

Open in app ↗

Sign up

Sign in

Medium

Search

Write



Ontology Generation and Visualization with Protégé

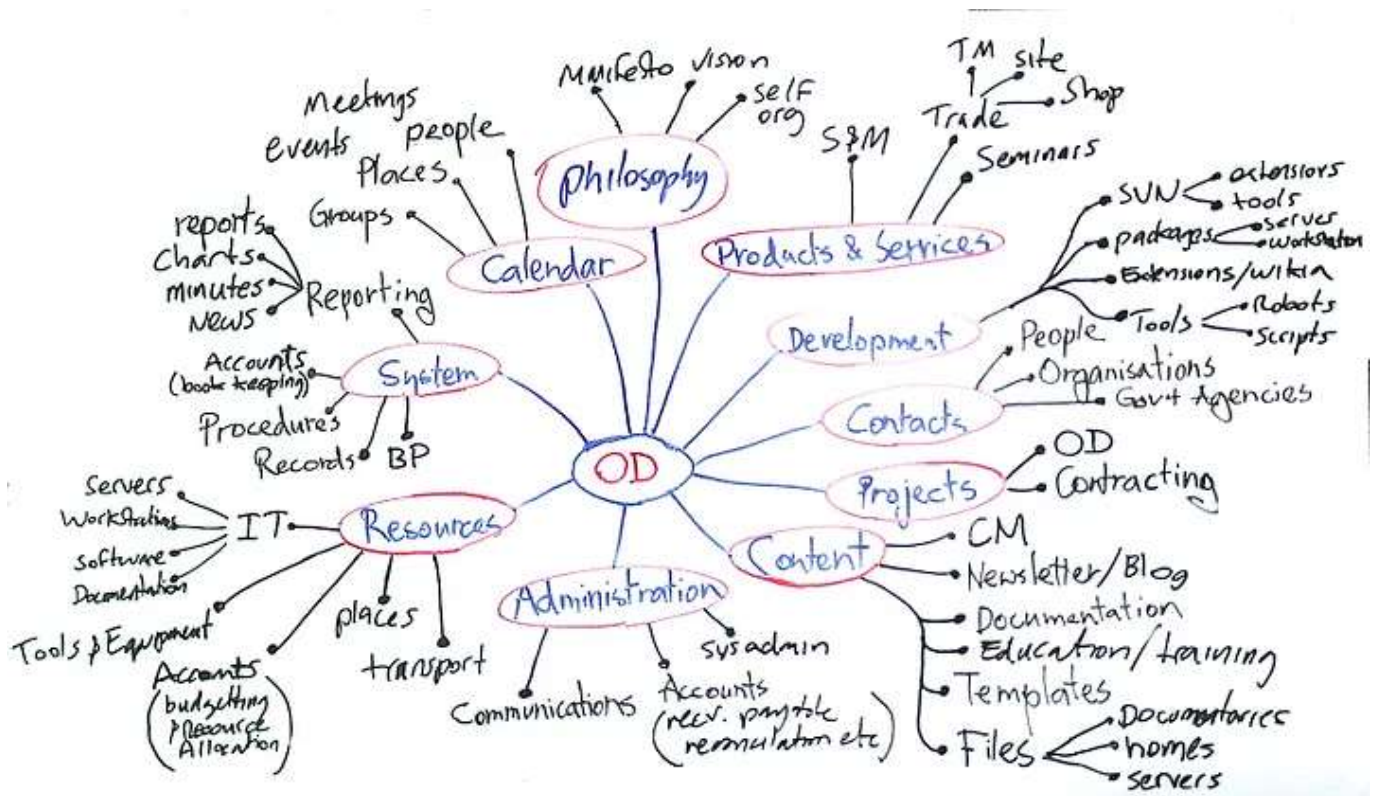


Vindula Jayawardana · Follow

7 min read · Mar 12, 2017

385

2



With a philosophically implied meaning of “subject of existence”, the word “ontology” seems to generate a profusion of controversy in discussions about

AI. It is also often confused with epistemology, which is about knowledge and knowing. However as a formal definition, T. R. Gruber illustrates ontology as a *formal and explicit specification of a shared conceptualization*. That is, an ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents.

In this article, it is expected the reader to have a basic understanding on ontologies and ontology specification. The following sections illustrate the use of Protégé editor to generate and visualize an ontology with basic ontology definition concepts being covered. We will be looking in to creating a simple university ontology and later visualize it within Protégé editor through the use of VOWL Plugin.

Following diagram indicates the final ontological hierarchy that will be built at the end of this article.

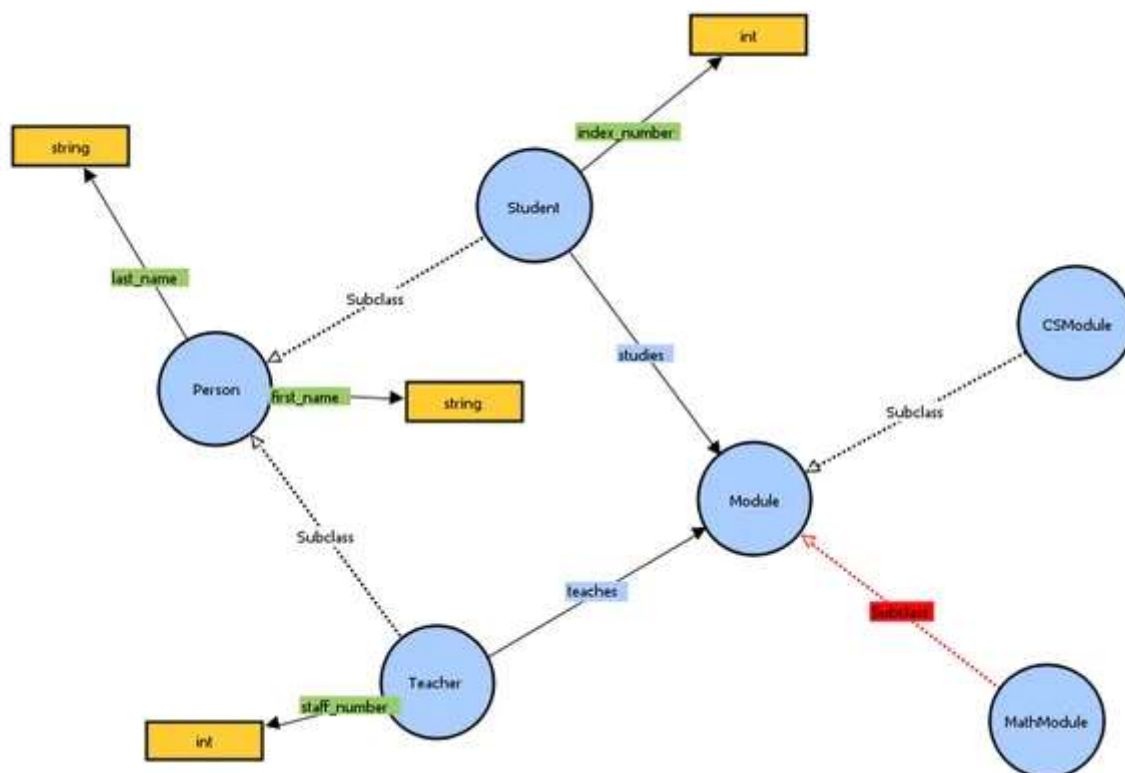


Figure 1: Final Ontology view

What is Protégé and how to set it up

Protégé is a free, open source ontology editor and framework developed by Stanford University for building intelligent systems . Protégé is supported by a strong community of academic, government, and corporate users, who use Protégé to build knowledge-based solutions in areas as diverse as bio-medicine, e-commerce, and organizational modeling.

Follow the below steps to get Protégé up and running.

1. Download Protégé from [here](#) and extract it.
2. Go inside the extracted folder and click protege.exe to run Protégé.

Creation of Ontology

The next step is to create your ontology in the Protégé. To do that, click on the “*Entities*” tab. When clicked, by default you will be put inside the “*Classes*” sub tab.

