# Knowledge and Data: Practical Assignment 4

## Modelling in OWL

* YOUR NAME: Sebastião Manuel Inácio Rosalino
* YOUR VUNetID: sxx209

(If you do not provide your NAME and VUNetID we will not accept your submission.)

For this assignment you will be engineering and reasoning over your very own OWL ontology.

You are free to choose the domain (subject) of the ontology you are going to build (e.g. on nutritional value, recipes, supermarkets, food safety, health, restaurants, planes, trains and automobiles, developing countries, modern slavery, political parties, refugees, you name it...). Just be creative and choose a domain we have not seen in class yet.

We expect extensive answers for this assignment: give a full account of what you did, such that a peer would be able to reproduce your ontology. This means that you must explicitly state the new axioms (in some Human readable syntax, such as Manchester Syntax, so no ttl) and that we expect you to motivate your choices (usually 1-3 lines).

**IMPORTANT: You will have to hand in your ontology (as ttl) together with the notebook**

### Learning objectives

At the end of this exercise you should be able to build and ontology and to reason over it:

1. You will be able to engineer an OWL ontology
2. You will be able to conceptualize a (small) domain
3. You will be able use conditions and property characteristics
4. You will be able to use a reasoner to infer implicit knowledge
5. You will be able to work in Protégé

### Preliminaries

There are several tools which can be used to create and edit RDF and OWL files (in addition to your favourite text-based editor). For this assignment we urge you to use the open-source tool [Protégé](https://protege.stanford.edu/), which is a stand-alone editor that is very much tailored towards OWL ontology editing.

To install Protégé on your system, please take a look at the [installation instructions](https://protegeproject.github.io/protege/installation/).

Protégé is a complex tool with many options, only few of which we will need for this assignment. There are various resources available to get you started:

* Watch a short [Protégé Screencast Tutorial](https://vimeo.com/183829740) (created by Rinke Hoekstra)
* Check out the [Practical Guide To Building OWL Ontologies Using Protégé 5](https://www.researchgate.net/publication/351037551_A_Practical_Guide_to_Building_OWL_Ontologies_Using_Protege_55_and_Plugins) that uses the [Pizza ontology](https://protege.stanford.edu/ontologies/pizza/pizza.owl) to describe how to create ontologies using Protégé.
* Check the [Assignment 4 document](https://docs.google.com/document/d/1Dw2winjfr2TJq3r1q6ZRpud9Qn--tq4ioWDqiBdrrzI) containing Tips & Recommendations on how to create ontologies.

### Task 1 (1 Point) : Creating an empty ontology

Create a new empty ontology in Protégé.

Be sure to:

* Choose your own unique ontology IRI, its namespace, and a prefix
* Add metadata in the form of rdfs:label, rdfs:comment, and rdfs:isDefinedBy annotations.  
  Use rdfs:comment to describe the domain and target audience of your ontology.

Write down the IRI, its prefix, and the metadata that you have added in the textfield below:

IRI: <http://www.semanticweb.org/sebas/ontologies/knowledge_and_data/assignment_4>

Namespace:

\* http://www.w3.org/2002/07/owl#

\* http://www.w3.org/1999/02/22-rdf-syntax-ns#

\* http://www.w3.org/2000/01/rdf-schema#

\* http://www.semanticweb.org/sebas/ontologies/knowledge\_and\_data/assignment\_4

\* http://www.w3.org/XML/1998/namespace

\* http://www.w3.org/2001/XMLSchema#

Prefix:

\* owl

\* rdf

\* rdfs

\* sebas

\* xml

\* xsd

Metadata: The data there's always there. Built-in protege things. For example, OWL:Thing. Add your properties. RDF Types, RDFS Syntax. Everything from RDF/RDFS semantics I've used. Properties of own entologies. Example Human label Human comment human being of Homosapiens. Use rdfs.label on every classes. Use comments. Labels for every class. Use everything from the Annotations. Only for the ontology. Comments should be enough.

### Task 2a (1 Point) : Populating your ontology

Populate your ontology such that it contains

* six classes
* three data properties
* four object properties

For **each** class, create **two** example instances:

* assert one instance as a member of this class (i.e. using rdf:type).
* leave the second instance without any type.

Your ontology should now have 12 instances: six instances with a certain rdf:type, and six instances without any rdf:type.

List and describe the 6 classes you created in the textfield below, together with their instances. Don't forget to motivate your choices.

Class Name: **faculty**

Description: As my ontology is based on a university structure (using VU University's situation), it is essencial to create a class covering the multiple faculties that may be present.

