Lab report - Design of Ontologies and data enrichment for the Semantic Web

Josué ALVAREZ

January 2017

1 Introduction

The lab is divided in two parts: the first part is about creating an ontology to describe data from the field of meteorology. The second part is about retrieving and enriching real world data and creating an application that makes uses of the rich data provided by the ontology.

2 Ontology creation

In this section, the ontology created during the lab will be briefly presented, and the concepts used will be explained.

2.1 Manipulating classes and instances

Classes are way of classifying real world objects and concepts. Relations can be expressed between classes through the class hierarchy or the usage of object properties.

Instances of classes are real world objects that share the properties of their class.

Subclasses Subclasses can be used when a specialization or generalization needs to be expressed. For instance, in our ontology, we use subclasses to describe the fact that "Continent", "Town", "Country" are different kind of "Place"s.

Class equivalence The EquivalentTo relation can be used to express restrictions under which a given instance is an instance of a specific class. In our ontology, we used this to express the fact that "Rain" is equivalent to a phenomenon which has as a symptom non-null pluviometry observations. Figure 1. shows how it is expressed in Protege with the Manchester syntax.

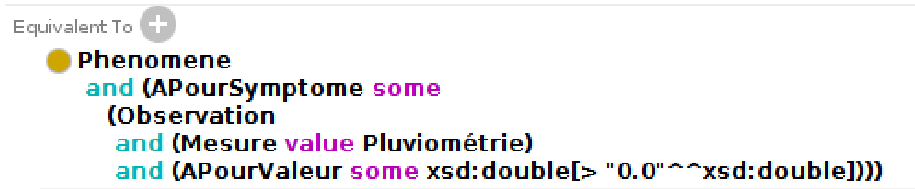

Figure 1. "Rain" phenomenon EquivalentTo relation.

Disjonction The Disjoint With relation can be used to inform the reasoner that instances of a class cannot also be instances of another class. We use this to express the fact that "Long Phenomenon" cannot be a "Short Phenomenon".

2.2 Meteorology ontology classes

Figure 2. shows the classes created during the lab.

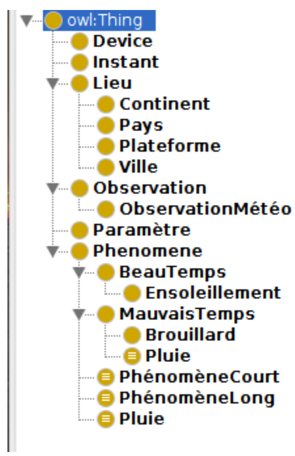

Figure 2. Classes created during the lab.

2.3 Manipulating object properties

Object properties are used to express custom relationships between two classes. Object properties have a domain and a range, which are sets of classes.

Here is an example scenario of usage of object property that we encoutered during the lab : "is located in" is an object property which has a "Place"as domain and another "Place" as range. It expresses the fact that the first "Place" instance is located in the second "Place" instance. For instance, we can say that the place "Toulouse" is located in the place "France".

Object properties characteristics Object properties have characteristics such as reflexivity and transitivity. We used transitivity in our ontology to express the fact that if a place "A" is included in a place "B", and the place "B" is included in the place "C", then the place "A" is included in the place "C".

Object properties description Object properties have a description which contains relations between object properties. For instance, we can say that the object property "is characterized by" is the inverse of "caracterizes". Or that "is the capital of" is a subproperty of "is located in". Thanks to this description, the reasoner can make additional deductions thanks to a single object property. In the example in Figure 3, we express the fact that the capital must be unique. The consequence of this fact is that, if a country has two "has capital" object properties on two instances of towns, then these instances refer to the same real word town.

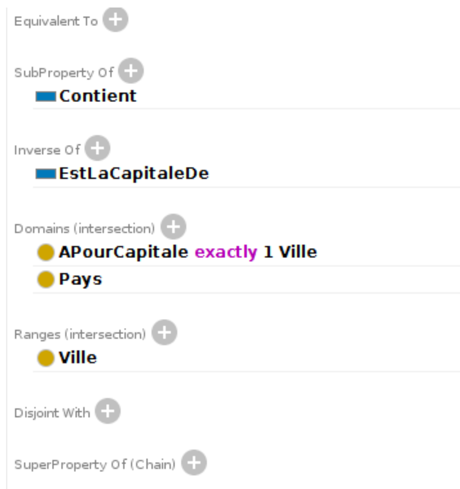

Figure 3. Description of example object property "has capital".

2.4 Manipulating data properties

While object properties define relations between classes (and therefore instances of objects), data properties define relations between a class and data values, which are not instances of objects. As a consequence : the domain of data properties is a class, and the range is a data value (float, integer, date, and so on).