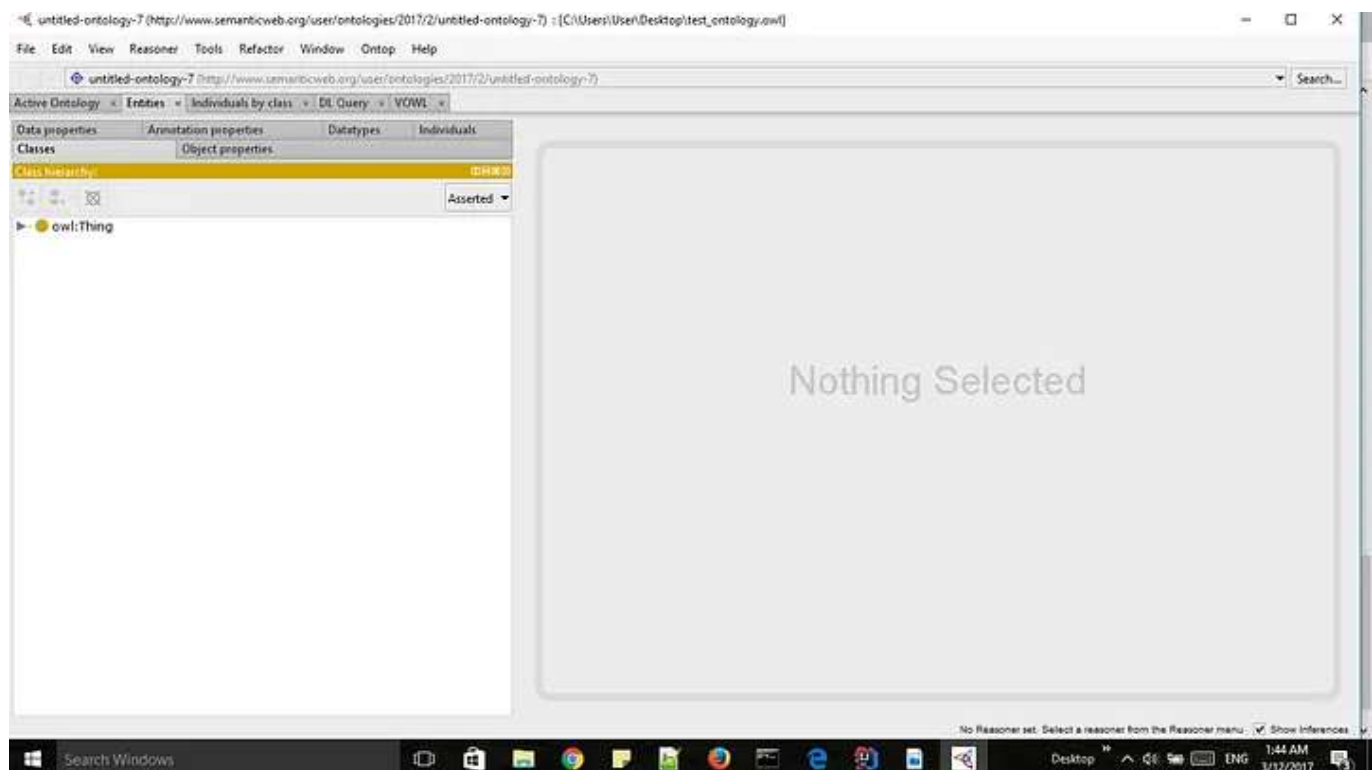


Figure 2: Entities tab view

In this example, we will be looking into creating an ontology based on very basic concepts/scenarios in a university.

1. *Teachers teach modules*

2. *Students study modules.*

With those two concepts in hand, following classes, data properties, object properties and individuals could be identified.

Classes

Classes in ontologies are abstract groups, sets or collections of objects.

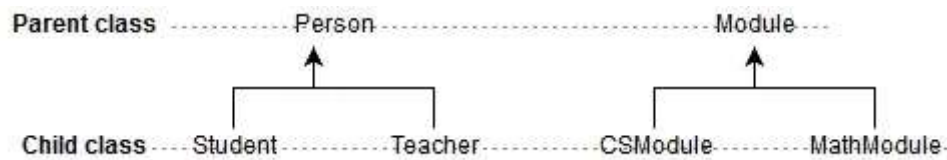
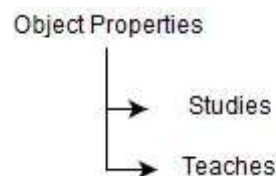


Figure 3 : Classes

Object Properties

In object properties we are concerned with describing how classes can relate to each other based on their instances.



Figures 4 : Object Properties

Data Properties

Data properties are just like object properties except their domains are typed literals.

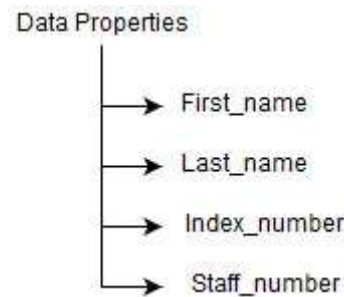


Figure 5 : Data Properties

Individuals

Individuals (instances) are the basic, “ground level” components of an ontology.

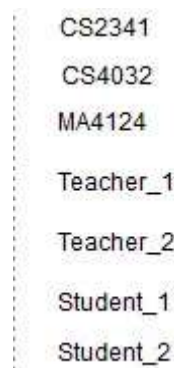


Figure 6: Individuals

For the simplicity purpose, certain data properties are omitted by intention.

Next step involves creating the identified classes, data properties, object properties in the ontology.

Creating Classes

1. Click on the classes tab and open the class hierarchy.
2. Note that all classes identified above are sub classes of “*Thing*” class by default.
3. Click on “*Add subclass*” icon and in the pop up menu, give the name of the class.

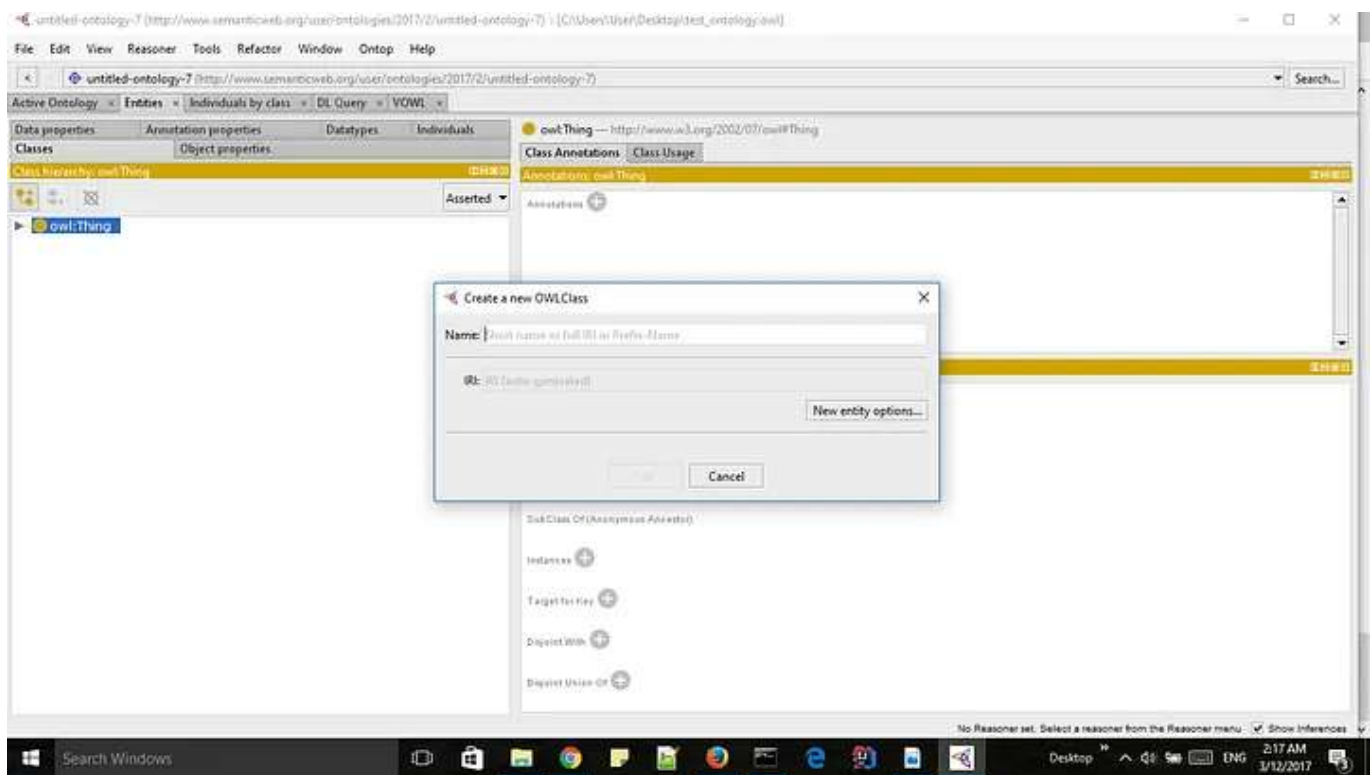


Figure 7 : Create class pop up

Remember to follow the class taxonomy identified in Figure 3 in creating the classes here. For an example, Class “*Teacher*” should be a sub class of “*Person*” class where “*Person*” class is again a subclass of “*Thing*” class. After creating class taxonomy, you should have the following class hierarchy defined.

