Berta Pauline
Conceicao Nunes Joao
07 Décembre 2020

TP ISS Semantic Data

1-Introduction

- Manipuler concrètement la notion d'ontologie.
- Voir les aspects principaux.
- Montrer les déductions que peut faire un raisonneur à différentes étapes du développement de l'ontologie.

Le contexte est le suivant : on veut développer une application de météorologie intelligente. Pour ce faire, on va décrire des stations météos pour rendre les données qui en sont issues le plus riches possibles.

2-Création de l'ontologie

2.1 L'ontologie légère

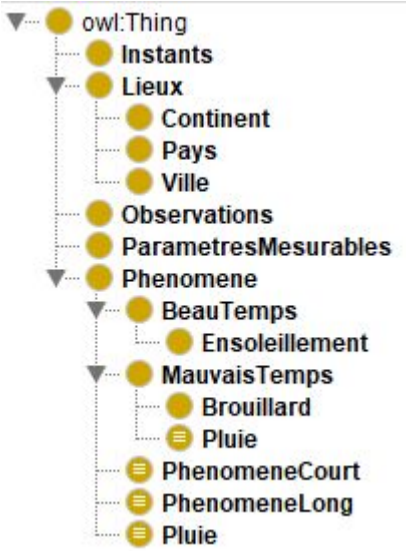
2.1.1 Conception

L'objectif de cette ontologie est de fournir un vocabulaire pour décrire sémantiquement des données issues du domaine de la météorologie. Plus précisément, cette ontologie devra représenter la connaissance en lien avec les phénomènes météorologiques (pluie, tempête, ...), les paramètres mesurables qui les caractérisent (température, humidité, ...), mais aussi les capteurs qui permettent de faire ces observations.

Etape N°1:

Création des différentes classes :

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 phénomène 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



Pour cette première étape, nous avons créé sur le logiciel Protégé différentes classes, regroupant des sous-classe, comme par exemple la classe Phénomène regroupant les sous-classes Beau Temps et Mauvais Temps (phrase n°1).

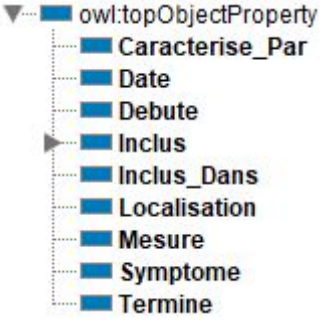
Etape N°2:

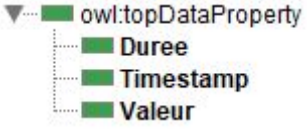
Ajout des différentes propriétés :

1. Un phénomène est caractérisé par des paramètres mesurables
2. Un phénomène a une durée en minutes
3. Un phénomène débute 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 représenterez pas l'unité
9. Une observation météo a 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 avons mis en place des "Data Properties" comme Durée en minutes (phrase n°2), qui vont être rattachées à différentes classes. De ce fait, un phénomène, qui est une classe, va être défini par une durée en minute.

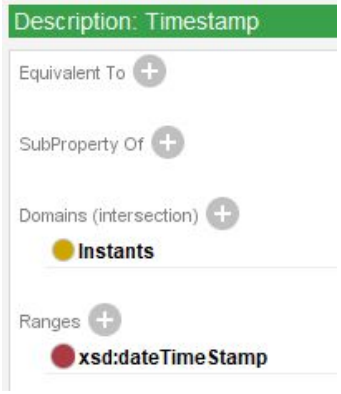
Cette classe est aussi mise en lien avec des "Objects Properties" comme l'action Débute et l'action Termine représentant un phénomène qui débute et se termine à un instant défini (une autre classe qui elle aussi à une durée en minutes).

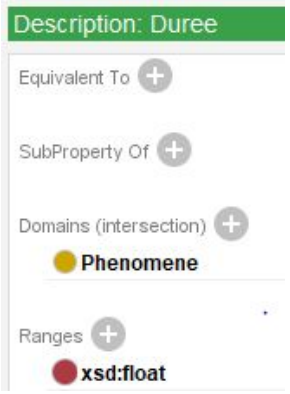


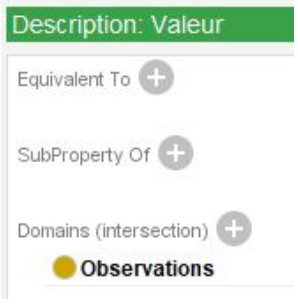


La classe Instant est définie par un Timestamp (phrase n°5).

