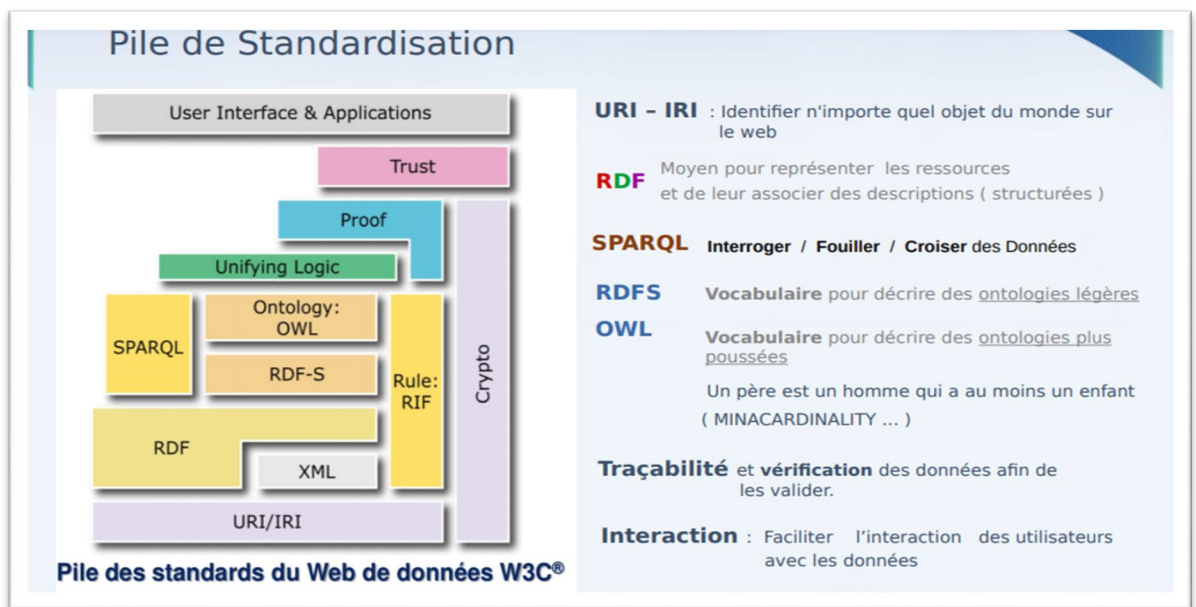


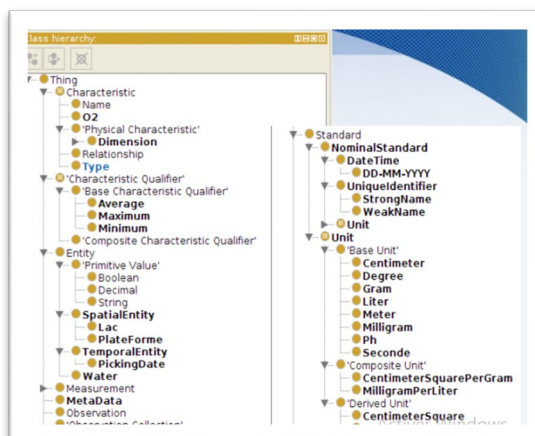
Rapport

TRAITEMENT DES DONNÉES SÉMANTIQUES

1-Introduction



Ontologie ? Une Ontologie est un réseau sémantique regroupant un ensemble de concepts décrivant un domaine. Ces concepts sont liés les uns aux autres par des relations hiérarchiques d'une part, et sémantiques d'autre part.



Ce TP a pour objet de vous faire manipuler concrètement la notion d'ontologie, de vous en faire voir les aspects principaux, et de vous 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 d'écrire des stations météorologique pour rendre les données qui en sont issues le plus riches possibles.

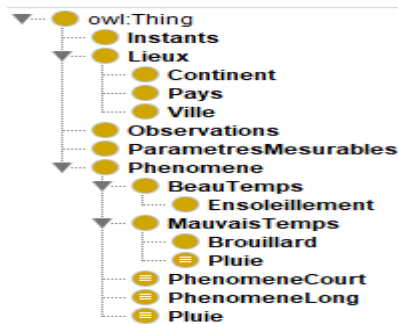
2-Création de l'ontologie

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.

En effet, 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 collectent 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, j'ai créé sur le logiciel Protégé différentes classes, regroupant des sous-classe, exemple la classe Phénomène regroupant les sous-classes Beau Temps et Mauvais Temps (Phrase 1).

