UNIVERSITY OF WATERLOO
Cheriton School of Computer Science

CS486/686 **Introduction to Artificial Intelligence**
Fall 2016

ASSIGNMENT 2:
AN OWL ONTOLOGY FOR A QUESTION-ANSWERING APPLICATION

Due by Monday October 31 5:00pm to the CS486/686 LEARN Assignment 2 DropBox
This assignment may be done individually or in teams of 2.

Assignment 2 Problem: An Ontology for Question-Answering

Your task for Assignment 2 is to construct an OWL ontology for a specific domain that will be the "knowledge base" for an automated Question-Answering (QA) system.

Question-Answering applications can have many purposes, for example:

(a) Online product help line;
(b) Automated "concierge" for making reservations (e.g., flights, hotels, restaurants);
(c) Virtual city guide;
(d) Newsfeed querying app;
(e) Digital literature analysis;
(f) Query engine for a game AI giving information about the current status of the game and characters;
. . . and so forth.

You are encouraged to be creative about why and how a QA system might be used.

A Simple Knowledge Engineering Methodology[1]

Some fundamental rules in ontology design:

1. There is **no one correct way** to model a domain—there are always viable alternatives. The best solution almost always depends on the application that you have in mind and the extensions that you anticipate.
2. Ontology development is necessarily an **iterative** process.
3. Concepts in the ontology should be close to objects (physical or logical) and relationships in your particular domain. These are likely to be nouns (objects) or verbs (relationships) in sentences that describe your domain.

[1] Material from Natalya F. Noy & Deborah L. McGuinness "Ontology Development 101: A Guide to creating your first ontology"; Tania Tudorache, "Ontology 101: The basics of developing ontologies", Protege Short Course, Stanford, March 21, 2016.
This paper is available on our LEARN site in "ASSIGNMENT2-MATERIALS".

Step 1: Choosing a Domain for a Question-Answering Application

To begin with you should choose **only one domain** to focus on and define its scope—answer these basic questions:

- What is the domain that the ontology will cover?
- For what types of questions the information in the ontology should provide answers?
- Who will use and maintain the ontology?

The answers may change during the ontology-design process, but at any given time help limit the scope of the model.

Step 2: Competency Questions

It is helpful to begin by listing a set of "competency questions", i.e., the typical questions the ontology will be used to answer. Competency questions are helpful for:

- Defining the scope of the ontology;
- Defining the vocabulary (main terms) of the ontology;
- Testing the ontology.

For example, for a Shopping ontology we might have such competency questions as:

- What beverages are sugar-free?
- What types of gluten-free breads are there?
- Does this product contain nuts (or traces of nuts)?
- Is this product sold at any stores in Waterloo?

Some further tips on Competency Questions:

- Competency questions can have different forms:
  - Yes/No questions: "Is this product high in salt?"
  - Wh-questions: "Which supplements are vegetarian?"; "Where can I buy dairy-free ice cream?"
- Try to formulate questions that can be **formalized**:, e.g., "What supplements are vegetarian?"
- Try to cover different areas of the domain **systematically**.
- Have a brainstorm session with yourselves playing the role of domain experts to come up with these questions.

  Each person might cover a different area of the domain, or you can develop your own brainstorming strategy.
- At the end of the session, group the questions, and document them;
- Create **test cases** based on these questions—use these questions throughout your ontology development.[2]

[2] We can use DLQuery in OWL's Protege editor to assist with these test cases.

Step 3: Identifying the Terms

Next, write down a list of all terms you would like either to make statements about or to explain to a user.

- What are the terms you would like to talk about?
- What properties do those terms have?
- What would you like to say about those terms?

The next two steps—developing the class hierarchy and defining properties of concepts (slots)—are closely intertwined. Lecture 9's OWL Tutorial describes these basic strategies for systematically building up the class hierarchy:

- Top-down.
- Middle-out.
- Bottom-out.

Step 4: Defining the Properties of Classes

Once you have defined some of the classes, you must describe the internal structure ("slots" and their fillers) of concepts—the properties of classes. Properties associated to a class describe the attributes and relationships of the instances of the class.

- Relations between classes are modeled using *object properties*, e.g., "All Flights have an Airline", "All Flights have a DepartureLocation", "All Flights have a DestinationLocation".

  You might also have **specializations** of relations (object properties).
- In addition, you will need to define *data properties*, i.e., a class property value like: "All Flights have a TimeDuration". This could be modeled by a "hasDuration" data property for class Flight with value an Integer range (e.g., 0–15 hours).

Step 5: Creating Instances (Individuals)

The last step in ontology-building is creating individual instances of classes in the hierarchy.

Interative Step: Using DLQuery to Test Your Ontology

As you develop your ontology, DLQuery can be used with your competency questions to ensure that correct assertions about your domain have been captured in the ontology. DLQuery can help test whether any incorrect axioms have been inadvertently added to the ontology. For example, in the Pizza ontology, "ice cream" is defined as a type of "FruitTopping". But "FruitTopping" is defined to be a type of "PizzaTopping". If we pose the following query to DLQuery to find all subclasses of this definition:

```
Pizza and (hasTopping some FruitTopping)
```

DLQuery will return "ice cream" as a subclass of this kind of Pizza!!!

Interative Step: Using OntoGraf to View Your Ontology

As you develop your ontology, OntoGraf allows you to visualize your ontology using different views and filters. This can be helpful in designing the ontology, e.g., if your ontology is flat and wide this may indicate you are not modelling sufficient specialization relationships in your domain.

Iterative Step: Documenting Your Methodology

Ontological "engineering" is a misnomer at present—at best even professional ontology developers have only tips and guidelines for best practices to follow. A single basic type of ontology development strategy (i.e., top-down, middle-out, bottom-up) may work for your domain. It is likely though that a combination of these strategies, plus other sub-strategies you develop (e.g., when to invoke DLQuery or OntoGraf) will be more relevant for your particular domain. (An interesting outcome from marking this assignment will be comparing the methodologies.)

Final Tip: Save your updates frequently.

Also, as you add definitions to your ontology, run the Reasoner to ensure there are no inconsistencies in your ontology. If you can't debug an error, record it and inform the Instructor or TA who will help correct the problem.

What to Hand In

Each team should submit the following to the online Assignment 2 Dropbox on our CS486/686 LEARN site:
(You may continue to update your submission until the deadline.)

(1) One copy of your OWL file, titled <TeamMember1LastName-TeamMember2LastName>.owl
See item (c) below for specifications for your ontology.

(2) One Word or PDF file, with answers to questions (a) and (b) shown below.
Don't forget: Include both your team members' names with your responses to (a) and (b).

(a) Give a brief (one-paragraph is sufficient) description of your domain and question-answering application, why you chose an existing type of QA system, or why you decided to invent a new QA application.

(b) Document **in detail** (3-4 paragraphs) the steps in your methodology for developing the ontology. What did you find to be the pros and cons of using this strategy?

(c) Your ontology should include the following types of definitions:

- 10–12 different Classes. These Classes should have subclasses, and at least one Class should have at least three levels deep of subclasses.
- 5–6 Object properties. Some of these Object properties should have subproperties.
- Several Object property restrictions: Quantifier restrictions (both Existential and Universal) and Cardinality restrictions.
- Several Datatype properties and restrictions.
- Individuals of several different Classes.
- At least one defined class, i.e., has necessary and sufficient conditions.
- At least one complex class, i.e., has two or more Boolean operators in its class definition

Annotations in your OWL ontology are strongly encouraged to assist the marker in understanding your ontology and giving all the credit deserved.