

Data Science Workshop Series - Ontologies

Institute of Data Science @UM

Ontologies are usually constructed by domain experts together with ontology engineers. Domain experts provide knowledge about the key entities and their relationships within the particular topic area, while the ontology engineers decide how to formalise this knowledge using their expertise in ontology languages, best practices and ontology editing software. In this case, we chose an ontology topic in which we are all experts :) An example solution ontology is provided in “resources/movie.owl” but don’t be tempted to peek until you have attempted :)

Practical 1: Building an ontology in Protégé.

For this task we would like you to build an ontology about movies.

We will be guiding you step-by-step through the task during the workshop. However, below you will find some **general guidelines and tips** on how to go about it:

1. Always think first about what questions you would like to ask your ontology. This will help you pick the appropriate terms, relations and axioms to represent the knowledge in the best way to answer them.
2. Start by giving your ontology a unique URI (a name usually in the form of a URL where the ontology is hopefully going to be published and thus become dereferenceable)
3. Using OWL annotations, add some descriptive metadata about your ontology such as title, description, authors (contributors) etc.
4. Add class and relation terms to your ontology relevant to the topic (don’t try to think too much about how these relate to each other yet!).
5. Categorise the terms in the class and relation hierarchies according to simple subclass/superclass relationships and synonyms (equivalence terms).
6. Think about more complex entities and what subclass relations can exist between them (e.g. movies that are funny and scary are comedy horrors). Try to formalise these using OWL subclass axioms NB: this will be the more challenging task for many of you because it requires an understanding of OWL semantics.
7. Try adding some individuals (data instances) to test if your knowledge is correct. I.e., if it accurately reflects the topic and it is free from logical errors (use the reasoner periodically to check).
8. Please annotate your entities with OWL annotations to describe what each term represents in the topic. This is like code comments in software engineering - best practice top tip!

Time: 90min (10:30 - 12:00)

Practical 2: Ontology based expert system

We have built a small prototype application “[UMMedSystem.jar](#)” to demonstrate ontology-based querying of disease information. The data is in local HTML pages in “[resources/data](#)” each providing a short description about a disease and a list of its common symptoms. The tool has two features: disease classification and disease comparison. Disease classification uses “[resources/diseases.owl](#)” to classify these diseases into more general disease categories. Disease comparison is a simple feature which finds common symptoms among two diseases.

Task 1:

Experiment with the features of the tool and monitor the results for different queries. Open “[resources/diseases.owl](#)” and examine the structure of the ontology. Use the DL query tab in Protege to “ask questions” about the ontology to get an idea of its structure. For example, you can ask for the subclasses of classes (both named ones and complex ones). Make a note of the design of the ontology and formulate an opinion of how you would have designed such an ontology. Would you follow a similar structure or would you change it? You may also use “[OntoGraf](#)” plugin, if you have it, to see a graphical representation of the ontology.

Task 2:

Pick one or two disease categories from UM Medical Expert System which do not have any diseases belonging to them yet (i.e., there are no disease data files in “[resources/data](#)” falling under these categories).

1. Look for information about diseases online that you know to belong to this category.
2. Pick one or two diseases from these which actually appear in the ontology (you can use the “search” feature in the top right corner in Protege to do this.)
3. For each of these chosen diseases, create new disease data files in “[resources/data](#)” similar to the others which give a short description of the disease and a few key symptoms.
 - a. What is really important here is to **annotate** these new HTML files with ontology terms from the disease ontology (“[resources/diseases.owl](#)”). Look at some other examples in the “[resources/data](#)” folder to see how to do this.
 - b. Save the data files using a descriptive file name (preferably the disease name).
 - c. Test the Expert System to see if it now classifies your new data file under the correct disease category.
4. We would also like to know which part of the body the disease affects or originates from. Add an axiom to the ontology to state this. Test whether this information is now reflected in the Expert System.

- a. The “originatesIn” relation in the disease ontology should be used in your axiom.
- b. The part of the body which the disease affects should be a subclass of “Organ” in the disease ontology.

Time: 90min (14:00 - 15:30)