Instances: faculty\_business\_economics, faculty\_of\_science (assigned rdf:type)

Class Name: **major**

Description: This class includes the degrees possible to graduate in the VU. Whether it is a bachelors or a masters.

Instances: data\_science (assigned rdf:type), economics, management

Class Name: **popularMajors**

Description: This class includes the bachelors and masters degrees offered by VU that have at least 500 students enrolled on them. This is a subclass of the major class. It is impossible for this class to have explicit instances assigned to it, because this is a class with a pre-condition that we do not know at the beggining for each major (=> 500 students).

Instances: economics, management

Class Name: **course**

Description: This class includes all the courses offered by the VU's majors.

Instances: articial\_intelligency, intelligent\_systems (assigned rdf:type), knowledge\_and\_data

Class Name: **people**

Description: People class includes any human being, whether a student, a professor or from outside from VU academic environment.

Instances: JohnDoe (assigned rdf:type), K.S\_Schlobach, VictorDeBoer, MariaScharakova, SebastiaoRosalino

Class Name: **professor**

Description: This class includes any professor lecturing on VU. Noteworthy to mention that it is necessary for someone to lecture at least on course to qualify as a professor.

Instances: K.S\_Schlobach, VictorDeBoer (assigned rdf:type)

Class Name: **student**

Description: This class includes any student attending the VU. Noteworthy to mention that it is necessary for someone to be enrroled in at least on course to qualify as a student.

Instances: MariaScharakova, SebastiaoRosalino (assigned rdf:type)

Class Name: **research**

Description: This class represents any investigation on any domain being conducted and financed by the VU.

Instances: advanced\_robotics (assigned rdf:type), social\_and\_economics\_inequalities

### Task 2b (1 Point) : Asserting properties

For each instance:

* assert at least one data property (e.g. ex:instanceA ex:hasFullName "Some Full Name").
* assert at least one object property, relating instances to each other (e.g. ex:instanceA ex:attendsCourse ex:instanceB)

List and describe 3 statements from your ontology, containing 3 different data property assertions:

First Data Property - **hasBudget**:

Brief description: This Data Property's domain is an instance of type research and it's range points at a xsd:decimal. This property was created with the intention to assign a research under the auspices of the VU to it's initial budget (it was built so that it could support budget values with decimal places).

Triple created: sebas:advanced\_robotics sebas:hasBudget 500000 .

Second Data Property - **hasFullName**:

Brief description: This Data Property's domain is an instance of type people and it's range points at an rdfs:Literal. This property was created with the intention to assign a fullname to people present in this ontology (it was built so that it could support any name, with spaces, from any language, with any special characters).

Triple created: sebas:SebastiaoRosalino sebas:hasFullName "Sebastião Manuel Inácio Rosalino"^^xsd:string .

Third Data Property - **hasNumberOfStudents**:

Brief description: This Data Property's domain is an instance of type major and it's range points at a xsd:integer. This property was created with the intention to declare some major's number of enrolled students for informational purposes. It was made to support any integer number of students.

Triple created: sebas:data\_science sebas:hasNumberOfStudents 325 .

List and describe 4 statements from your ontology, containing 4 different object property assertions:

First Object Property - **isColleague**:

Brief description: This Object Property's domain and range must be instances of type student. This property was created with the intention to establish an academic relathionship between students. This Property was meant to be Symmetric and Irreflexive meaning that, on what symmetry is concerned, if someone is colleague of a third party, that third party is also a colleague of someone. Speaking about irreflexivity, it was stated that someone can never be colleague of themselves, in order to avoid miscalculations for any course's group formation for instance.

Triple created: sebas:SebastiaoRosalino sebas:isColleague sebas:MariaScharakova .

Second Object Property - **isBeingLeadBy**:

Brief description: This Object Property's domain must be of type research and it's range of type people. This property was created with the intention to state that an on-going investigation must be led strictly by only someone. This Property was meant to be Inverse Functional meaning that if an investigation is being led by two differently represented people instances they must point to the same people entity.

Triple created: sebas:advanced\_robotics sebas:isBeingLeadBy sebas:K.S\_Schlobach .

Third Object Property - **isEnrolledIn**:

Brief description: This Object Property's domain must be of type student while it's range must be of type course. This property was created with the intention to state the enrollment of a student into a certain course.

Triple created: sebas:SebastiaoRosalino sebas:isEnrolledIn sebas:knowledge\_and\_data .

Forth Object Property - **teaches**:

Brief description: This Object Property's domain must be of type professor while it's range must be of type course. It is assumed in this ontology that a course can only be taught by a single professor.

