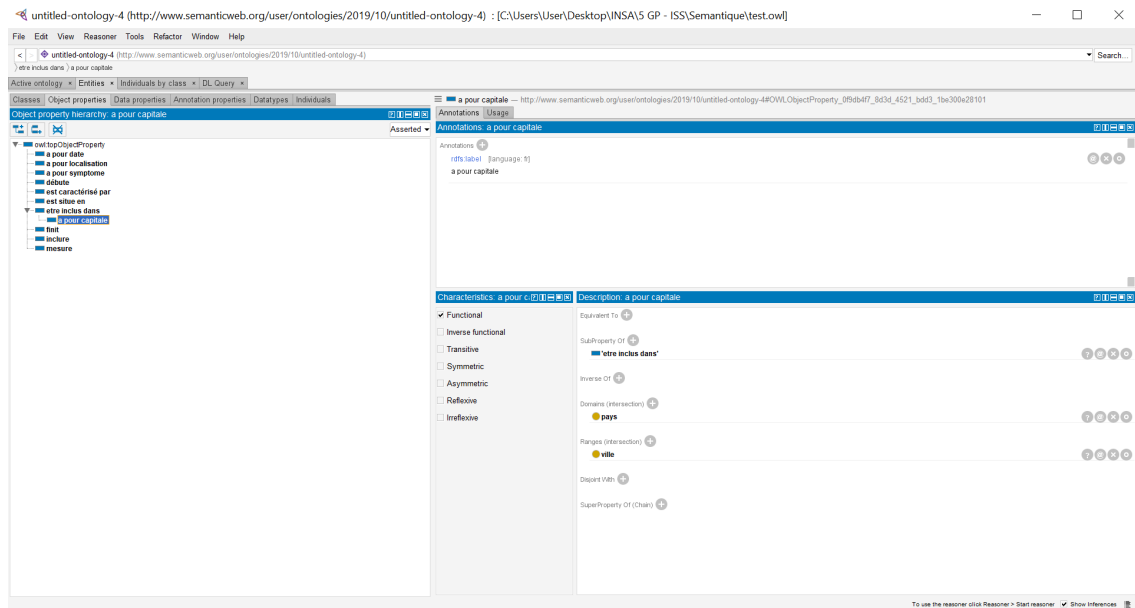
## 2.2 L'ontologie lourde

### Conception

Les axiomes logiques pouvant être représentés en owl incluent des disjonctions comme pour la question 2.2.3.1, qui assure que toute instance de ville ne peut pas être un pays.



Dans la question 2.2.3.2 et 2.2.3.9, on modélise des classes définies, qui ont pour particularité d'être des restrictions de classes déjà existantes.



Pour la question 2.2.3.8, on utilise la notion de sous-propriété : si P1 est sous-propriété de P2, alors tout individu ayant la propriété P1 a également la propriété P2.

## Peuplement

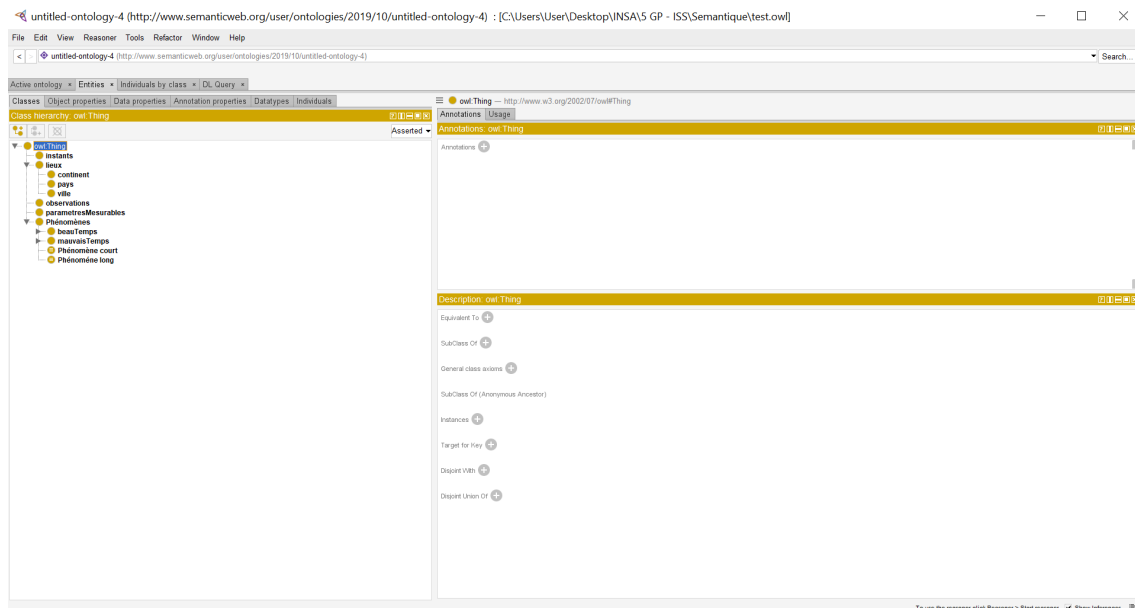
Quand on définit à la fois *Paris* et *La ville lumière* comme capitales de la France, le raisonneur déduit, de la propriété d'unicité de la capitale pour chaque pays, que Paris est La ville lumière. Le raisonneur déduit de A1 que c'est un phénomène car il sait que A1 a pour symptôme P1.

