Rapport Travaux pratiques Web Semantics

2020 Luan TRUONG Michael EJIGU

TP1

Objectif

Nous utilisons Protegé pour décrire des ontologies. Nous utilisons ensuite un raisonneur (HermiT) pour valider et compléter des assertions.
On nous demande des ontologies légères puis lourdes.

Fichier de description des ontologies

https://github.com/budbudmike/ISS_Semantics_Michael_Luan/blob/main/lab1.owl

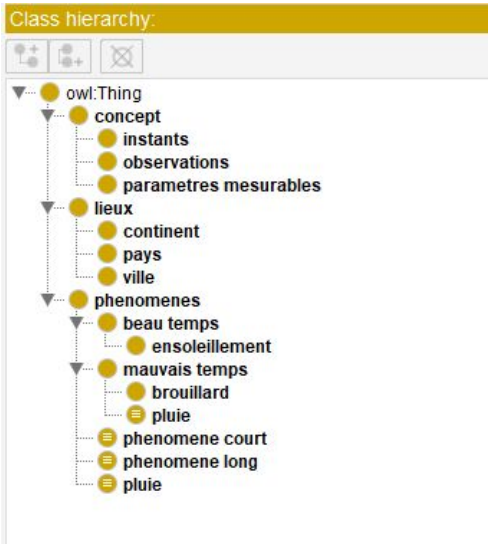
Analyse

Ontologie légère

- Description

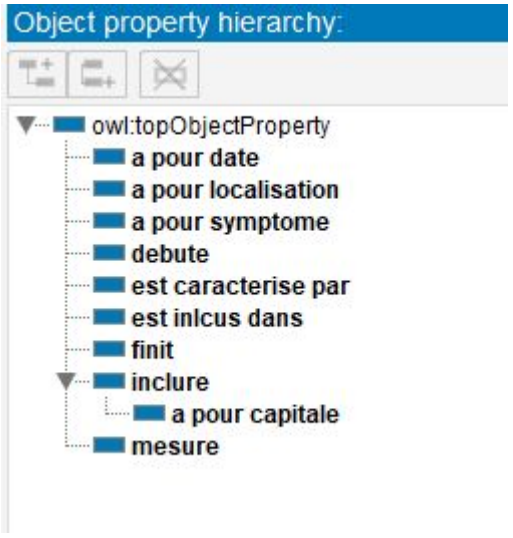
1. Le beau temps et le mauvais temps sont deux types de phénomènes.
2. La pluie et le brouillard sont des types de phénomènes de mauvais temps, l'ensoleillement est un type de ph´enom`ene de beau temps
3. Les paramètres mesurables sont une classe de concept, ainsi que les instants et les observations
4. Une ville, un pays et un continent sont des types de lieux

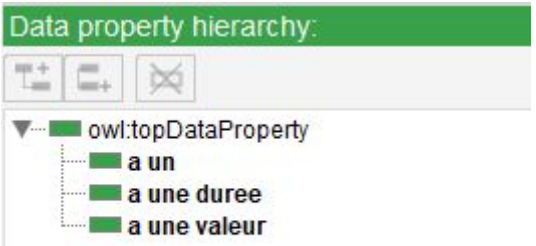
Nous créons les classes et les sous classes nécessaire comme visible sur la capture ci-dessous:



