

# OWLuedo : building a murder ontology using Protégé

In the previous tutorial we looked at graph querying to solve a Cluedo-like puzzle. In this tutorial, we'll be concentrating on using OWL <sup>1</sup> to create an ontology providing reasoning capabilities to identify the murderer of a new puzzle. You'll use the Protégé<sup>2</sup> software to do this.

## 1 Draft ontology

### 1.1 Creating classes

The ontology used in the previous tutorial (SPARQLuedo) is extremely lightweight, both in its taxonomy and in the reasoning properties described. The aim is to extend this ontology by modifying `Owluedo.owl`. You need to express the following knowledge as a class hierarchy:

- An object can be mobile or immobile
- There are five main types of moving object: blunt, sharp, soft, firearm and product.
- Sharp objects, blunt objects and firearms are weapons, which are themselves considered mobile objects.
- People can be victims, murderers or innocents.
- A house and a room are kinds of places
- Floor is a third kind of place
- A product is either a liquid or powder.
- There are several causes of death, divided into injury, drowning and poisoning.
- An injury can be caused by a bullet, a sharp object or a blunt object, defining as many injury subclasses.

---

<sup>1</sup><https://www.w3.org/TR/owl-ref/>

<sup>2</sup><http://protege.stanford.edu/>

- A murderer can be described alternatively as “killer” or “assassin”.
- In French, murderer is “meurtrier”.
- A room can contain zones, which are places.

## 1.2 Creating properties

Add properties (and possibly additional concepts) to express the following relationships. Some properties will be `owl:ObjectProperty`, others `owl:DatatypeProperty`, depending on whether the property links two resources (class instances) or a resource to a literal value (e.g. an integer).

- A place can be located in another place.
- The `roomInHouse` property is a special case of a place being located in another place.
- In particular, a room can be located on a floor, and a floor is located in a house.
- An area is organized around a stationary object.
- A victim has a cause of death.
- A cause of death is attributable to a moving object.
- An object can be or have been in someone’s possession.
- In particular, it may still be in his or her possession.
- A victim has been killed by a murderer.
- A person may be or have been in a room (in particular, they may be there now).
- A product has a density
- A person has been dead for a certain amount of time (in minutes) at the time of discovery.

## 1.3 Creating and describing instances

You’re going to describe some facts based on the ontology you’ve just built. After representing each of the following facts, run the Hermit reasoner, analyze the deductions made. Make any necessary modifications to the ontology.

- The scalpel is in the Greenhouse.
- The scalpel and the scalpel are one and the same entity.
- Ana Conda is in the Library.

- The Library is on the second floor.
- Omar Tyr is no longer alive.
- Omar Tyr was discovered 3 minutes after his death.

## 2 Guided construction and instancing of the heavy ontology

As it stands, the ontology allows only a limited number of inferences. The next step is to add additional axioms relating to our classes and properties, in order to enable reasoning to identify the killer. Using class properties (disjunction, disjoint union...), we'll express the following knowledge:

- Any product is either liquid or powder.
- A house cannot be a floor.
- A floor cannot be a room.
- People can be alternately victim, murderer or innocent.

Using the characteristics of relations (Functional, Symmetric, etc.) express the following knowledge:

- A product can only have one density
- If location A is located in location B and location B is located in location C, then location A is located in location C (use the characteristics of the relation)
- A room cannot be its own neighbor.
- If a room A is a neighbor of B, then B is a neighbor of A.

Describing class equivalences using Manchester syntax<sup>3</sup>, express the following knowledge (NB: it may be necessary to add intermediate classes):

- A victim is no longer alive.
- A murderer is someone who has killed at least one person.
- A victim has one and only one cause of death.
- A cause of death by gunshot is due to a firearm.
- A crime scene is a room in which a victim is found.
- A murderer is a person who possessed the object responsible for the murder, and was present in the room where the crime took place.