## 3. Etat de l'ontologie

Voilà l'état de notre ontologie à la fin de la première partie.

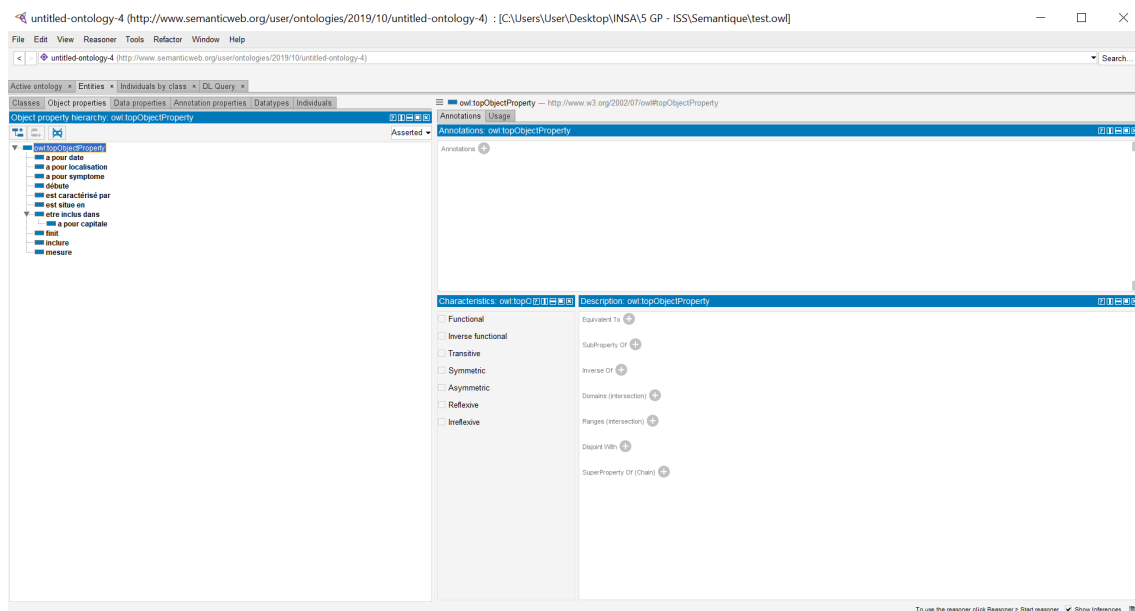
### Classes

Voici les classes de notre ontologie :