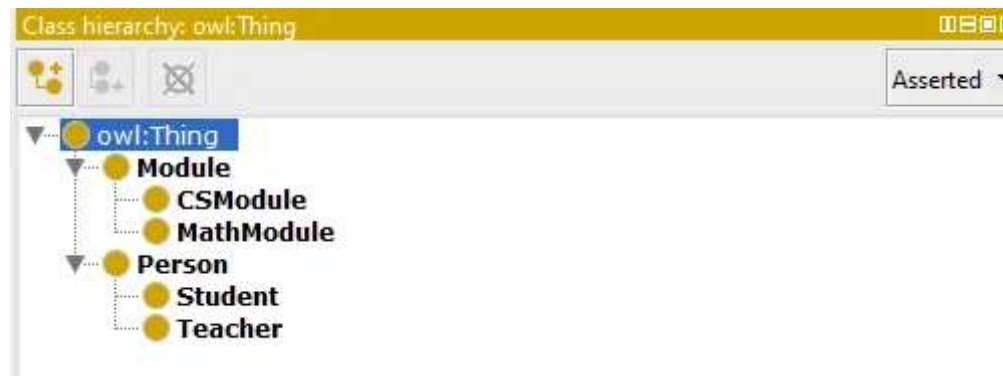


Figure 8 : Class hierarchy

Creating Object Properties

To create the identified object properties in the Figure 4, select the Object Properties tab and click on “Add sub property” icon. Note that all properties will be sub properties of the “*topObjectProperty*” by default.

Create a new OWLObjectProperty

Name:

IRI:

Figure 9 : Add new object property

However it should be noted that we can add domain and range to these object properties.

Syntactically, domain is a built-in property that links a property to a class description. An domain axiom asserts that the subjects of such property statements must belong to the class extension of the indicated class description.

Range is a built-in property that links a property to either a class description or a data range. An range axiom asserts that the values of this property must belong to the class extension of the class description or to data values in the specified data range.

Hence for the object property “*studies*” we can add “*Student*” as the domain and “*Module*” as the range.

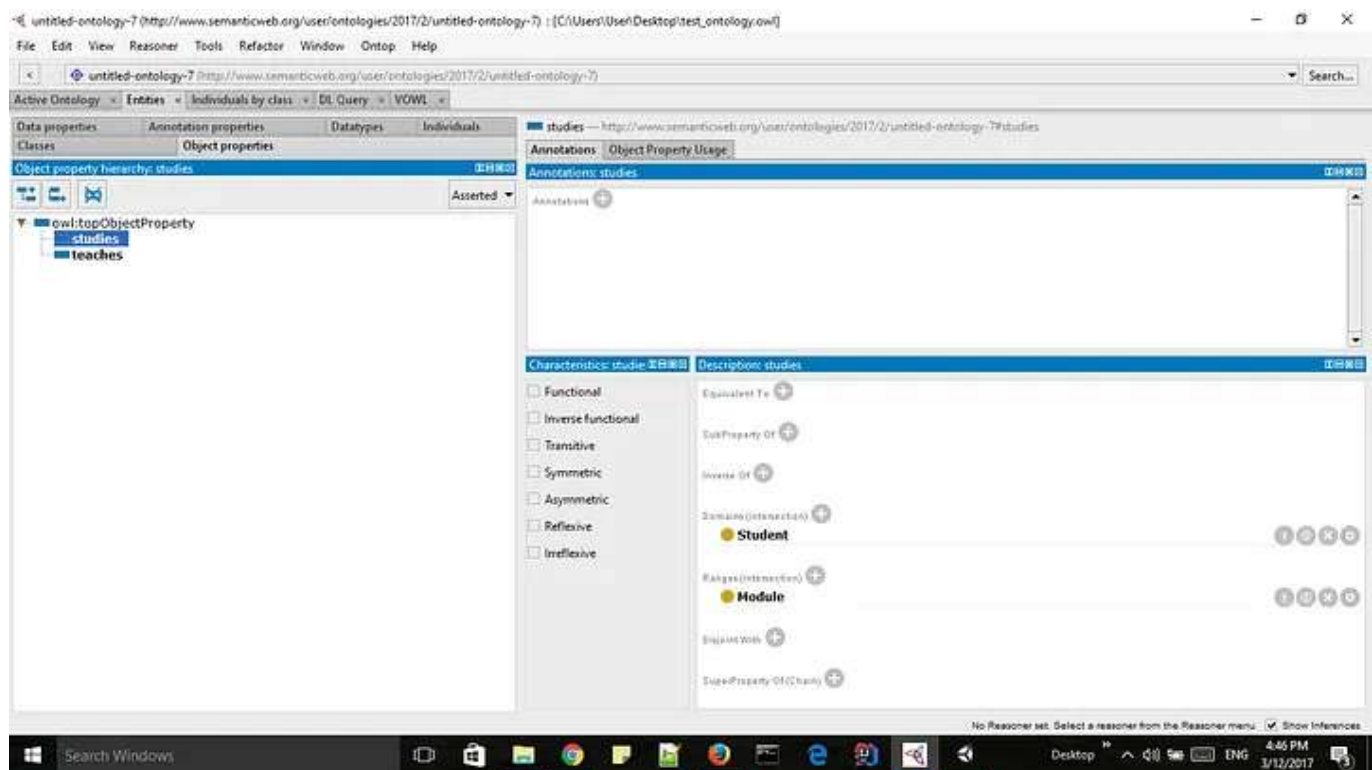


Figure 10 : Add domain and range for object property “studies”

To add domain or range, simply click on the + button near the required item and select the items you want to add.