Les "Data Properties" sont donc configurés de la façon suivante.







Les "Objects Properties" sont donc configurés de la façon suivante.

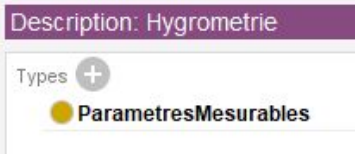
2.1.2 Peuplement

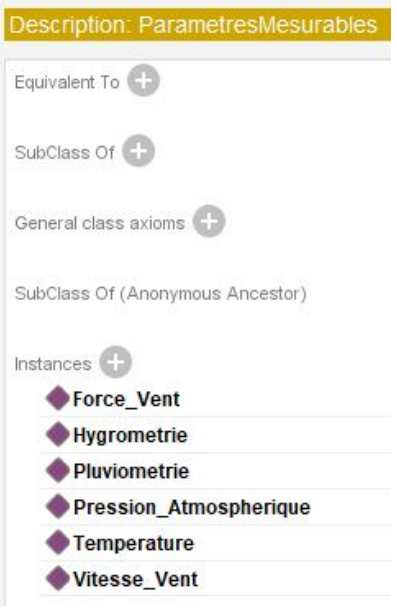
Suite à notre précédente configuration, on constate que le raisonneur déduit à quel classe appartient une annotation en s'appuyant sur sa description.

En utilisant les domain/range d'une data property, le raisonneur peut établir un lien avec la classe regroupant ce type de "Data Properties" ainsi que de déterminer l'unité de celle-ci , alors que pour une "Object Property", le raisonneur établit directement un lien entre deux classes traduit par une propriété reliant ces deux classes.

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 temperature 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 ?
5. Toulouse est une ville
6. La France a pour capitale Paris. Ici aussi, Paris est un individu non typé
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

Dans cette partie, nous créons des variables "Individuals", prenons l'exemple de la configuration de l'Hygrométrie qui est un paramètre mesurable (phrase n°1), il en est de même pour tous les autres paramètres.



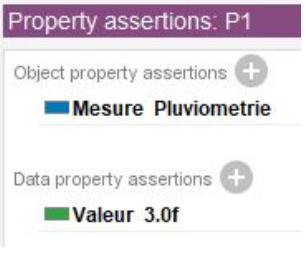


D'autre part, nous avons fait en sorte que le terme Température soit compris aussi bien en français qu'en anglais (phrase n°2).

Dans la même catégorie nous avons déclaré similaire la force et la vitesse de vent (phrase n°3).

A ce stade, si nous créons Toulouse et Paris comme des individus (phrase n°4), le raisonneur les classifie comme des lieux mais reste incapable de les classer dans les sous-classes Ville ou Pays.

Création des annotations I1 et P1 (phrase n°7 & n°8).



