CITY
UNIVERSITY OF LONDON
EST 1894

INM713 Semantic Web Technologies and Knowledge Graphs

Laboratory 5: Modelling OWL 2 Ontologies with Protégé

Ernesto Jiménez-Ruiz

Academic course: 2020-2021
Updated: February 18, 2021

Contents

1 Git Repositories

Support codes for the laboratory sessions are available in *GitHub*. There are two repositories, one in Python and another in Java:

`https://github.com/city-knowledge-graphs`

2 Ontology editor Protégé: Demo

Protégé is an editor of OWL ontologies (`https://protege.stanford.edu/`). We will use the Desktop version.

I will rely on an exercise similar to the one in the lab session 2.

- Change *Ontology IRI* in "Active ontology/Annotations". *e.g.*, `http://www.semanticweb.org/inm713/lab5`.
- Create *prefixes* in "Active ontology/Ontology Prefixes".
  - city: `http://www.example.org/university/london/city#`
  - foaf: `http://xmlns.com/foaf/0.1/`
- Create *classes* Person and Module in Tab "Entities/Classes":
  - `city:Module`
  - `foaf:Person`
  - `city:Lecturer`
- Create a *subclass* relationship between `foaf:Person` and `city:Lecturer`.
- Create an *object property* `city:teaches` in Tab "Entities/Object properties" .
- Create a *data property* `foaf:name` in Tab "Entities/Data properties".
- Create an *axiom* that indicates that a lecturer teaches at least one module and only modules.
- Create an *axiom* that indicates that person has at most one name.
- Create *individuals* in Tab "Entities/Individuals".
  - `city:ernesto`
  - `city:inm713`
  - `city:inm373`
- Assign *types* to `city:ernesto` and `city:inm713`. Select a type from class hierarchy.

- Create *Object property assertions* (*e.g.*, triples with property city:teaches) associated to the individual city:ernesto.

- Create a *Data property assertion* (*e.g.*, a triple with property foaf:name) associated to individual city:ernesto.

- Indicate that city:ernesto is teaching city:inm713 in 2021.

- Perform classification/reasoning.

3 The Pizza ontology

The pizza ontology and its tutorial are well-known resources in the Semantic Web community. They were developed for educational purposes by the University of Manchester. The tutorial have been recently updated by Michael DeBellis.

The pizza ontology and the tutorial are found at:

- https://tinyurl.com/NewPizzaTutorialV2

- http://protege.stanford.edu/ontologies/pizza/pizza.owl

In the following exercises we are going to explore the Pizza ontology and the ontology editor Protégé.

Exercise 3.1: Open the pizza ontology in Protégé. Take some time to browse the class hierarchy, the property hierarchies and the individuals and note how the ontology describes the domain of pizzas.

Exercise 3.2: Find Margherita and see how it is defined as a pizza with only cheese and tomato topping. Look at the definition of VegetarianPizza. Is a MargheritaPizza a vegetarian pizza? Why / why not?

Exercise 3.3: Find hasIngredient. What is the domain and range of this property? What are the subproperties of hasIngredient? What is the inverse property of hasIngredient? What property characteristics does hasIngredient have?

Exercise 3.4: Classify the ontology by choosing a reasoner (*e.g.*, HermiT) and then "classify/start reasoner" in the reasoner menu. In the entities/classes tab we can chose between the asserted classification and inferred classification (as a results of reasoning).

- In the "Inferred class hierarchy" two classes show up as subclasses of owl:Nothing. What does it mean for a class to be a subclass of owl:Nothing? Why these two classes appear as subclasses of owl:Nothing?

- Find Margherita in the inferred class hierarchy and see which classes are inferred as superclasses of Margherita.

Exercise 3.5: Add a new class RomanoPizza as a subclass of NamedPizza. Define RomanoPizza as a pizza with:

- hasTopping some AnchoviesTopping,
- hasTopping some TomatoTopping and
- hasTopping some MozzarellaTopping.

Classify the ontology. What superclasses are inferred as superclass of RomanoPizza? Why?

Exercise 3.6: Follow the tutorial Chapters 1-8:
https://tinyurl.com/NewPizzaTutorialV2.

4 Real-world Ontologies

There are a large number of ontologies that are being currently applied in real-world solutions. Prominent examples are the ontologies in life sciences. For example, BioPortal (https://bioportal.bioontology.org/) contains 842 ontologies.

SNOMED CT ontology is one of the largest ($\geq 300,000$) and most important ontologies. SNOMED CT vocabulary has been adopted by the NHS. There are also other smaller ontologies with a more focused scope (***e.g.***, CMR-QA, an ontology I developed in the past, https://bioportal.bioontology.org/ontologies/CMR-QA).

Ontologies can also be found in many other domains (***e.g.***, Food: https://foodon.org/, Oil&Gas: https://gitlab.com/logid/npd-factpages).

Exercise 4.1: Explore some of the above ontologies with Protégé.

5 Managing Ontologies Programmatically

Both OWLready and the Jena API include methods to load, explore and extend ontologies. The creation of the ontologies, however, is typically done via editors like Protégé.

Exercise 5.1: Extend the scripts (loadOntology) in the respective GitHub repositories (folder/package lab5) to print out the list of object properties, data properties and individuals. For OWLready, I have created an interface to access the main method which may be useful as reference (onto_access.py); otherwise one can directly access the available OWLready methods. Note that the onto_access.py interface has an additional dependency: logging module https://pypi.org/project/logging/.