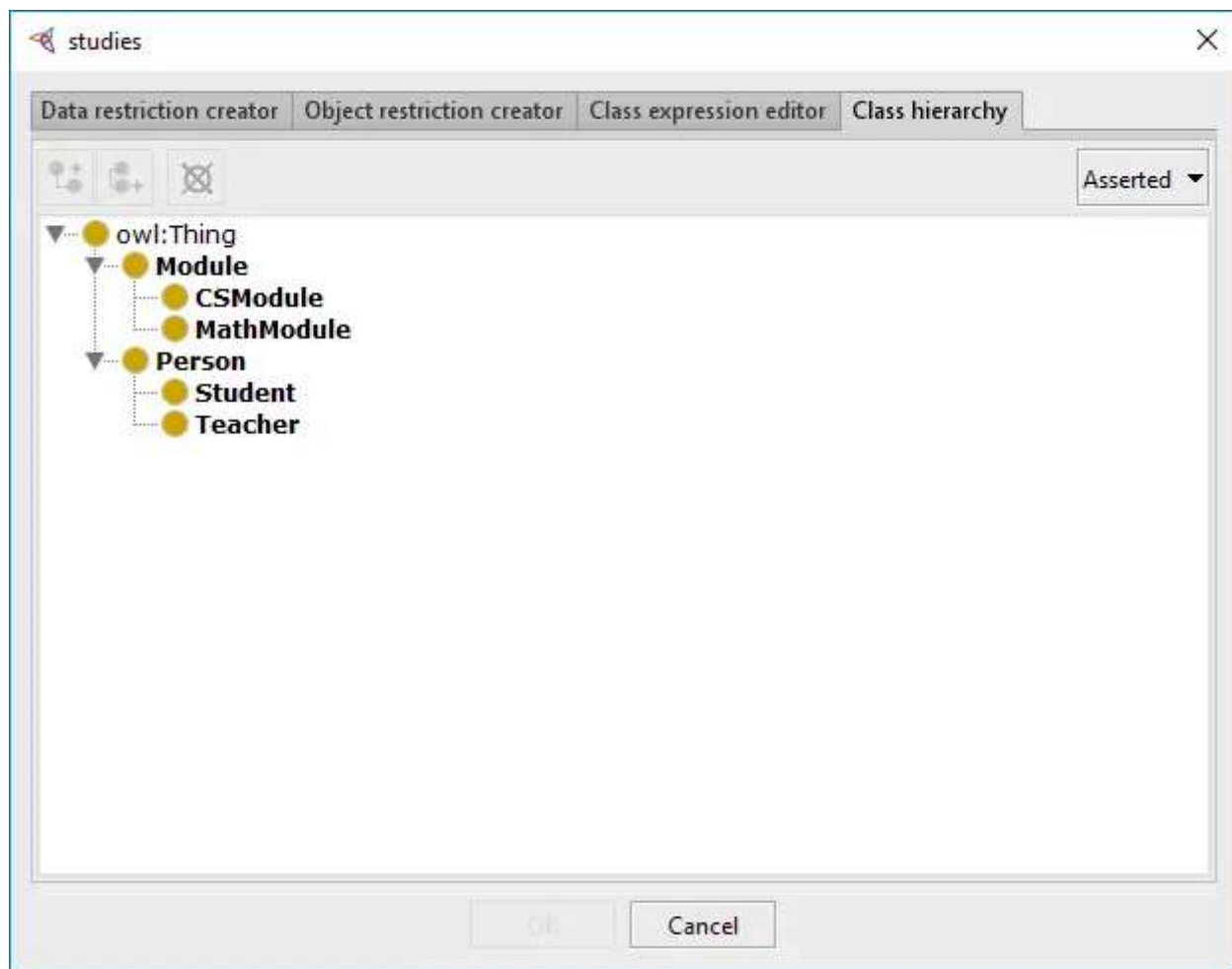


Figure 11 : Select domain or range for “studies” object property

In the same way, for the object property “*teaches*” we can add “*Teacher*” as the domain and “*Module*” as the range.

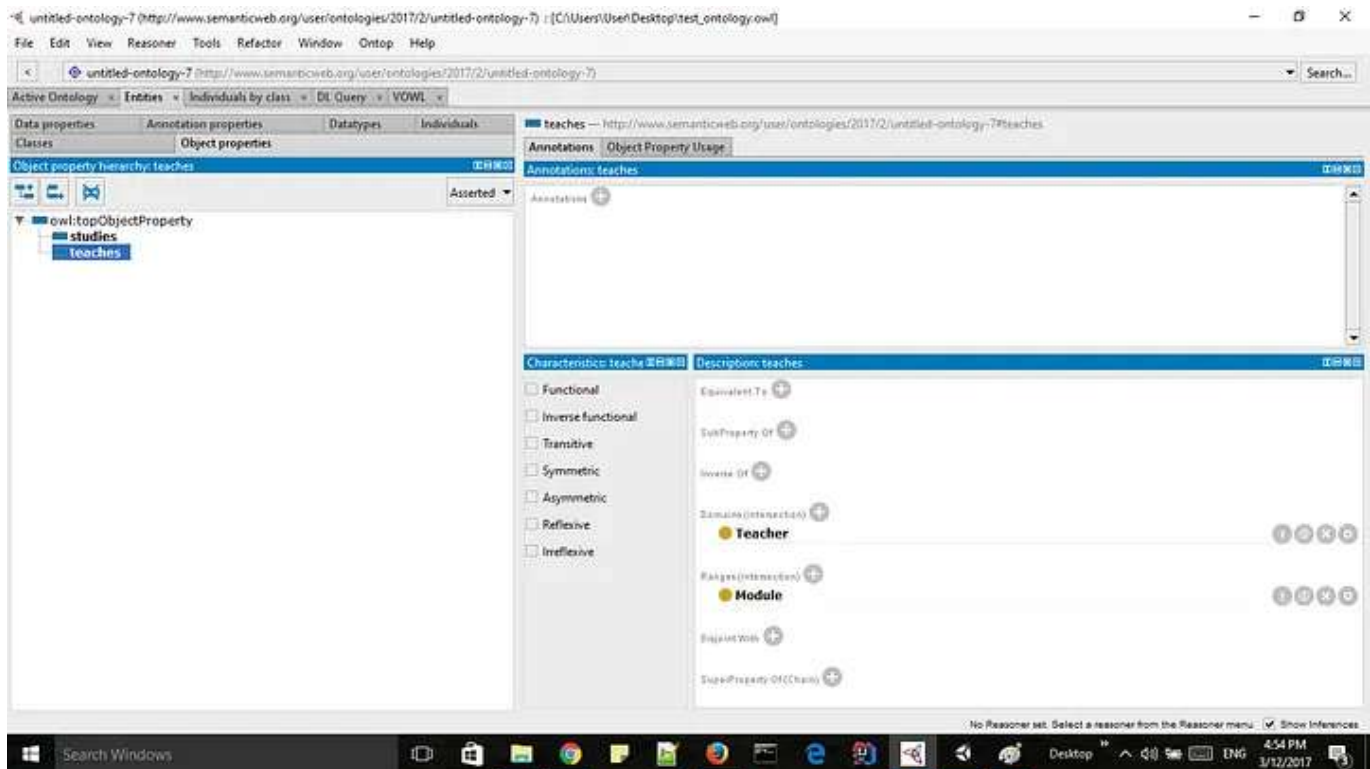


Figure 12 : Add domain and range for object property “teaches”

Creating Data Properties

The next step is to create the identified data properties in Figure 5. To do this, simply click on the “*Data Properties*” tab and click “*Add sub property*” icon. Note that all properties will be sub properties of the “*topDataProperty*” by default.

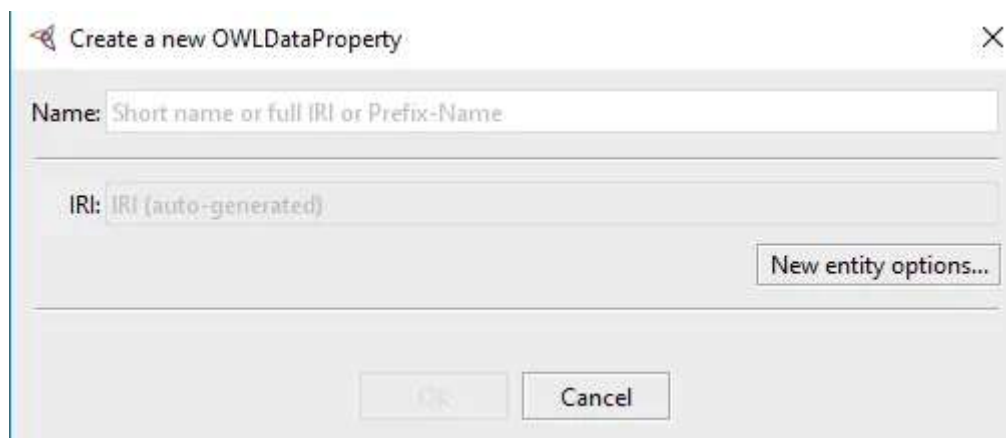


Figure 13 : Add new data property

After adding all the data properties identified, you should get the following overview.

