Information engineering

Documentation, backgrounders and tutorial material related to information design, engineering, semantics, ontologies, and vocabularies

→ Standards and namespaces
→ Controlled vocabularies
→ Semantic Web Tools
→ Learning resources

Tutorial: Introduction to RDF and OWL

A tutorial to introduce RDF and OWL concepts using the TopBraid Composer (Free) Editor.

In this tutorial, we will learn to:

1. Build a very simple Pizza ontology from scratch
2. Import an ontology into the Topbraid editor
3. Begin querying data using RDF and OWL

1. Learning Objectives

Attendees will:

- Understand some of the RDF and OWL language elements, and their explicit semantics
- Learn basic elements of information modelling using OWL
- Gain hands-on experience with ontology development and querying using Topbraid Composer tools
- Learn the facets of the SPARQL language and how to query using SPARQL

2. Pre-requisites and assumptions

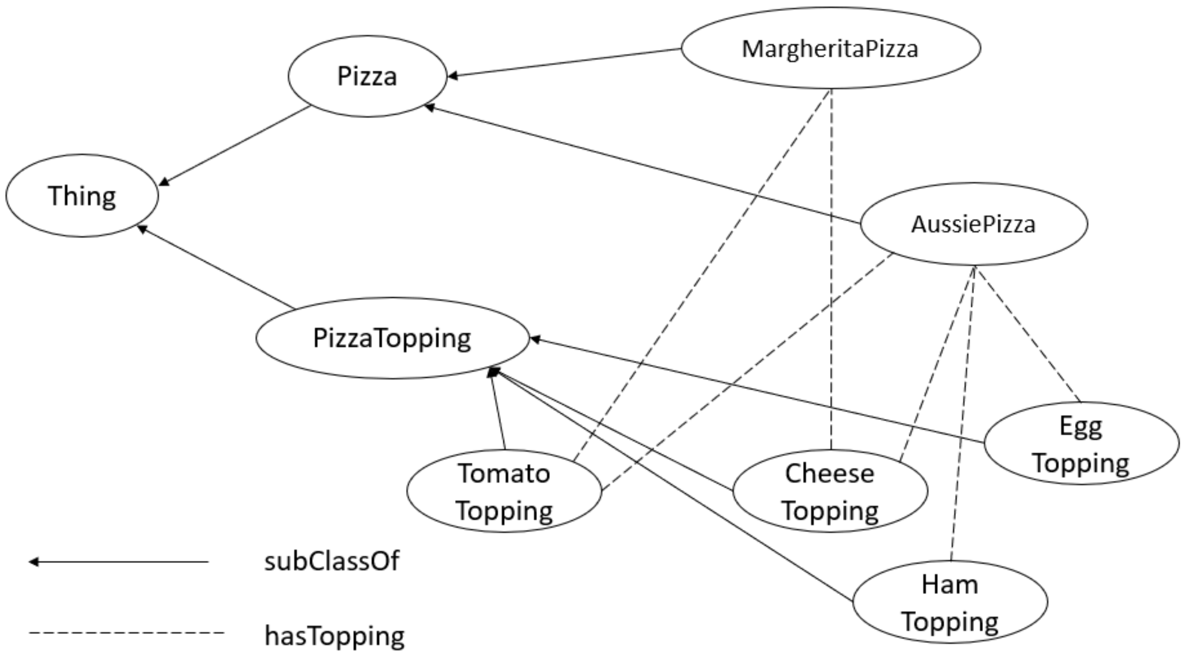
- Topbraid Composer Free installed
- A little familiarity with OWL and RDF concepts

3. Part 1. Creating a simple Pizza ontology using RDF and OWL (15-20mins)

Objective: Creating a simple Pizza ontology in RDF and OWL using Topbraid Composer

In this part of the tutorial, we will be creating a simple Pizza ontology. Before we start using the Topbraid Composer tool, we'll need an information model to guide us.

The figure below shows a suggested information model for a simple Pizza ontology. We have 2 main classes, called `Pizza` and `PizzaTopping` and we'll be creating 2 pizza types - the classic, `MargheritaPizza` (which, according to legend, in 1889 was created and named after her…) and the `AussiePizza` (a local favourite).