Data property example In our ontology, we used the "has timestamp" property to express the fact that to an "Instant" is associated a dateTimeStamp.

2.5 Reasoner deductions

The reasoner uses the facts in the ontology to generate new knowledge. Below are examples of deductions that the reasoner made on the ontology created during the lab.

2.5.1 Light ontology

Example 1 . Given the following facts :

1. Toulouse is a city.
2. The capital of France is Paris.
3. The 11/10/2015 10h00 is an instant that we shall call I1.
4. P1 is an observation which measures the value 3 mm of pluviometry in Toulouse at I1.
5. A1 has symptom P1.

The reasoner can deduce that :

1. Paris is a Town. Because we told it that 'France has capital Paris', and that the range of the object property 'has capital' is 'Town', the reasoner is able to make this deduction, as shown in figure 4.
2. France is a country. This time, the domain of the 'France has capital Paris' fact indicates to the reasoner that France is a country, as shown in figure 5.

Figure 4. Explaination given for the deduction "Paris Type Ville".

Figure 5. Explaination given for the deduction "France Type Pays".

2.5.2 Heavy ontology

The heavy ontology presented in the appendix expresses more complex relationships between objects. The following section will list a few interesting deductions made by the reasoner on this ontology.

1. A1 is a short phenomenon. The reasoner knows that a short phenomenon is a "Phenomenon" which has the data property "has duration", and whose value is greater than 15. The reasoner also knows that A1 has symptom P1. Because the domain of "has symptom" is "Phenomenon", it deduces that A1 is a Phenomenon. And because the value of A1 "has duration" is actually greater than 15, the reasoner can finally deduce that A1 is a short phenomenon, as shown in figure 6.

```
Explanation for: A1 Type PhénomèneCourt
1) A1 APourSymptome P1
2) A1 APourDurée "30.0"^^xsd:double
3) APourSymptome Domain Phenomene
4) PhénomèneCourt EquivalentTo Phenomene and (APourDurée some xsd:double[> "15.0"^^xsd:double])
```

Figure 6. Explaination given for the deduction "A1 is a ShortPhenomenon".

2. A1 is a rain phenomenon. The deduction is similar to the previous one, the reasoner makes use of the Range and Domain of the relation "has symptom" to deduce the class of A1 and P1, then exploits the "EquivalentTo" relation in order to make its final deduction.

```
Explanation for: A1 Type Pluie
1) A1 APourSymptome P1
2) APourSymptome Range Observation
3) P1 Mesure Pluviométrie
4) P1 APourValeur "4949.0"^^xsd:double
5) APourSymptome Domain Phenomene
6) Pluie EquivalentTo Phenomene and (APourSymptome some  (Observation and (Mesure value Pluviométrie) and (APourValeur some xsd:double[> "0.0"^^xsd:double])))
```

Figure 7. Explaination given for the deduction "A1 Type Rain".

3. Paris is the same as "The city of light". In other words, "The city of light" is another name of Paris. Because we told it that "Paris is the capital of France" and "The city of light is the capital of France", and that "is the capital of" links a unique town to a country, the reasoner deduces that "Paris" and "The city of light" are actually the same individuals. However, if we told the reasoner that "Paris" and "The city of light" were two distinct entities, it would report an inconsistancy, such as seen in figure 8.

```
Explanation for: owl:Thing SubClassOf owl:Nothing
    France APourCapitale Toulouse
    APourCapitale Domain APourCapitale exactly 1 Ville
    France APourCapitale Paris
    Toulouse DifferentFrom Paris
    APourCapitale Range Ville
```

Figure 8. Example of inconsistency detected by the reasoner.

4. Paris is situated in "France". This is a good example of usage of subproperties. Because the capital of France is Paris, and that "has capital" is a sub-property of "contains", and "contains" is the inverse of "is situated in", the reasoner deduces that Paris is situated in France.

3 Ontology enrichment

In the second part of the lab, we retrieved raw data and processed them to make them 5-star data, that we injected in the ontology we previously created. The process of data enrichment is composed of ... major steps :

1. Retrieve the data from an open data platform, or any other data source.
2. Parse the data in a specific format (CSV, RDF) and convert it to any convenient representation.
3. Use an API such as OWL API (we used a custom API in this lab) to load the ontology, and add the facts previously retrieved to the ontology.
4. Export the newly enriched ontology.

The code used in the lab to perform the ontology enrichment is available as an appendix to this document.

4 Conclusion

During this lab, we saw how to create an ontology describing whichever topic we want, how to populate it with data coming from various sources, and how the reasoner makes use of its knowledge to deduce new facts. The next step would be to learn how to use all of this to build applications that could query such a knowledge base, which is an exciting part of the using Ontologies !