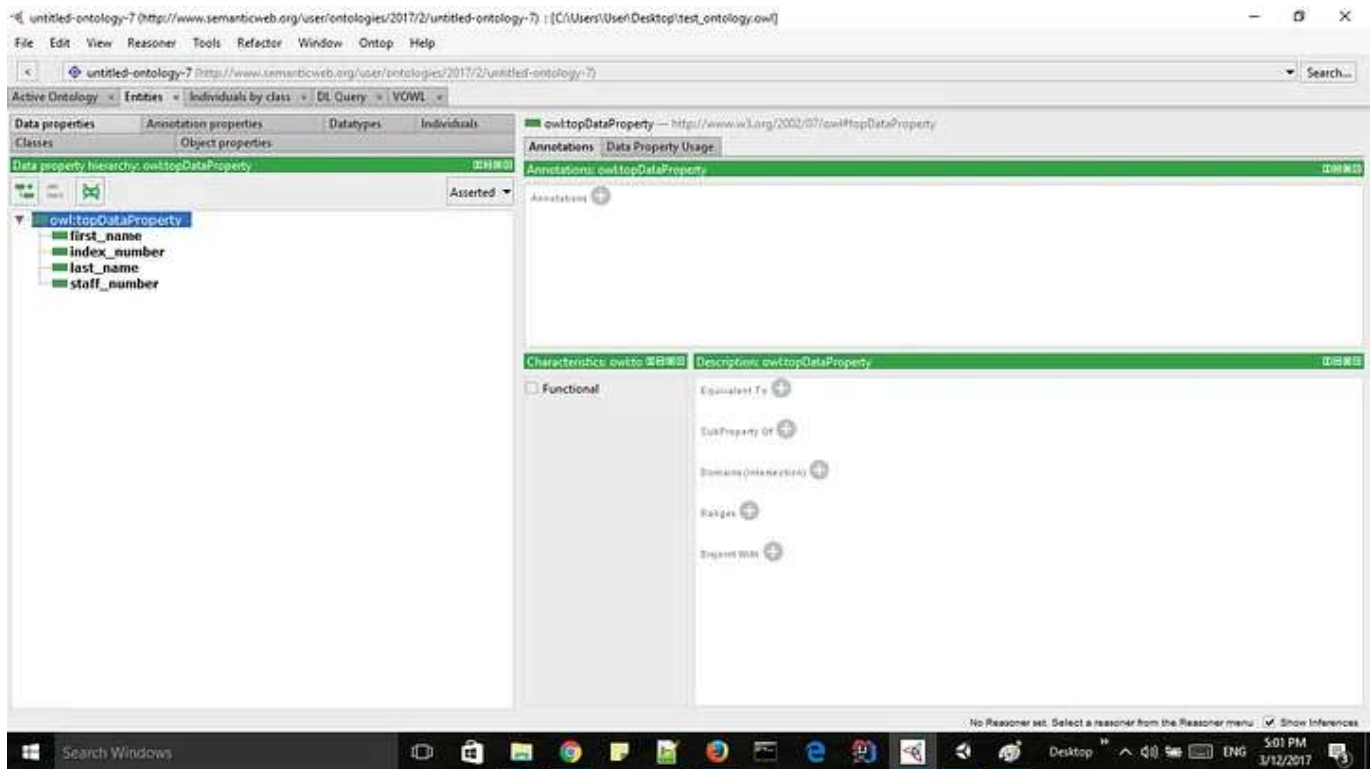


Figure 14 : Data property final overview

In the same way, we did with object properties, we can define domain and range for each data property as well. However, in the range selection, rather having a class description, need to add a data types. For an example,

“first_name” data property has the range of *“xsd:string”*.

To add this data types click on the + button near *“Range”* and select *“Built in data types”* and select the relevant data type from the given list.

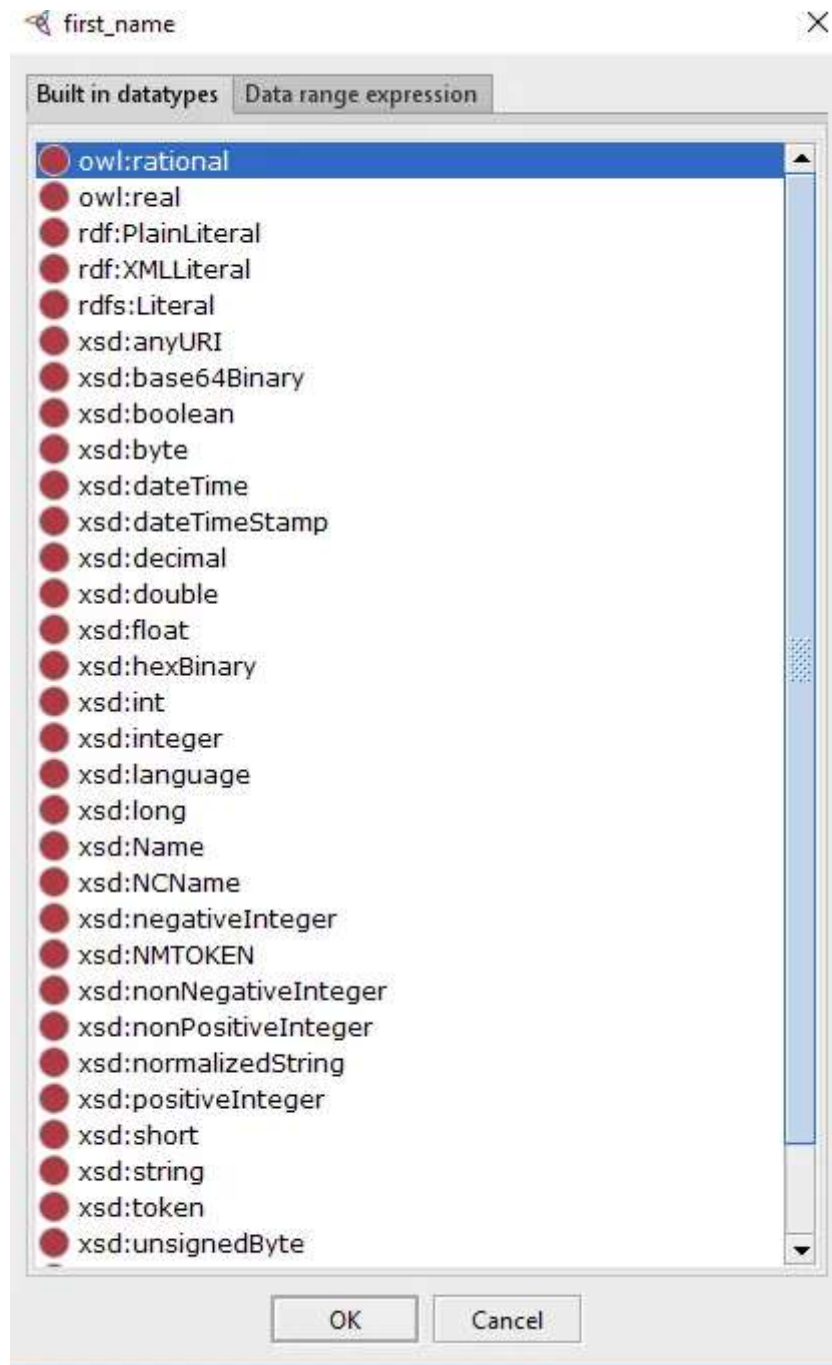


Figure 15 : Select the range for first_name

Add the following ranges to the other identified data properties as well.

first_name → xsd:string

last_name → xsd:string

index_no → xsd:int

staff_id → xsd:int

Adding Individuals

The final task in creating the ontology is adding individuals. To do this, click on the “*individuals*” tab and click on the “*Add individual*” icon.

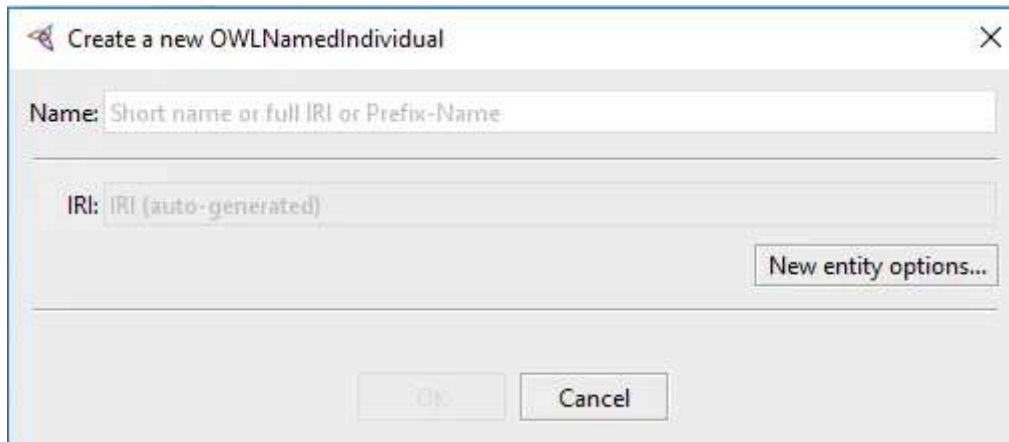


Figure 16 : Add new individual

First, let's create a “*Module*” individual. Follow the below steps to set the individual properly.