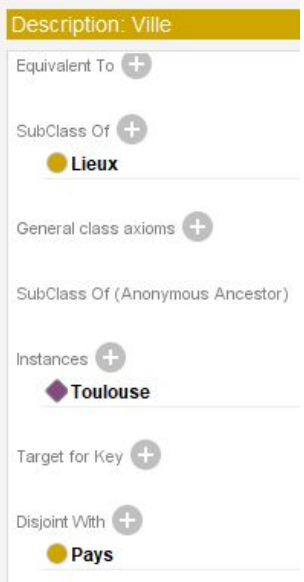
2.2 L'ontologie lourde

2.2.1 Conception

1. Toute instance de ville ne peut pas être un pays

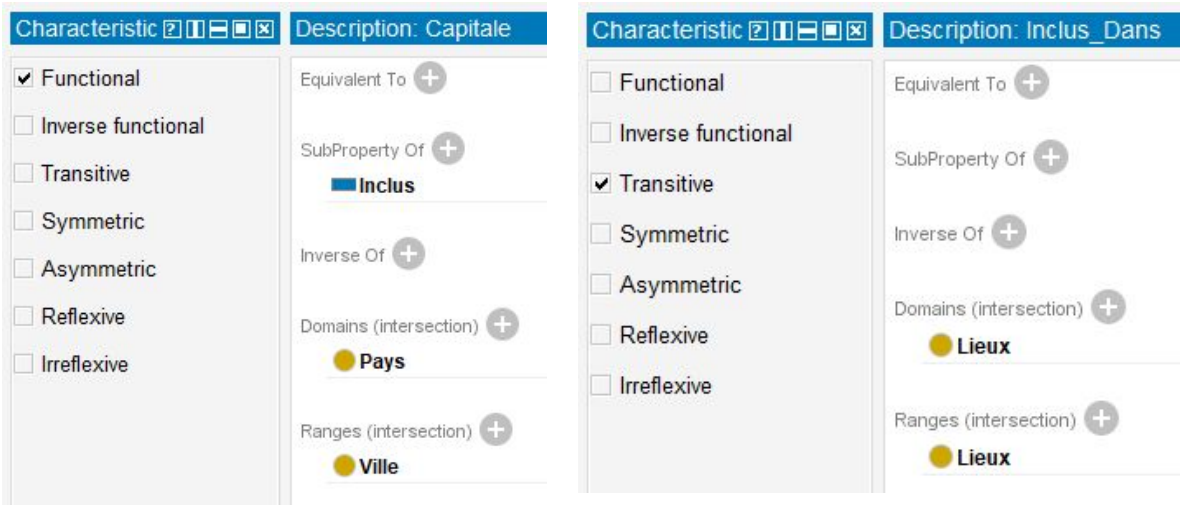
2. Un phénomène court est un phénomène dont la durée est de moins de 15 minutes
   — En syntaxe de Manchester : `Phénomène that 'a une durée' some xsd:float [< 15]`
3. Un phénomène long est un phénomène dont la durée est au moins de 15 minutes
4. Un phénomène long ne peut pas être un phénomène court
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.
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)
7. À tout pays correspond une et une seule capitale (utilisez les caractéristiques de la relation).
8. Si un pays a pour capitale une ville, alors ce pays contient cette ville (utilisez la notion de sous-propriété).
9. La Pluie est un Phénomène ayant pour symptôme une Observation de Pluviométrie dont la valeur est supérieure à 0.
   — `Phénomène that 'a pour symptôme' some (Observation that ('mesure' value Pluviométrie) and ('a pour valeur' some xsd:float [> 0]))`

Mise à jour des classes au cours de cette étape :



On modélise des classes définies (Phénomène Cours et Long), étant des sous-classes de Phénomène, qui ont pour particularité de ne pas seulement être définies par une durée mais qui sont classées en fonction de la durée qui leur est attribuée. La syntaxe de Manchester nous permet d'inscrire dans le description de ces classes ces intervalles d'appartenance.

Mise à jour des "Objects Properties" au cours de cette étape :



Les axiomes logiques peuvent être définies au niveau des caractéristiques des "Objects Properties".

Pour Capitale, nous voulons définir le fait qu'un pays ne peut avoir qu'une seule capitale, c'est pour ça que nous utilisons la fonctionnalité Functional. Lorsqu'une Capitale sera attribuée à un Pays, aucune autre ne pourra venir s'y ajouter (phrase n°7).
Dans le cas de Inclus_Dans nous avons fait en sorte d'établir une relation transitive en lui appliquant la caractéristique Transitive.

Un lieu inclus dans un lieu sera vrai dans les deux sens mais une Ville inclus dans un Pays sera automatiquement inclus dans le Continent qui inclus le Pays (phrase n°5 & n°6).

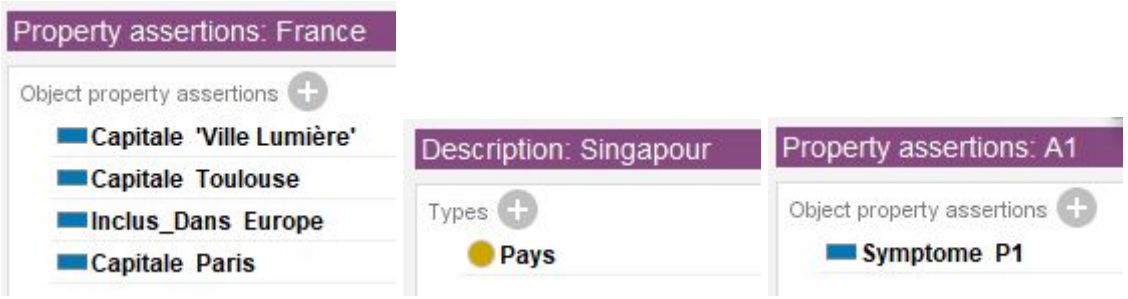
La classe Pluie, sous-classe de Phénomène, est créée (phrase n°9).