---

<sup>3</sup><http://www.w3.org/TR/owl2-manchester-syntax/>

- A powdered product has a density greater than 2.5 (ua).
- A liquid product has a density of less than 2.5.
- A person shot to death is discovered in less than five minutes.
- A person killed by other means is discovered in more than five minutes.

Finally, using property chains, fill in the axioms corresponding to the following assertions:

- If a person is or has been in a place, he or she has been in any containing place.
- If a room contains an area organized around an object, then it contains said object.
- Conversely, if an object is in a room, then the zone of which it is the center is also in the room.
- If a person owns an object located in a room, that person is also in that room.

Before attempting to solve the riddle, let's add the following instance facts:

- The greenhouse is a floor
- The greenhouse contains the room greenhouse.

Depending on your representation, you may then get an error when launching the reasoner. What is the source of this error? Suggest a solution.

### 3 Puzzle solution

We're now going to add a large number of individuals to our ontology, in order to test the reasoner's ability to solve a complex puzzle.

You have three .csv files, containing information on people, objects and rooms respectively. To create suitable rdf data, we'll use a mapping. To create this mapping, we'll use the RMLEditor graphical interface<sup>4</sup> <sup>5</sup>. To create the mapping, proceed as follows:

- Accept to add a basic URI. Add a new namespace `cluedo`, corresponding to your ontology's namespace (accessible from Protégé).
- For each file, drag and drop the headers of the various concepts. Remember to distinguish between entities (target of `ObjectProperty`) and literal values (target of `DatatypeProperty`).

---

<sup>4</sup><https://app.rml.io/rmleditor/>

<sup>5</sup>A tutorial is available here: <https://www.youtube.com/watch?v=2y3M8QuGZpY>

- If necessary, use functions on the values contained in the columns to obtain the desired URIs. In particular, make sure that the URIs produced for individuals already defined in the previous step match. Similarly, for objects, the type column corresponds to the object sub-type to which the object in question belongs. You must therefore ensure that the URIs also match.
- Use the interface to specify the type of each entity/value. For entities, you can't use the search function, as your ontology isn't available online. Simply copy the class URIs from your ontology and paste them into the interface.
- Now that you've done that, add the relationships between the various components of the same file. For each, specify the name of the corresponding relationship in your ontology. For object typing, use `rdf:type`.
- Once the mapping is complete, start generating the results using the **Play** button on the right-hand side of the interface. View the results to check their consistency.
- When the result is convincing, export it as a Turtle file.

Open the resulting Turtle file. Based on what you observe, add the triples representing the following knowledge:

- The Anteater Manor is a house, and it contains four floors (Ground floor, 1st floor, 2nd floor and the Greenhouse floor).
- The scaffold for painting is organized around the mural painting.

Having done this, copy the contents of the Turtle file, and add it manually to your ontology file. Once this has been done, open the file in Protégé.

Run the reasoner, and observe the inferences. Despite everything we've represented, we still can't infer who the murderer is. This is due to the open-world assumption underlying OWL reasoning. This states that we cannot conclude on the truth of a fact that is not expressed in our knowledge base. In particular, nothing says that the entities described are the only ones to be considered. For example, there may be other firearms in the house that have not yet been discovered. To enable the reasoner to conclude that an individual is guilty, we'll need to add axioms that restrict us to our case study:

- Declare that there is only one instance of cause of death, which we'll create as a new entity. Synchronize the reasoner, and observe the new inferences.
- The reasoner can now infer that the victim was shot. How can he infer which weapon is the murder weapon?
- It can be seen that, while we can now infer who is the murderer and who is the victim, we can't say who is innocent. Suggest a solution.

And that's it, now the reasoner can conclude on the status of all the protagonists of our enigma. It should be noted, however, that the methods used to reach this conclusion are limited in their application. At the scale of the Web, an ontology can be reused in multiple knowledge graphs. By axiomatically restricting the instances belonging to a class, we prevent the reusability of our ontology in multiple knowledge graphs.

Conclusion: the open-world assumption avoids inferring erroneous facts due to lack of knowledge, but restricts the automatic reasoning enabled by OWL.