1. Create the “*Module*” individual “*CS2341*”.
2. Add the Type for the “*CS2341*” as “*Module*”.

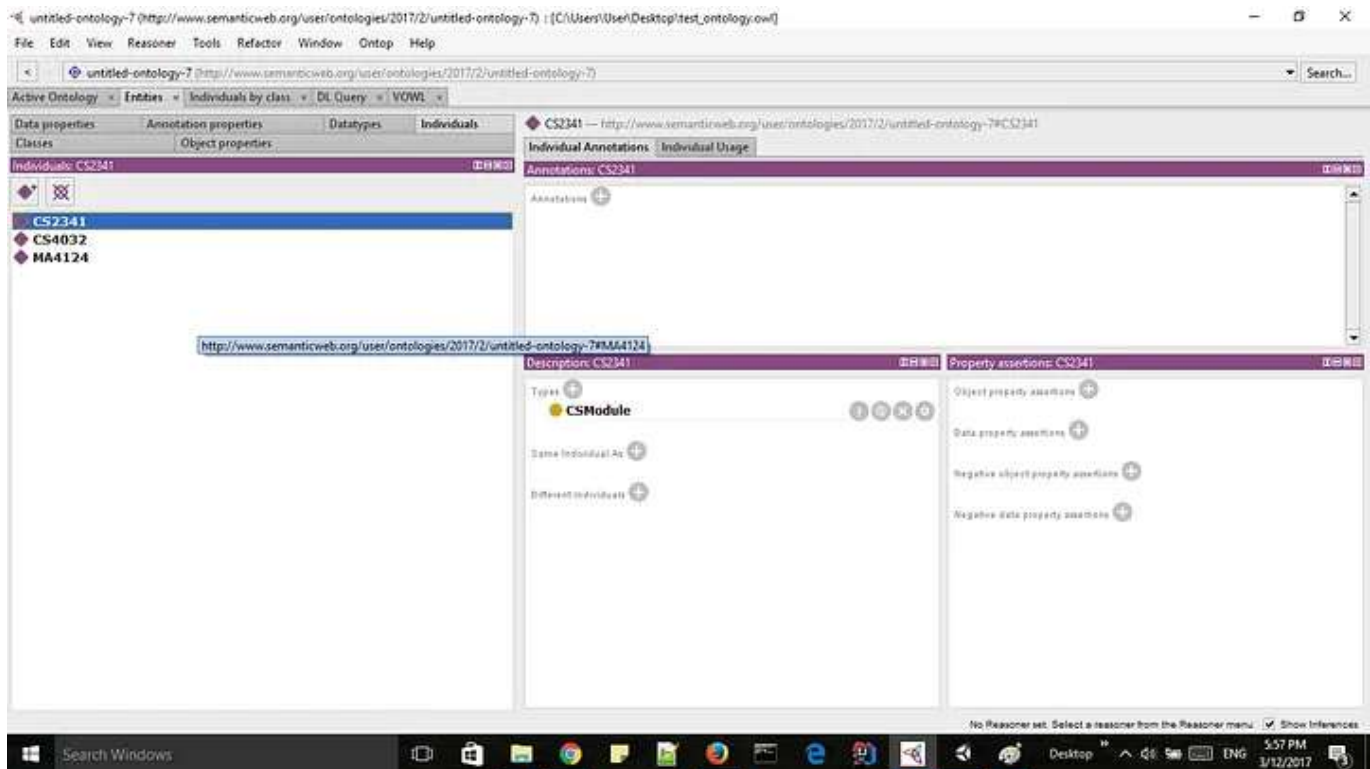


Figure 17 : Define “CS2341” module individual

Do the same for other two modules “CS4032” and “MA4124”.

Secondly, let’s add individual “*Student_1*”. Follow the below steps to set the individual properly.

1. Create the individual “*Student_1*”.
2. Add the Type for the “*Student_1*” as “*Student*”.
3. Add the object properties for “*Student_1*”.

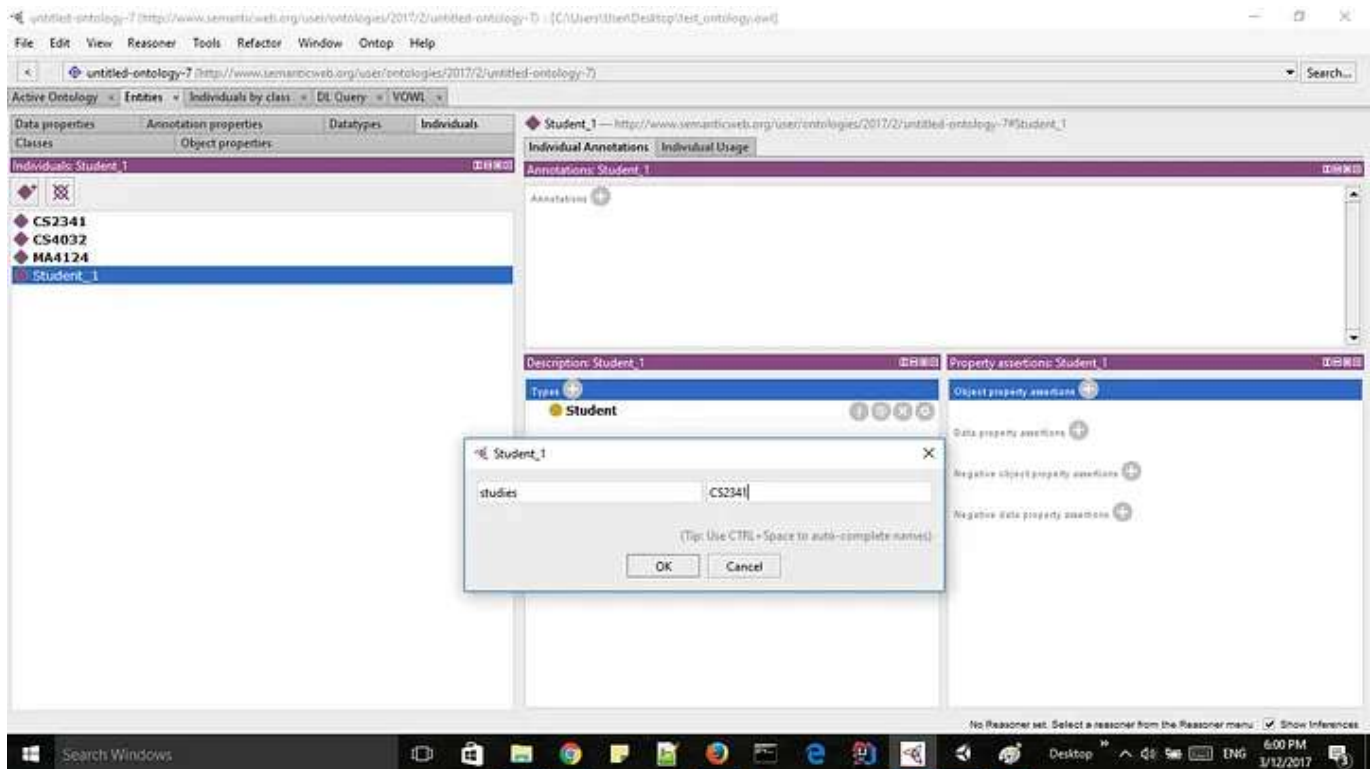


Figure 18 : Adding object properties to “Student_1”

