Semantic Web
Lab Assignment 4

Goal: *Be able to create ontology and instance data using Protégé and OWL*

Download and install the latest release of Protégé at https://protege.stanford.edu/.

Windows, Linux, and Mac users

Scroll down to "Protege Desktop"

Select "Download for (your OS)"

"Protege Desktop":

The site asked you to register. Then, download the latest version (as of 10/14/2019, "Protégé Desktop 5.5").

The platform independent version may be the best option for novice computer users as it does an automatic install.

Save to local machine (in handy location).

To run:

See the installation instruction on the Protégé website for your OS at:

https://protegewiki.stanford.edu/wiki/Install_Protege5

Other users

Select "Use Web Protégé"

You may use the Web Protégé hosted by Stanford, or install it locally. I like my tools to be local, but it is up to you.

The tutorial, below, was written for the Windows version. Owl:imports are a challenge: create a basic ontology, save it, and download it to local file. Add owl:imports using a text editor, then upload it and it will ask additional questions.

1. Protégé OWL Introduction

See https://protegewiki.stanford.edu/wiki/Main_Page for documentation. In the "Tutorials and Getting Started" section, see the "Getting started with the Protege Desktop editor - a short guide to the interface" for a short presentation on essential screens. Also, see the "Protégé OWL Tutorial: A step-by-step guide to modeling in OWL using the Protégé OWL tools" for a PDF tutorial with updated references. Also, see YouTube videos on Protege.

Work through Chapters 1 - 4.7 of the Protégé Owl 4 Tutorial v1.3:

See the file "Protege OWL Tutorial v1_3.pdf"
Same link as above:
http://owl.cs.manchester.ac.uk/publications/talks-and-tutorials/protg-owl-tutorial/

You may ignore any discussions regarding add-ons and prerequisites. The serious changes are at Protégé start-up (near the beginning of Chapter 4):

After getting started, the tutorial says a "Create Ontology URI" wizard will appear (but it doesn't appear – the tool starts up ready for work in OWL).

NOTE: Protégé properly calls the namespace an IRI instead of URI in the tutorial.

- Use the tutorial's provided Ontology IRI (as directed in Exercise 2).
- Put ontology file in a convenient location you can find again.
- When you save your files, select OWL/XML as the type of file. You can store in other formats, like TTL, as well.

For more information on the opening screens see (as above):
http://protegewiki.stanford.edu/wiki/Protege4GettingStarted
and note that we use the Protégé Owl views)

After the wizard finishes, the view is set to the "Active Ontologies" tab. We haven't added the common plug-ins, so the view is simpler than in the Tutorial. Add the comment as described in the Tutorial and proceed on to the Tutorial's section 4.1.

Continue performing the instructions of sections 4.1 through 4.7 of the Tutorial (using hints below to make it through since the tutorial is a bit different than the current tool).

Hints:

Exercise 6 – You may have to scroll the Description pane to see the added disjoint classes.

Exercise 8 – You have to select the topObjectProperty in the window and select to make a sub-property of it.

Exercise 10 – Step 1: You have to select the topObjectProperty in the window and select to make a sub-property of it.

Step 4: In the sub-window you should first select isIngredientOf and make sub-property isBaseOf

Step 6: You have to first select isIngredientOf and make sub-property isToppingOf

Exercise 13 – Step 3: When the dialogue pops up, select the class Hierarchy tab and navigate to the desired class.

Before Exercise 15 (NOTE) – We have not added the necessary plug-in's for Protégé to automatically populate the inverse properties with domain and range.

2. **Read and understand how Protégé Handles Owl Imports:**

   http://protegewiki.stanford.edu/wiki/Importing_Ontologies_in_P41
   The older docs may also provide some helpful information:
   http://protegewiki.stanford.edu/wiki/How_Owl_Imports_Work and
   http://protegewiki.stanford.edu/wiki/How_Owl_2.0_Imports_Work
   The link from the P41 page to the Owl_2.0 page is misconfigured, so you'll need to use the above two URLs.

   Note that previous versions of Protégé had an Ontology Repository Manager to help specify where to retrieve ontology files. In this version of Protege, the Ontology Repository Manager is now a mixture of items on the file menu and in the Active Ontology tab that look nothing like in the document above, but the required functionality is available:

   a. Loaded ontology sources
   b. Edit ontology libraries
   c. Edit active ontology library
   d. Gather ontologies
   e. Add the Ontology imports item (on the Active Ontology tab)

3. **Create a new ontology file with IRI:**

   The ontology will use the IRI "http://utdallas.semweb/class".
   It should supports OWL/XML. Save project in file named "lab4.owl" in an empty directory named Lab4_3_<YourID> where <YourID> should be replaced with your UTD NetID.

   Modify the metadata of the ontology to include prefix "utdswc" (short for UT Dallas Semantic Web Class) associated with the namespace "http://utdallas.semweb/class#".

   Add a copy of the FOAF ontology named "FOAF.rdf" to the same directory as your owl file. Then, add it as an import to your ontology:

   a. Copy the provided file, "foaf.rdf" to the same directory as your ".owl" file.

   b. Add the file as an import to your ontology:

      With your ".owl" file open in Protégé, on the Active Ontology tab, there is a tab near the bottom named Ontology Imports – select that and choose "Direct Imports" and the select "Import an Ontology contained in a specific file".

      Your ontology now imports (i.e., includes) FOAF, which is relatively large and somewhat complicated and additionally uses some of the Dublin core definitions. When Protégé first adds this import, it creates a "catalog*.xml" file in the directory—leave that for Protégé to use. You should review the file. The contents should start to make sense for how Protégé handles owl imports.

Then add a class named "utdswc:Person" and say that "foaf:Person" is a subclass of it where "foaf:" refers to the appropriate namespace (also, the Superclasses item is useful). Note that you need to hover your mouse over class names in the "Class Hierarchy" view to understand their namespaces. The view setting can show or hide the namespaces.

Add the following **datatype** properties with noted domains and ranges:

Property	Comment	Domain	Range
utdswc:title	Name of position held as part of an organization	utdswc:Person	xsd:string
utdswc:email	eMail address	utdswc:Person	xsd:string
utdswc:award	Award received for recognition of achievement	utdswc:Person	xsd:string

Add an instance, "utdswc:QuentinTarantino", of type "utdswc:Person" with properties shown in the table below. Note that the prefix "utdswc:" refers to the default namespace (the Individuals tab in the Members List area and then Properties assertion area may be of some help):

Property	Value
utdswc:email	QuentinTarantino@directors.org
utdswc:title	Actor
utdswc:title	Screenwriter
utdswc:title	Director
utdswc:title	Producer
utdswc:award	Academy Award for Best Original Screenplay
foaf:givenName	Quentin
foaf:familyName	Tarantino
foaf:gender	male

4. Save

Save your project, restart Protégé, and reload your project. Confirm the desired structures and data are still there. Be prepared to discuss this lab and any related issues in class.

Please compress your entire directory of owl files from steps 3 and 4 into one zip file and submit the zip file on eLearning.

Grading (100 points):

-100	Nothing submitted
-25	FOAF.rdf not included
-10	Incorrect “utdswc:” namespace
-10 each	Missing the “utdswc:” property definitions
-5 each	Missing “utdswc:” domain definitions
-10	Lacking FOAF import
-10	FOAF import from web rather than local file
-5	Incorrect directory or filenames