Triple created: sebas:VictorDeBoer sebas:teaches sebas:knowledge\_and\_data .

### The reasoner

The questions following this point make use of Protégé reasoning capabilities, which are available via plugins but which are disabled by default. Install (if necessary) and start the Pellet reasoner before you continue with the next question.

Refer to page 15 of the [Protégé guide](https://www.researchgate.net/publication/351037551_A_Practical_Guide_to_Building_OWL_Ontologies_Using_Protege_55_and_Plugins) for instructions on how to install and run the reasoner.

### Task 3 (0 Points): Reasoning on a basic ontology

All assertions that were addded up til now were explicit. Yet, it is certainly possible that your ontology also contains one or more implicit assertions, that have emerged from the interactions between the added explicit assertions.

Run the reasoner on your yet-basic ontology and write down the inferences occurred (if any) below:

On my Ontology the Reasoner was able to uncover several implicit assertions. I will go through 3 of them in detail below:

First Implicit Assertion -

Brief description: There is a condition saying that every course having 500 or more students is classified as a popularMajor, therefore, as we stated that economics had 521 students, the reasoner was able to infer that economics is also of type popularMajor.

Triple inferred: sebas:economics rdf:type sebas:popularMajors .

Second Implicit Assertion -

Brief description: As we have stated before, the domain of the property teaches must be an instance of type professor, therefore, although we did not specify at first that K.S\_Schlobach was a teacher, as he teaches artifical\_intelligency the reasoner was in conditions to infer that K.S\_Schlobach must belong to the class of professors.

Triple inferred: sebas:K.S\_Schlobach rdf:type sebas:professors .

Third Implicit Assertion -

Brief description: It was stated in this ontology that the property lecturesMajor had as domain a faculty, therefore, although we did not specify that faculty\_business\_economics was of type faculty, as it is responsible for lecturing the major economics, the reasoner inferred that the instance of faculty\_business\_economics is of type faculty.

Triple inferred: sebas:faculty\_business\_economics rdf:type sebas:faculty .

### Task 4a (1 Point): Necessary and sufficient conditions

Select two classes from your ontology.  
For **both** classes:

* add necessary and sufficient conditions (e.g. engineers are people who hold an engineering degree, and any person holding an engineering degree is an engineer)
* infer class membership of at least one instance using each condition

List and describe the four conditions that you have added (i.e. axioms) in the textfield below. Do not forget to motivate your choices.

**First Class**: student

**Necessary Condition**: sebas:hasFullName

As we have stated untill now, any student must be also of type people, that means that having a **fullName** Data Property is a necessary condition to be a student, however, not necessary and sufficient. As for a counterexample of this situation the instance JohnDoe has a fullName property however, that only classifies him as a people and not a student. Therefore, a necessary condition to be a student is to have a fullname, but it is not enough in itself.

**Necessary and Sufficient Condition**: sebas:fullName and sebas:isColleague

As we have stated untill now, any student must be also of type people. In that regard, having a fullName Data Property and being related academiclly with any other student by the Object Property **isColleague** makes an instance be for sure of type student, therefore these are necessary and suficient conditions for the belonging on the class student. Therefore, a necessary and sufficient condition to be a student is in the one hand to have a fullname and on the other hand to have a fellow colleague at the same university.

**Second Class**: popularMajors

**Necessary Condition**: rdf:type major

We know that for any major to be classified as popular it firstly as to be classified as a **major**, as it would be ilogical for some instance to belong to the popularMajors class without belonging at first to the major class. Therefore, a necessary condition to be a popularMajors is to be a major, but it may not be sufficient (because it may have less than 500 students).

**Necessary and Sufficient Condition**: sebas:hasNumberOfStudent **some** xsd:integer[>=500]

Finally, if any instance has a numberOfStudents greater or equal to 500, we unmistakably know that instance is not only of type major but also a **popular major**. Therefore, a necessary and sufficient condition to be a popularMajor is to be a major and have more than 500 students.

### Task 4b (1 Point): Reasoning over conditions

Run the reasoner once again (after having added the two conditions).

Write down the different steps of the reasoning process (ie, what happens when you run the reasoner). Also write down and explain the resulting inferences below.

First in general, then in my case

**First Class**: student

1. First example with MariaScharakova:

MariaScharakova has a fullname, so she has the necessary condition to be a student. However, with just this information it is not possible tem um fullname, logo cumpre a condição necessária para ser estudante. Porém, só com esta informação não podemos concluir que seja estudante porque não está verificada uma condição suficiente para que o seja.