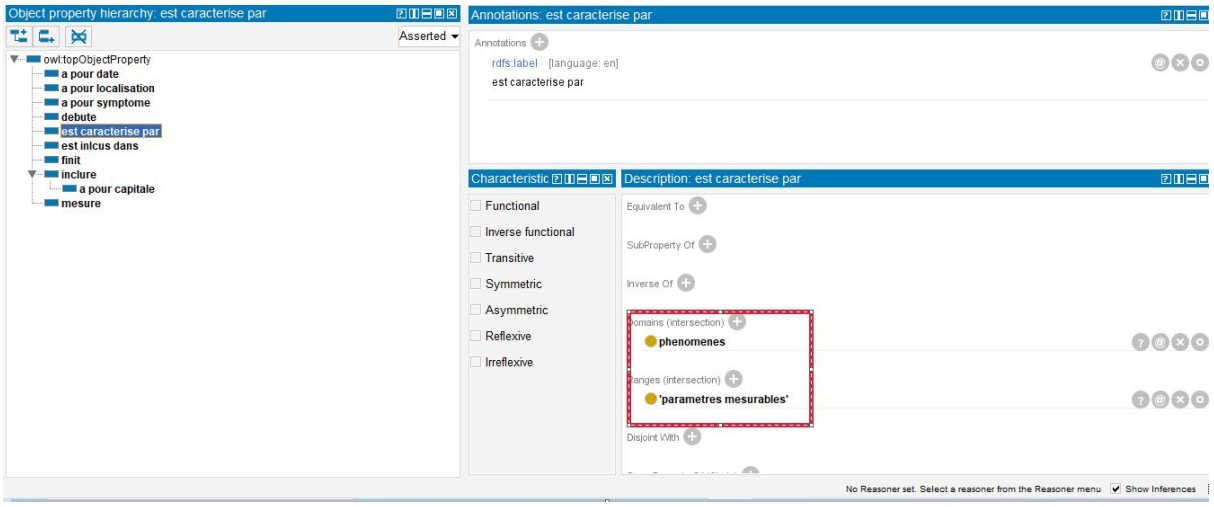
1. Un phénomène est caractérisé par des paramètres mesurables
2. Un phénomène a une durée en minutes
3. Un phénomène d´ebute à un instant
4. Un phénomène finit à un instant
5. Un instant a un timestamp, de type xsd :dateTimeStamp
6. Un phénomène a pour symptôme une observation
7. Une observation météo mesure un paramètre mesurable
8. Une observation météo a une valeur pour laquelle vous ne représentez pas l'unité
9. Une observation météo pour localisation un lieu.
10. Une observation météo a pour date un instant
11. Un lieu peut être inclus dans un autre lieu
12. Un lieu peut inclure un autre lieu
13. Un pays a pour capitale une ville

Nous créons les Object Properties et le Data Properties nécessaires pour représenter les relations entre les objets. Les captures suivantes le montre:





Pour lier les objets avec une relation nous utilisons les champs Domain et Range des propriétés.
Par exemple pour exprimer la relation 1 :



- Nous peuplons l'ontologie avec les faits suivants. Nous mettons en gras les déductions du raisonneur.

1. La température, l'hygrométrie, la pluviométrie, la pression atmosphérique, la vitesse du vent et la force du vent sont des paramètres mesurables (Attention, pas des types de paramètres, mais des instances de paramètres)
2. Le terme température est un synonyme anglais de température. Examinez les "Annotations" de l'individu.

3. La force du vent est similaire à la vitesse du vent

4. Toulouse est située en France. Remarquez que les individus dans cette phrase ne sont pas typés : créez Toulouse et France non pas comme une ville et un pays, mais comme des individus sans classe. Comment les classifie le raisonneur ?
Le raisonneur les classifie en tant que Ville et Pays
5. Toulouse est une ville
6. La France a pour capitale Paris. Ici aussi, Paris est un individu non typé
Classifié en tant une ville
7. Le 10/11/2015 à 10h00 est un instant que l'on appellera I1 (noté 2015- 11-10T10 :00 :00Z) 8. P1 est une observation qui a mesure la valeur 3 mm de pluviométrie à Toulouse à l'instant I1 (pas besoin de représenter l'unité)
9. A1 a pour symptôme P1
A1 est un phénomène

Ontologie lourde

- Description

1. Toute instance de ville ne peut pas être un pays
Nous utilisons la Description "Disjoint With"

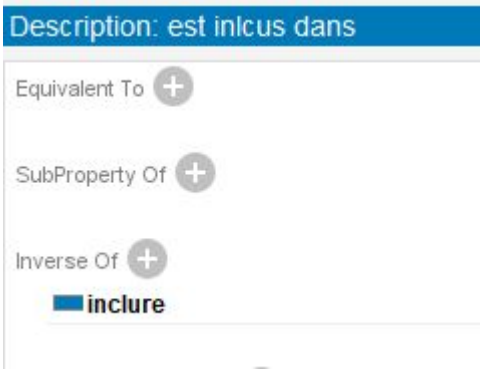
2. Un phénomène court est un phénomène dont la dur´ee est de moins de 15 minutes
— En syntaxe de Manchester : phénomène that 'a une dur´ee' some xsd:float [< 15]

3. Un phénomène long est un phénomène dont la dur´ee est au moins de 15 minutes
Pour le 2 et 3, nous créons les classes et complètons leur "Equivalent To" avec une expression. Pour la 2:

4. Un phénomène long ne peut pas être un phénomène court
Nous réutilisons "Disjoint With"

5. La propriété indiquant qu'un lieu est inclus dans un autre a pour propriété inverse la propriété indiquant qu'un lieu en inclue un autre.