4. Add the data properties to the “Student_1”. Refer below image for more details.

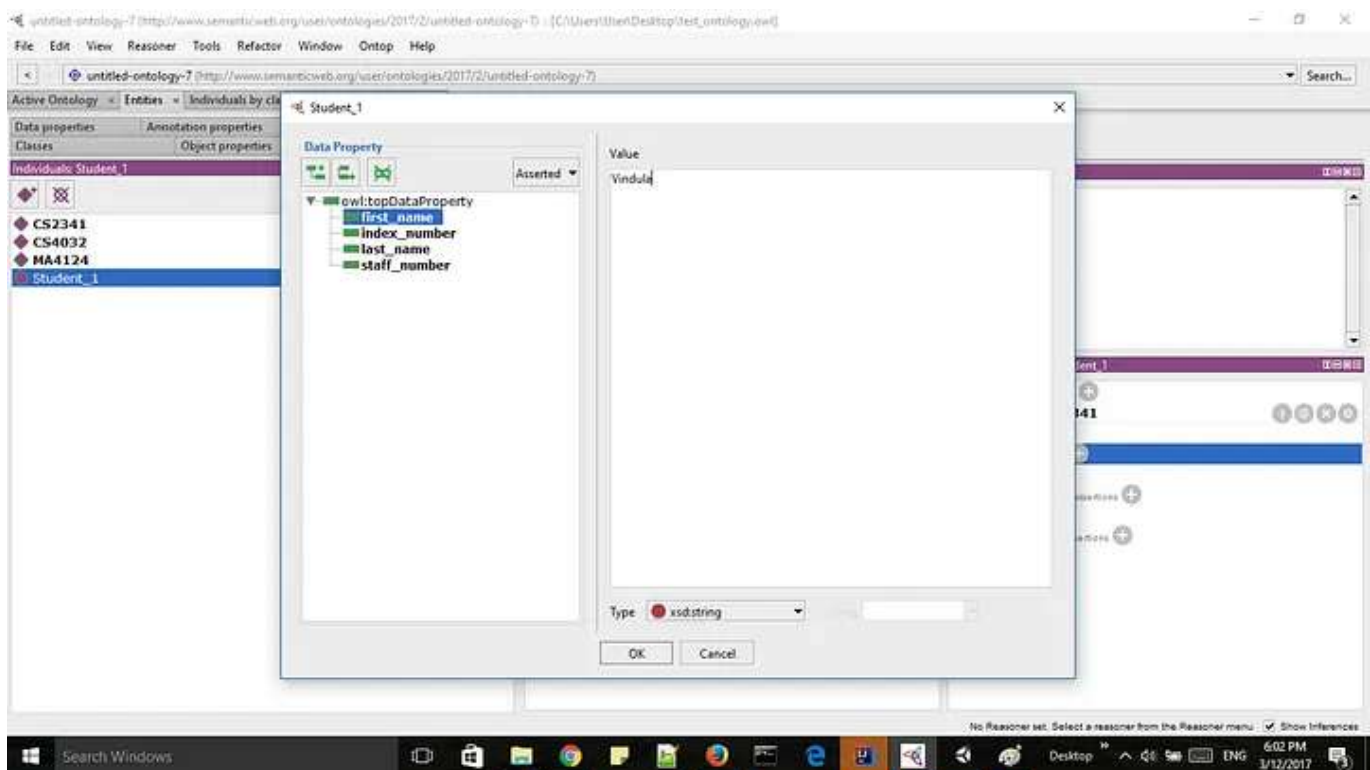


Figure 19 : Add first_name data property to Student_1

First select the “*first_name*” data property and then select type as “*xsd:string*”. Finally add the the value in the space provided (in the above example value = *Vindula*). Do the same for other data properties as well.

After finishing the above iterative process you should get the “*Student_1*” individual as below.

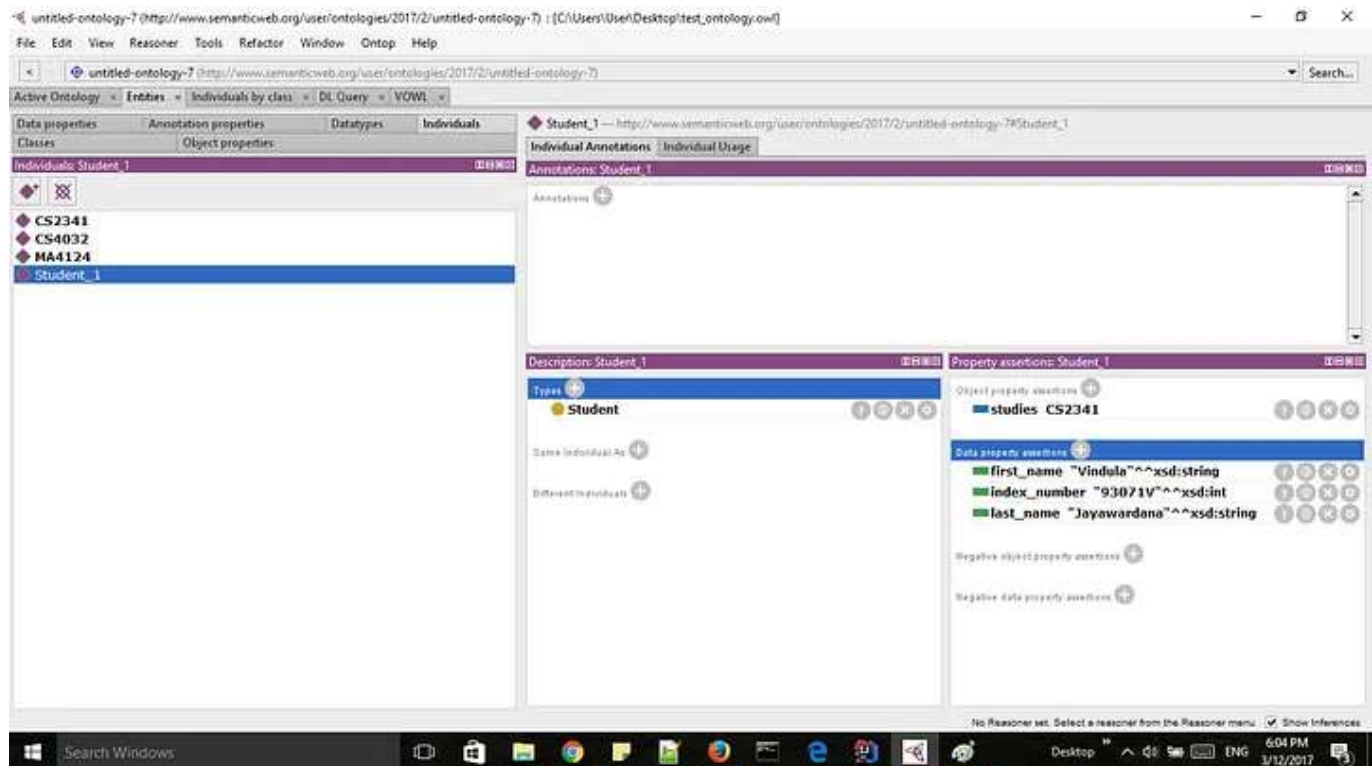


Figure 20 : Final “Student_1” setting

In the same way create the other individuals identified in Figure 6.

If you have completed the above steps correctly, Congratulations!! you just created the university ontology. Now the next step is visualization of the created ontology. Alternatively you could download the created ontology as an OWL file by File → Save As → OWL/XML Syntax.

Visualization of the Ontology

To visualize the ontology in the graphical manner, we need to install the *ProtégéVOWL* plugin. You can download the jar file from [here](#). After downloading the VOWL jar, copy it to the *plugins* folder of Protégé, re-start Protégé and activate the plugin via *Window* → *Tabs* → *VOWL*.

That's it. Simply click the “VOWL” tab in the Protégé and you could see the ontology created in graphical manner as below.