1. As it is known Sabemos, agora, que MariaScharakova é colega de SebastiaoRosalino, logo podemos concluir que está verificada a condição necessária e suficiente, porque sabemos que MariaScharakova tem Fullname e que SebastiaoRosalino é seu colega e é estudante. Logo se a MariaScharakova é colega de SebastiaoRosalino, então ela também é estudante. O reasanor fez esta inferência automática.

Triple Infered: sebas:MariaScharakova rdf:type sebas:student .

Second Reasoning over condition: An instance named JohnDoe (with fullname "John Cameron Doe") was created as just belonging to class people without any colleague or any other academic reference. Thus, as having a fullname is just a necessary but not sufficient condition, the reasoner cannot infer any academic relation between JohnDoe and the VU. So it is not possible to conclude that JohnDoe is a student.

**Second Class**: popularMajors

First Reasoning over condition: Firstly, the major economics was inferred to be a major because it had a Data Property **hasNumberOfStudents** (which has its domain of type major). As it was created the condition that a major having 500 or more students would automaticlly be of type popular major, and the major economics has exactly 521 students, the reasoner was able to infer the membership of the instance major economics to the class popularMajors.

Triple Infered: sebas:economics rdf:type sebas:popularMajors .

Second Inferrence: An instance named BrunoFernandes (with fullname "Bruno Miguel Borges Fernandes") was created without have being specified to belong to any class. After that, it was stated that BrunoFernandes was enrolled in the course intelligent systems. As the Object Property **isEnrolledIn** has as its domain a student the reasoner was able to infer that BrunoFernandes was a student.

Triple Infered: sebas:BrunoFernandes sebas:isEnrolledIn sebas:student .

Student:

A Maria é logo aluna porque é colega do Sebastião

John Doe apesar de ter fullname não é student

Professor:

Sebastião é amigo de VictorDeBoer mas isso não faz dele um professor

A partir do momento que o Victor teaches é logo um professor

It inferred this and that because of this and that rule. Why did it inferred something.

### Task 5a (1 Point): Property characteristics

Add **four** different property characteristics.

Examples are

* transitivity
* symmetricity
* functionality
* (ir)reflexivity
* property chain
* disjoint properties
* etc

All four different characteristics can be asserted on one property (or on four different properties).

List and describe the four property characteristics you defined (i.e. axioms) in the textfield below. Do not forget to motivate your choices.

First Property characteristic:

* Inverse Functional

The inverse functional characteristic was added on the Object Property isBeingLeadBy. The porpose of this addition was to state if two subjects representing researches are being led by the same leader, we automaticlly know that those different subjects are the same and represent the same research indentity.

Second Property characteristic:

* Symetric

The symetric characteristic was added on the Object Property isColleague. The porpose of this addition was to state that if a someone is a colleague of a third-party that third-party is also a colleague of that someone.

Third Property characteristic:

* Irreflexive

The irreflexive characteristic was added on the Oject Property isColleague. The porpose of this addition was to state that a certain student can never be a colleague of themselves.

Forth Property characteristic:

* Functional

### Task 5b (1 Point): Reasoning over properties

Run the reasoner once again (after having added the four properties).

Write down the steps of the reasoning process (ie, what happens when you run the reasoner). Also write down and explain the resulting inferences below.

Explicar quando as caracteristicas de propriedeades

Your answer here

Mostrar triplos que inferi com as caracteristicas das propriedades

### Task 6 (1-2 Point): Saving your ontology

Go over your ontology to ensure that it is consistent and that it meets all requirements as asked through the various questions. If both criteria are met you will obtain an additional point (making a total of 2 points for this question). Note that there are several [online validators](http://mowl-power.cs.man.ac.uk:8080/validator/) that can help you check your ontology for consistency.

Next, export/save your ontology to a file using Turtle as serialization format. Use save as to ensure that later modifications won't end up in this file.

**Submit this file together with your notebook**

### Task 7 (1 Point): An inconsist ontology

Add one or more axioms to your ontology in such a way that it becomes inconsistent. Note that the added axioms themselves must be consistent; adding a single inconsistent axiom is not allowed.

**IMPORTANT: do not submit this version**

List and describe the axioms that you added. Motivate your choices and explain why the ontology became inconsistent.

Save the good version first

I gotta broke the reasoner. Popularmajor disjoint of major

## Submitting your answers

To submit your answers for this assignment, create a zip-file containing both this notebook and your consistent ontology (saved during task 6). Name this zip-file **assignment\_4\_VUnetID.zip** (where VUnetID is of course to be replaced by your personal VUnetID, eg **rss220**), and submit it via Canvas.