2.2.2 Peuplement

1. La France est située en Europe
2. Paris est la capitale de la France
3. La Ville Lumière est la capitale de la France
4. Singapour est une ville et un pays

1. Que sait le raisonneur de A1 ?
2. Que sait le raisonneur sur Paris ?
3. Constatez la réaction du raisonneur si Toulouse est déclarée comme la capitale de la France

Mise à jour des "Individuals" au cours de cette étape :



Le raisonneur classe bien A1 comme un Symptome de P1. De plus, il sait que Paris est la capitale de la France.

Quand on définit à la fois Paris et La Ville Lumière comme capitales de la France, le raisonneur sort une erreur puisqu'il ne peut y avoir qu'une seule capitale pour un pays donné, malgré tout nous retrouvons les différentes capitales attribuées à la France dans ses assertions.

3-Exploitation de l'ontologie

En ce qui concerne le TP 2, nous avons eu quelques problèmes lors de la compilation du projet sur Eclipse. Cela nous a pris un certain temps, mais nous avons réussi à implémenter l'intégralité des fonctions demandées dans IModelFunction.

Il s'agit des méthodes de peuplement suivantes :

1- CreateInstance:

```
String result;
result = this.model.createInstance(name,this.model.getEntityURI("Lieu").get(0));

    return result;
}
```

2 - CreateInstant:

```
public String createInstant(TimestampEntity instant) {
    String result;
    String timestamppropertyURI = this.model.getEntityURI("a pour timestamp").get(0);

    result = this.model.createInstance(instant.getTimeStamp(), this.model.getEntityURI("Instant").get(0));
    this.model.addDataPropertyToIndividual(result, timestamppropertyURI, instant.getTimeStamp());
    return result;
}
```

3 - GetInstantURI:

```
public String getInstantURI(TimestampEntity instant) {
    String result = null;
    List<String> list_instants;
    String timestamppropertyURI = this.model.getEntityURI("a pour timestamp").get(0);

    list_instants = this.model.getInstancesURI(this.model.getEntityURI("Instant").get(0));

    for(int i = 0; i < list_instants.size(); i++) {
        if (this.model.hasDataPropertyValue(list_instants.get(i),timestamppropertyURI , instant.getTimeStamp())) {
            return list_instants.get(i);
        }
    }


    return result;
}
```

4 - GetInstantTimestamp:

```
public String getInstantTimestamp(String instantURI)
{
    String result = null;
    List<String> list_labels = this.model.listLabels(instantURI);
    return list_labels.get(0);
}
```

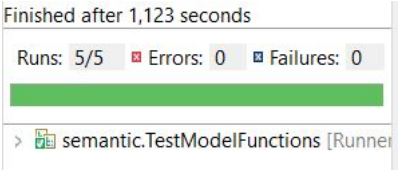
5 - CreateObs:

```
public String createObs(String value, String paramURI, String instantURI) {
    String ObsURI = this.model.getEntityURI("Observation").get(0);
    String timestp = this.getInstantTimestamp(instantURI);

    String instanceObsURI = this.model.createInstance(timestp, ObsURI);
    this.model.addDataPropertyToIndividual(instanceObsURI, this.model.getEntityURI("a pour valeur").get(0), value);
    this.model.addObjectPropertyToIndividual(instanceObsURI, this.model.getEntityURI("a pour date").get(0),instantURI);
    this.model.addObservationToSensor(instanceObsURI, this.model.whichSensorDidIt(timestp, paramURI));

    return instanceObsURI;
}
```

Après l'implémentation de ces méthodes de peuplement nous avons vérifié que leur comportement est celui attendu avec le test fourni :



Pour la suite des fonctions avec notamment la méthode instantiateObservations nous n'avons pas eu assez de temps pour finir.

Nous pouvons cependant répondre à la dernière question, " Quelle est la différence entre un objectProperty et une dataProperty?"

La différence entre les deux est que l'objectProperty permet de créer une relation qualitative entre deux types. Alors que la dataProperty permet de rajouter une relation quantitative, par l'ajout d'une donnée à une classe. Cette data ou "variable" lui est ajouté comme un attribut de classe finalement.