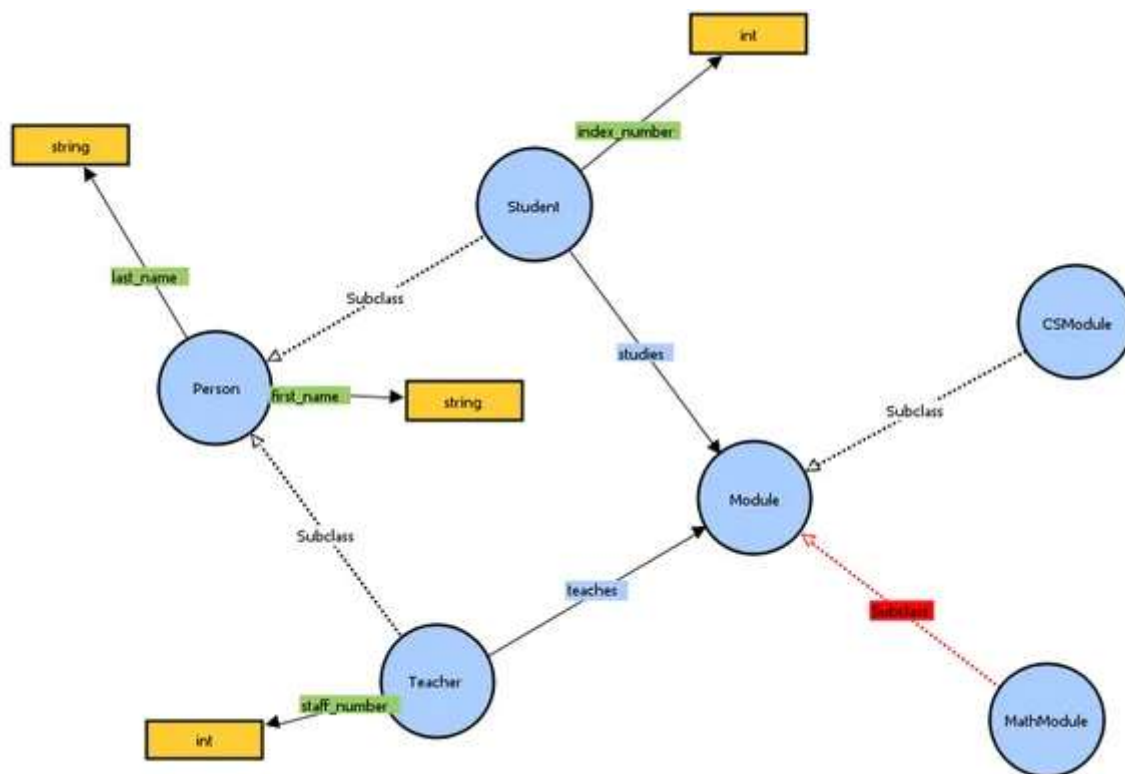


Figure 21 : Final Ontology view

Semanticweb

Ontology

Protege



Written by Vindula Jayawardana

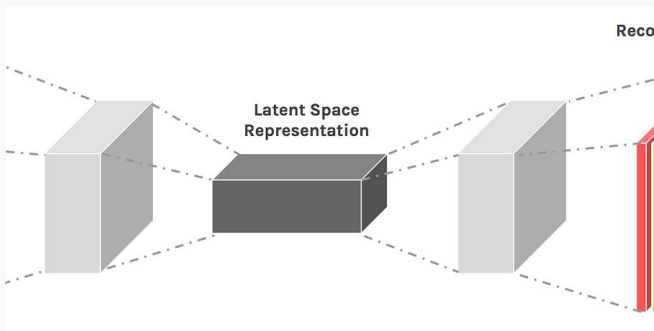
364 Followers

PhD Student at MIT | www.vindulaj.com

Follow



More from Vindula Jayawardana



Vindula Jayawardana in Towards Data Science

Autoencoders — Bits and Bytes of Deep Learning

One way to think of what deep learning does is as “A to B mappings,” says Andrew Ng, chi...

5 min read · Aug 4, 2017



652



2



Vindula Jayawardana in Towards Data Science

Adaptive Authentication And Machine Learning

“We use an analogy: the front door of your house has a handle that is easily turned to g...

3 min read · Apr 20, 2017



120



3





Vindula Jayawardana

SCIM 2.0 Compliance Test Suite — GSoC 2017 with WSO2

This post is on my Google Summer of Code 2017 project as an open source contributor o...

4 min read · Aug 27, 2017



96



3



Vindula Jayawardana

SCIM 2.0 API for WSO2 Identity Server 5.3.0

The SCIM API can be called in order to perform identity provisioning in the WSO2...

4 min read · Dec 22, 2016



58

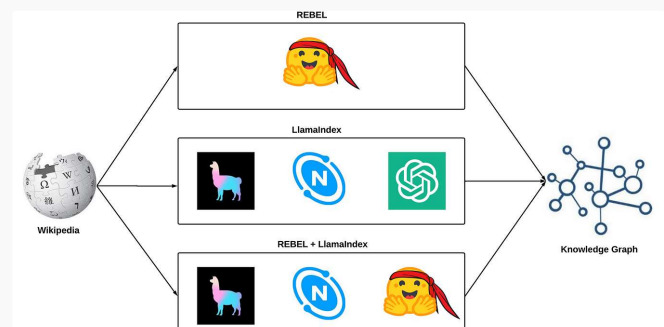
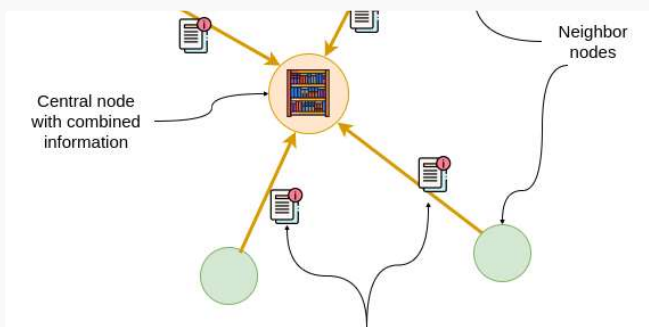


5



See all from Vindula Jayawardana

Recommended from Medium





Tomaz Bratanic in Towards Data Science

Integrate LLM workflows with Knowledge Graph using Neo4j an...

OpenAI and VertexAI endpoints are now available as APOC Extended procedures

8 min read · Jun 7



285



2



Saurav Joshi

Building Knowledge Graphs: REBEL, LlamaIndex, and REBEL +...

Exploring building knowledge graphs using LlamaIndex and NebulaGraph

10 min read · Oct 3



85



1



Lists



Staff Picks

507 stories · 456 saves



Stories to Help You Level-Up at Work

19 stories · 308 saves



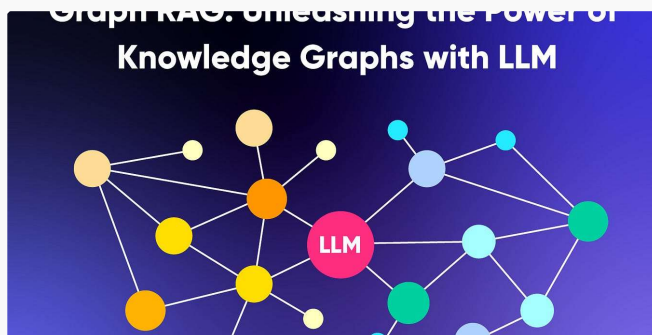
Self-Improvement 101

20 stories · 904 saves



Productivity 101

20 stories · 826 saves

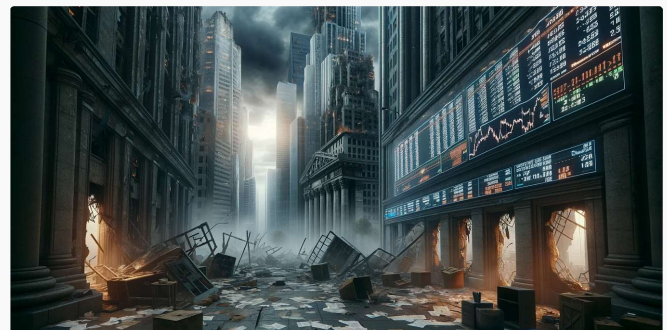


NebulaGraph Database

Graph RAG: Unleashing the Power of Knowledge Graphs with LLM

In the era of information overload, sifting through vast amounts of data to provide...

6 min read · Sep 8



Ignacio de Gregorio

OpenAI Just Killed an Entire Market in 45 Minutes

The Story Everyone Should Have Seen Coming



6 min read · Nov 9



133



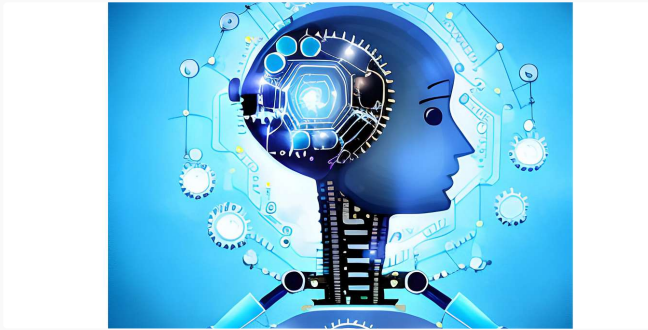
1



12K



168



Sruthi Radhakrishnan

Navigating AI's Reality: The Role of KGs and LLMs in Fact-Checking

"Ever wondered how we can catch our AGI's 'hallucinations' and keep them grounded in..."

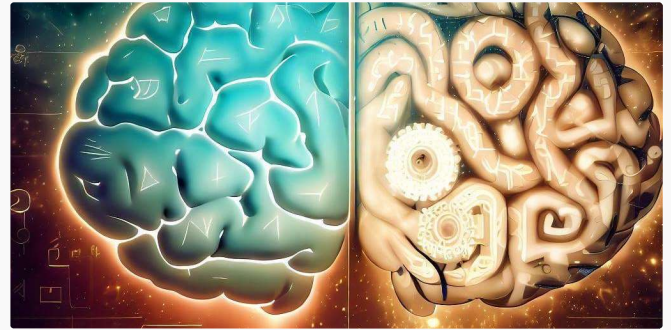
3 min read · Jul 18



66



1



Anthony Alcaraz in AI Mind

Integrating Symbolic and Neural Knowledge: A Unified Framework...

★ · 5 min read · Jul 26

See more recommendations