Nous utilisons la Description des Propriétés Objets "Inverse Of"



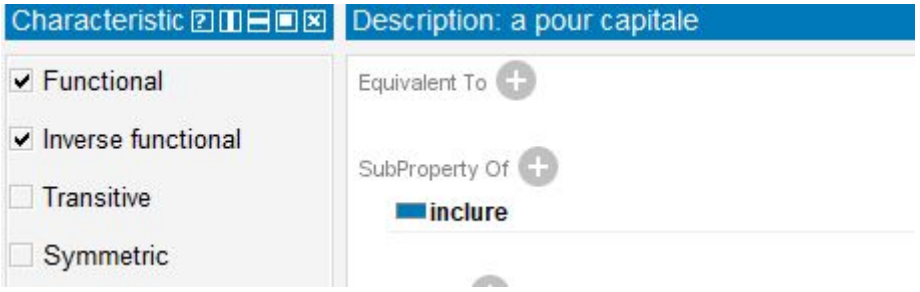
6. Si un lieu A est situé dans un lieu B et que ce lieu B est situé dans un

lieu C, alors le lieu A est situé dans le lieu C (utilisez les caractéristiques de la relation)

C'est la transitivité, nous la rajoutons pour cette propriété

7. A tout pays correspond une et une seule capitale (utilisez les caractéristiques` de la relation).

C'est la bijectivité. Nous la marquons pour la Propriété Objets ' a pour capitale'. Functional=Injective, Inverse Functional=Bijectivité



8. Si un pays a pour capitale une ville, alors ce pays contient cette ville (utilisez la notion de sous-propriété).

Nous mettons la propriété a ' a pour capitale ' en sous classe de 'inclure'

9. La Pluie est un phénomène ayant pour symptôme une Observation de Pluviométrie dont la valeur est supérieure à 0.

—phénomène that 'a pour sympt^ome' some (Observation that ('mesure' value Pluviom´etrie) and ('a pour valeur' some xsd:float [> 0]))

Nous rajoutons l'expression nécessaire dans la description de la classe Pluie

- Nous peuplons l'ontologie avec les faits suivants. Nous mettons en gras les déductions (inférences) du raisonneur.

1. La France est située en Europe
Paris et Toulouse sont aussi inclus dans l'Europe (et l'Europe inclus les deux premiers)
2. Paris est la capitale de la France
3. La Ville Lumière est la capitale de la France
Paris et la ville lumière sont les mêmes individus.
4. Singapour est une ville et un pays
Le raisonneur nous force à créer deux éléments différents.

TP2

Objectif

En utilisant une librairie java capable d'organiser des ontologies, développer et tester des méthodes permettant l'automatisation d'instanciation d'objets et de relations à partir de fichiers CSV.
On a à notre disposition des classes contenant :

- des classes d'objets et leur URI
- des méthodes d'instanciation d'objet d'une certaine classe et la mise en relation avec des propriétés
- des méthodes "parsant" du CSV
- les squelettes des méthodes à développer
- les tests pour les méthodes à développer

Sources

https://github.com/budbudmike/ISS_Semantics_Michael_Luan

Analyse

La classe “DoItYourselfModel”
Méthode *createPlace*

Nous utilisons la méthode *getEntityURI* pour récupérer l’URI de la classe Place.
Nous utilisons la méthode *createInstance* pour créer l'instance à partir de l’URI récupéré précédemment.

Méthode *createInstant*

Pour l'instanciation d’un Instant nous mettant le timestamp en tant que label de l’objet, mais nous attachant également à l’objet le timestamp avec une Data Property avec la propriété “a pour timestamp”.
Pour vérifier l’unicité de l’Instant nous avons donc le choix d’utiliser le label ou la relation de propriété, nous optons pour le label.
Pour instancier l’objet nous réutilisons les méthodes *getEntitURI* et *createInstance* mais aussi les méthodes *addDataPropertyToIndividual* et *getInstancesURI* (pour récupérer les instances d’Instant).

Méthode *getInstantURI*

Pour récupérer l’URI liée à un timestamp, nous parcourons les Instance de l’objet Instant et vérifions s' il correspondent au timestamp recherché.
Nous utilisons la méthode *hasDataPropertValue.*

Méthode *getInstantTimestamp*

A partir d’un URI, nous vérifions si c’est une instance d’Instant et retournons son timestamp.
Nous vérifions d’abord si l’URI correspond bien à une Instance (*getInstancesUri*), et si elle est présente, récupérons ses propriétés *(listProperties)* et retournons la data liée à la propriété “a pour timestamp”.

Méthode *createObs*

Nous donnons à l’objet un Label étant la concaténation de l’URI de la mesure physique et l’URI du timestamp.
Nous lui attachons une date avec (*addObjectPropertyToIndividual)* et une valeur avec (*addDataPropertyToIndividual*).
Finalement nous relions cette observation au sensor en utilisant les méthodes *addObservationToSensor* et *whichSensorDidIt .*

Les méthodes passent tous les tests.