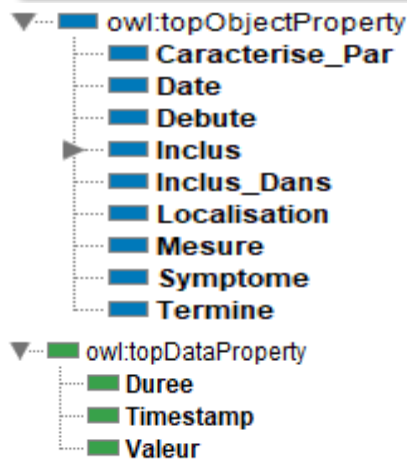
Etape N°2:

Ajout des différentes propriétés :

J'ai mis en place des "Data Properties" comme Durée en minutes (Phrase 2), qui vont être rattachées à différentes classes. De ce fait, un phénomène, qui est une classe, va être défini par une durée en min

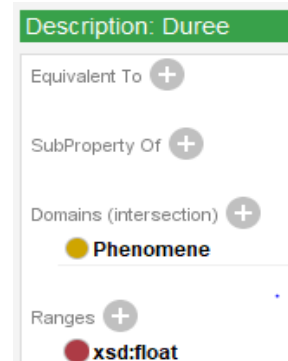
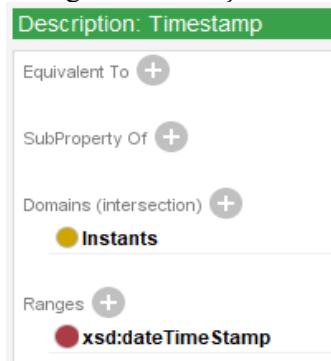
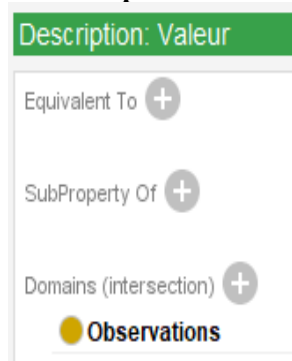
8. Une observation météo a une valeur pour laquelle vous ne représenterez pas l'unité
9. Une observation météo a pour localisation un lieu.
10. Une observation météo a pour date un instant
11. Un lieu peut être inclus dans un autre lieu
12. Un lieu peut inclure un autre lieu
13. Un pays a pour capitale une ville

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

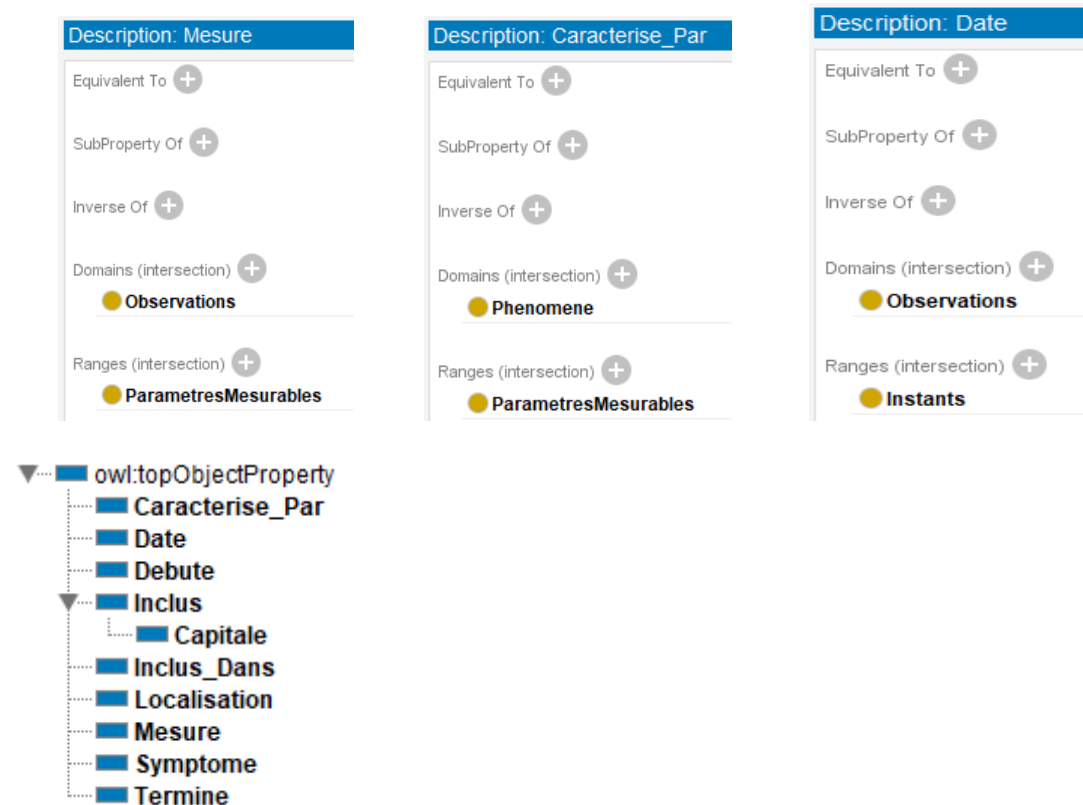


La classe Instant : définie par un **Timestamp**

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



Configuration "Objects Properties" :