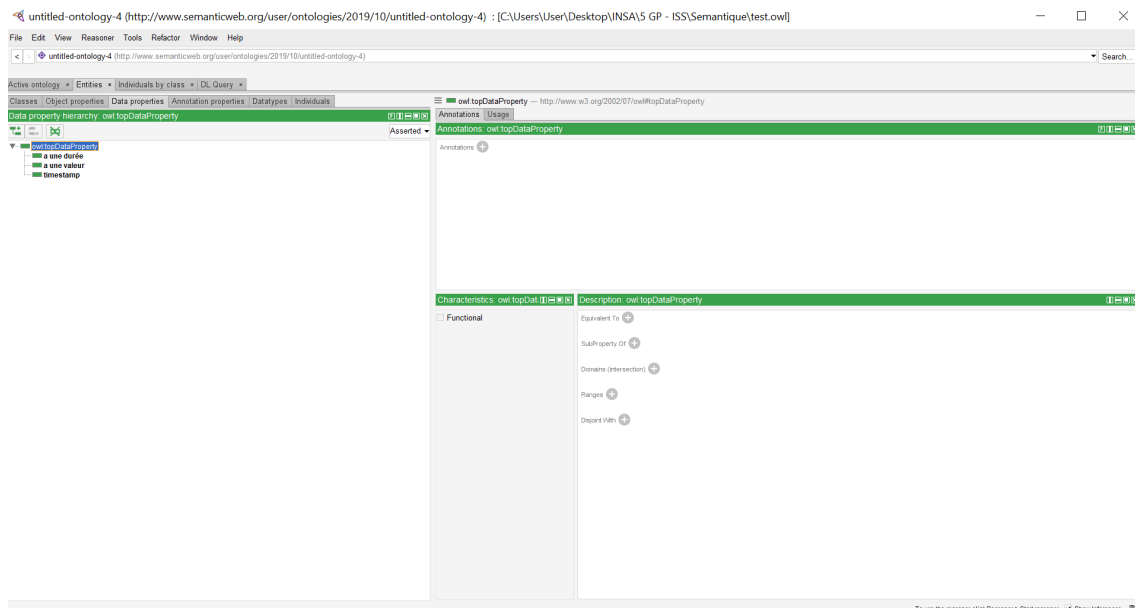
## Object Properties

Voici les Object Properties de notre ontologie :



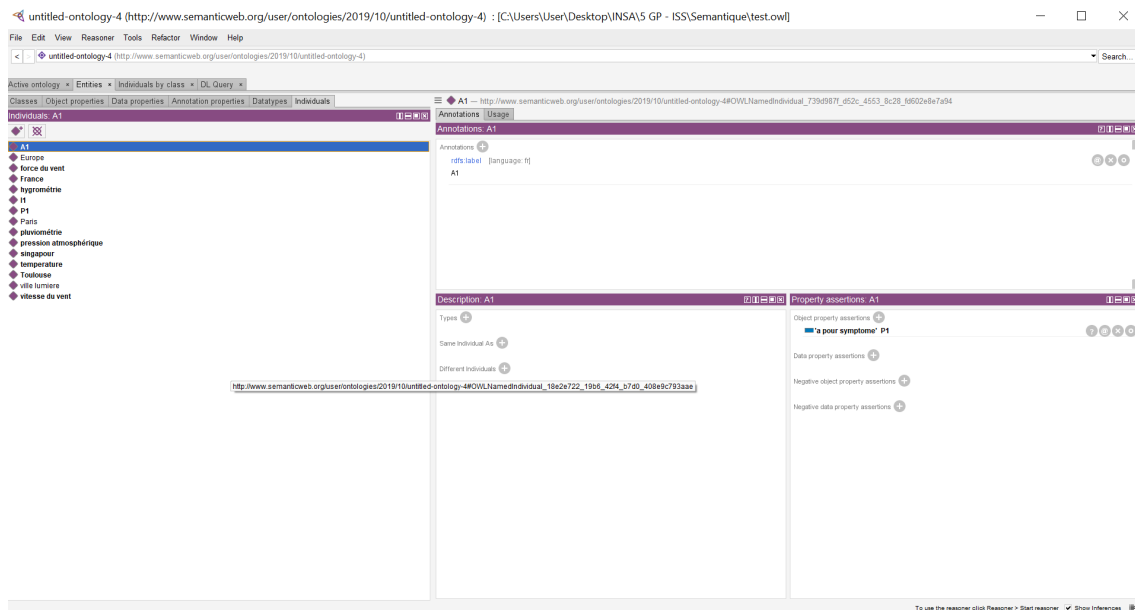
## Data Properties

Voici les Data Properties de notre ontologie :



## Individuals

Voici les individus de notre ontologie :



## 4. Exploitation de l'ontologie dans Protégé

### 4.1 Démarche générale

On se propose de peupler notre ontologie météo à l'aide de données brutes mesurées au Danemark. Ces données sont dans un format accessible (JSON), mais sont brutes car elles ne sont pas liées à une ontologie.

### 4.2 Implémentations des interfaces

- liens entre contrôleurs et données
- choix pour le cache, dont défauts

### 4.3 Exploitation dans Protégé

Deux des capteurs de température: *TemperatureSensor1* et *TemperatureSensor2*, ont une Object Property *has operating range* d'un rang de températures positives. Les mesures étant situées au Danemark, ce choix est questionnable puisqu'on imagine tout à fait que les températures puissent y être négatives.

Le dernier capteur *TemperatureSensor3* lui a un champ d'opération allant de -30C à 50C, ce qui paraît plus judicieux.

### 4.4 Différence entre Object Property et Data Property

Une Object Property est une relation ou une propriété dont le domaine est une classe d'objets, et dont le rang d'application est également une classe d'objets. On lie des objets en termes qualitatifs.

Une data property exprime une relation ou une propriété entre une classe d'objets, et une valeur (au sens atomique). On lie un objet à une donnée quantitative