La classe “DoItYourselfControl”

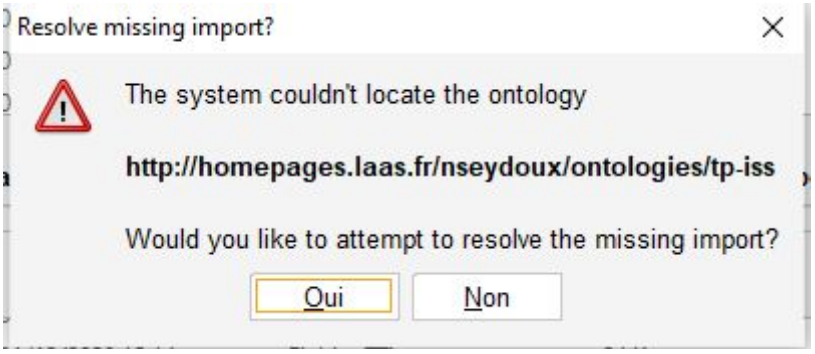
Méthode *instantiateObservations*

Nous utilisons la méthode implémentée *createObs* pour instancier les objets dans le fichier CSV.

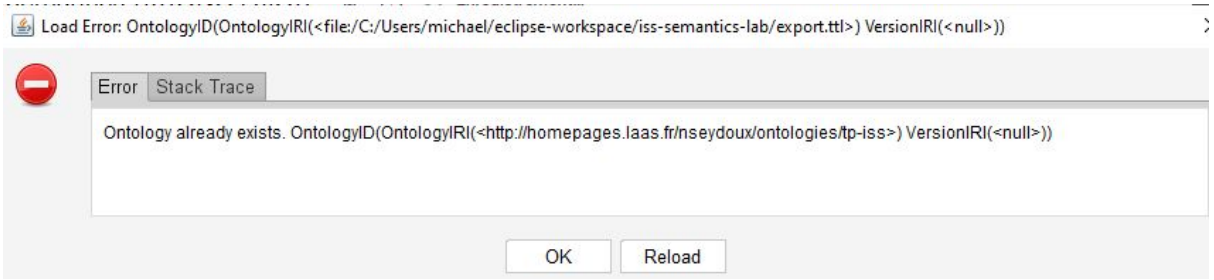
Test : La classe Controller
Nous nous aidons des méthodes *parseObservations* qui parcourent les fichiers CSV et nous retourne une liste d'objets de la classe "ObservationEntity" . Cette dernière contient les méthodes qui nous permettent d' extraire les infos nécessaires pour instancier une Observation dans notre ontologie.

Le programme prend beaucoup de temps à s'exécuter, des optimisations dont surement à faire dans les méthodes implémentées. (Nous avons opté pour le fonctionnel et pas assez pour l'optimal).

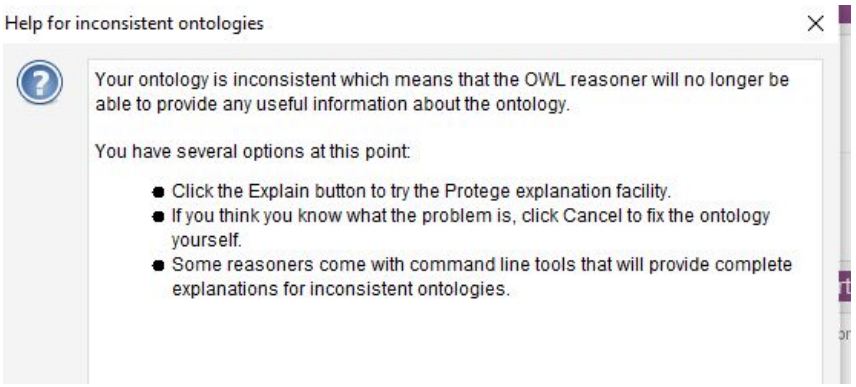
On obtient bien l'export export.ttl
Au moment de l'importation dans Protégé, il nous indique qu'il ne trouve pas le fichier tp-iss.ttl.



Lorsque nous lui indiquons le chemin il nous retourne ce message d'erreur :



Nous ignorons donc le warning et lui disons de ne pas prendre en compte ce fichier.
Lorsque on l'importe dans Protégé nous avons bien toutes les instanciations des timestamp et des Observation, ainsi que les relations entre elles.
Mais nous n'avons pas le lien avec les sensors.
Lorsque nous lançons le raisonneur, on obtient le message suivant :



Nous comprenons que les timestamps doivent être liés à des sensors, mais nous ne retrouvons pas cela dans Protégé, malgré le fait que nos méthodes avaient passé les tests associés.