2.1.2 Peuplement

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 ,

- "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 classe 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

- ◆ A1
- ◆ Europe
- ◆ Force_Vent
- ◆ France
- ◆ Hygrometrie
- ◆ I1
- ◆ P1
- ◆ Paris
- ◆ Pluviometrie
- ◆ Pression_Atmospherique
- ◆ Singapour
- ◆ Temperature
- ◆ Toulouse
- ◆ Ville Lumière
- ◆ Vitesse_Vent

Dans cette partie, nous créons des variables "Individuals", exemple

Description: ParametresMesurables

Equivalent To +

SubClass Of +

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

- ◆ Force_Vent
- ◆ Hygrometrie
- ◆ Pluviometrie
- ◆ Pression_Atmospherique
- ◆ Temperature
- ◆ Vitesse_Vent

configuration de l'Hygrométrie qui est un paramètre mesurable (Phrase 1)

Description: Hygrometrie

Types +

- ParametresMesurables

[rdfs:label](#) [language: fr]


Temperature


[rdfs:seeAlso](#)


Temperature


Ainsi ,le terme Température soit compris aussi bien en français qu'en anglais (Phrase 2).

Description: Force_Vent

Types 

-  ParametresMesurables


Same Individual As 


-  Vitesse_Vent


Même pour la force et la vitesse de vent (Phrase 3).

Création des annotations I1 et P1 (Phrase 7 & 8).

Property assertions: I1

Object property assertions 

Data property assertions 

 **Timestamp**
"2015-11-10T10:00:00Z"^^xsd:dateTimeStamp

2.2 L'ontologie lourde

Une ontologie lourde est en fait une ontologie légère possédant des axiomes logiques. Les axiomes logiques nous permettent d'enrichir la logique de notre ontologie. Voici comment nous avons enrichi notre ontologie: Toute instance de ville ne peut pas être un pays. Pour faire cela, j'ai dû définir l'axiome qui dissocie des 'lieux' différents. Dans notre cas, 'ville', 'pays' et 'continent' sont tous dissociés. Par exemple, Paris ne peut pas être une ville.

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]))`

Description: Ville

Equivalent To +

SubClass Of +

● Lieux

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

◆ Toulouse

Target for Key +

Disjoint With +

● Pays

Mise à jour des classes au cours de cette étape :

Description: PhenomeneCourt

Equivalent To +

● Phenomene and (Duree some xsd:float[< 15.0f])

SubClass Of +

● Phenomene

Description: PhenomeneLong

Equivalent To +

● Phenomene and (Duree some xsd:float[>= 15.0f])

SubClass Of +

● Phenomene

Modélisation les 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.

Le phénomène court est un phénomène dont la durée est de moins de 15 minutes et n phénomène long est un phénomène dont la durée est au moins de 15 minutes.



Pour faire cela il me faut définir des expressions logiques, dans un langage spécifique, comme dans les images précédentes pour exprimer les axiomes voulus.

J'ai du dissocier les deux car un phénomène long ne peut pas être un phénomène court.

La propriété indiquant qu'un phénomène est caractérisé par des paramètres a pour propriété inverse la propriété 'e' représentant que des paramètres caractérisent un phénomène.

Pour cela, J'ai rempli la partie 'Inverse Of' avec 'caractérise'.

Characteristics: 'car'

Description: 'caractérisé par'

☐ Functional
☐ Inverse functional
☐ Transitive
☐ Symmetric
☐ Asymmetric
☐ Reflexive
☐ Irreflexive

Equivalent To +

SubProperty Of +

Inverse Of +

Domains (intersection) +

Ranges (intersection) +

Disjoint With +

SuperProperty Of (Chain) +

caractérise

phénomènes

paramètres

3-Exploitation de l'ontologie

A cause des Problèmes PC je n'ai pas pu terminer.
j'ai pu comprendre les méthodes de base :

Methode `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.getInstanceURI(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;
}
```

Methode `getInstantTimestamp`

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

Methode 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;
}
```

Methode createObs

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

    String instanceObsURI = this.model.createInstance(timestamp, 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(timestamp, paramURI));

    return instanceObsURI;
}
```

- Quelle est la différence entre la propriété d'objet et la propriété de données?
 - ✓ **propriété d'objet** permet de lier un individu (instance d'une classe) à un autre en définissant un lien entre les deux, de type "est inclus dans", "mesure", "a pour capitale", Dans un graphe, ce sera une boîte reliée aux deux individus.
 - ✓ **propriété de données** permet de lier un individu à une valeur (donnée) en spécifiant au besoin son unité, par exemple "**a pour valeur**" reliée à un float ou "**a pour timestamp**" reliée à un timestamp.