Semantic Web

Lab 1 - Creation of an ontology

In this first lab, available at https://zwifi.eu/public/docs/tp1.pdf, we create an ontology to learn to use Protégé, an ontology creation tool. We created a meteorology ontology to test both the ontology creation functionality and the reasonner, a tool that uses the different properties of objects to deduce additional information.

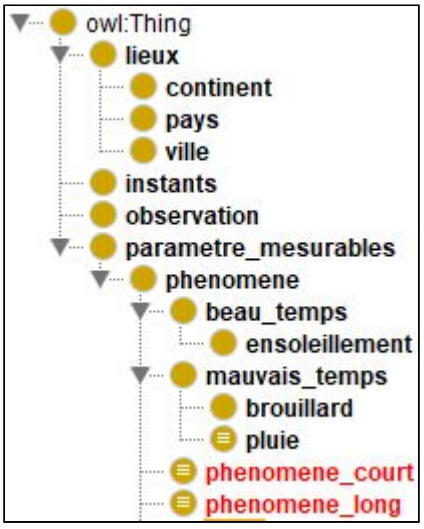


fig1. Ontology Classes

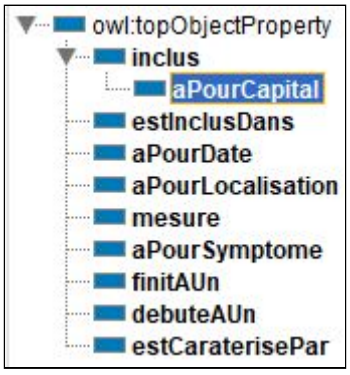


fig2. Object Properties

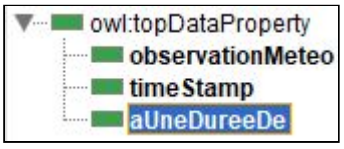


fig3. Data Properties

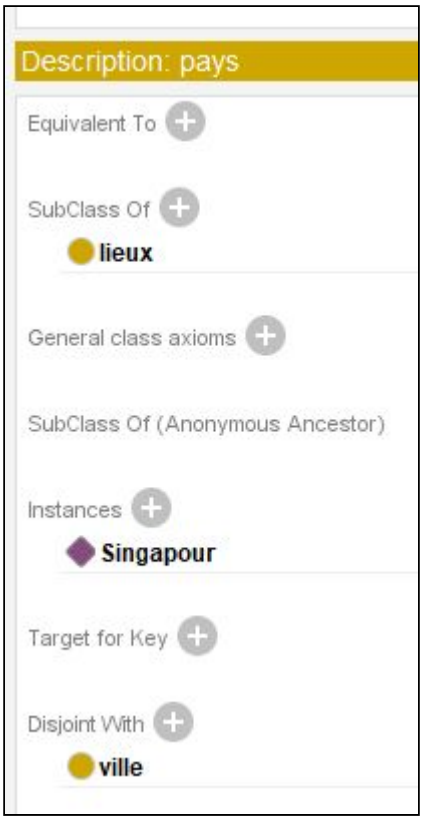
fig4. Entities

We initially constructed a “light” ontology, with simple relationships. For instance, “rain” is a sublass of “bad weather” which is itself a subclass of “phenomena”. By defining relationships between objects (by specifying their domain and range), we can also use the reasoner to deduce additional information: We can create “Toulouse” and “France” as typeless objects, but after specifying that “France” **contains** “Toulouse”, the reasoner automatically specifies

that both objects are “Places”, since the relationship **contains** has “Places” as both its domain and range.
We then created a “heavy” ontology, by adding more specific relationships and properties to objects using the manchester syntax (available at http://webont.org/owled/2008dc/papers/owled2008dc_paper_11.pdf)
As we defined that a country has one unique capital, and indicated that both “Paris” and “La ville lumière” are the capital of “France”, the reasonner has found that “La ville lumière” is equivalent to “Paris” :

If Toulouse is also declared as the capital of France, the reasoner will detect that it’s the same individual as Paris and “La ville lumière”.

We also specified that a country cannot also be a city:



This property seems elementary, but after defining Singapore as both a city and a country, the reasoner detects an issue and flags the object.

The reasonner was supposed to find that A1 is a rain phenomenon, since we defined rain as: “a phenomenon that has as a symptom an observation of pluviometry of positive value”. However, our observation has its value stored as an integer, and the rain property is looking for a float value, so A1 is not automatically found to be rain. It’s important to check these seemingly unimportant characteristics to be sure the ontology works as intended.

Lab 2 - Semantic Web of Things

```java
@Override
public String createPlace(String name) {
    return this.model.createInstance(name, this.model.getEntityURI("Place").get(0));
}
```

```java
@Override
public String createInstant(TimestampEntity instant) {
    String uri = this.model.createInstance(instant.getTimeStamp(), this.model.getEntityURI("Instant").get(0));
    this.model.addDataPropertyToIndividual(uri, this.model.getEntityURI("has timestamp").get(0), instant.getTimeStamp());
    return uri;
}
```