3.1. Exercise 1. Building your simple Pizza ontology

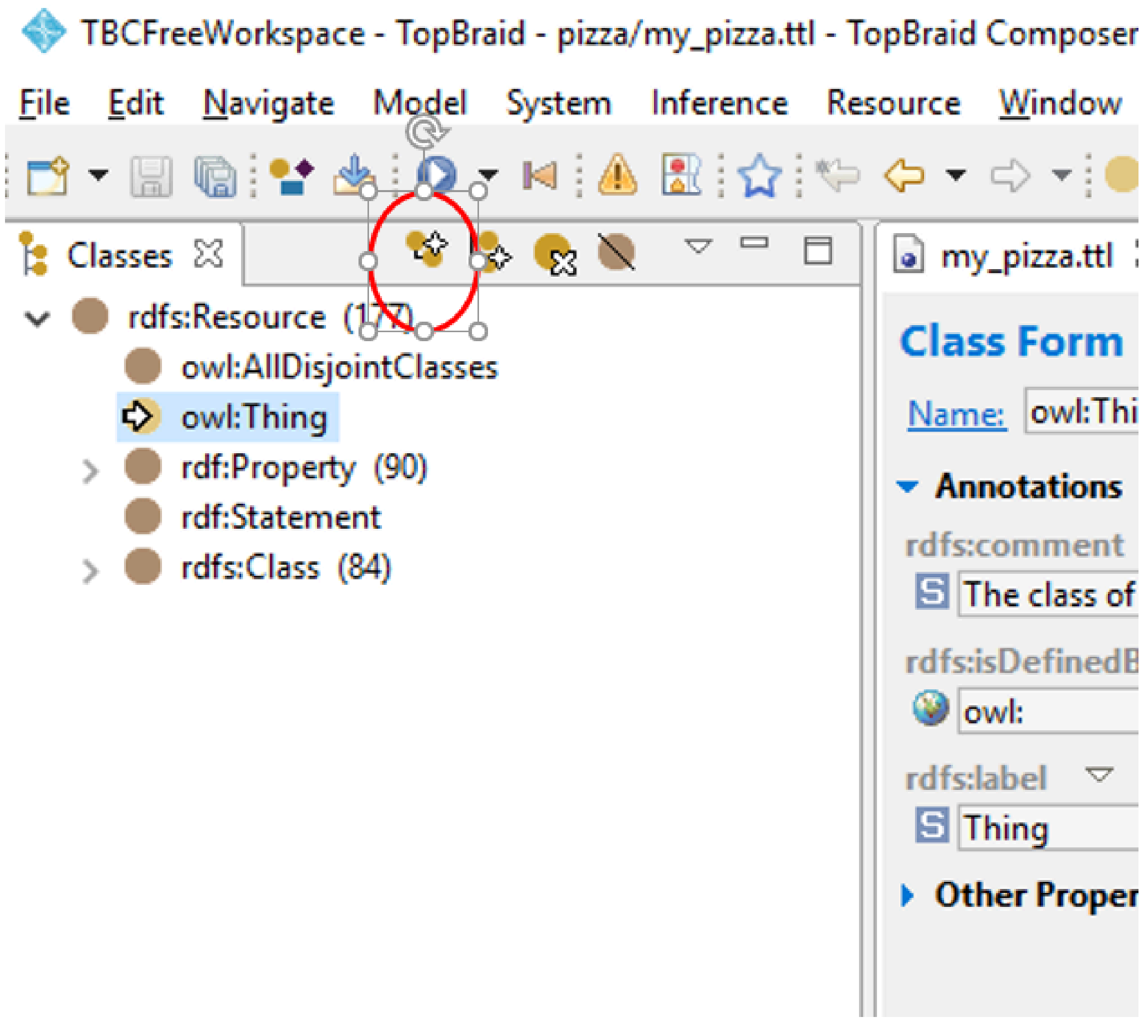
3.1.1. Create your first class: Pizza class

Fire up Topbraid Composer. We'll be creating:

- A new project, called "Pizza"
- A new RDF file, called "mypizza"

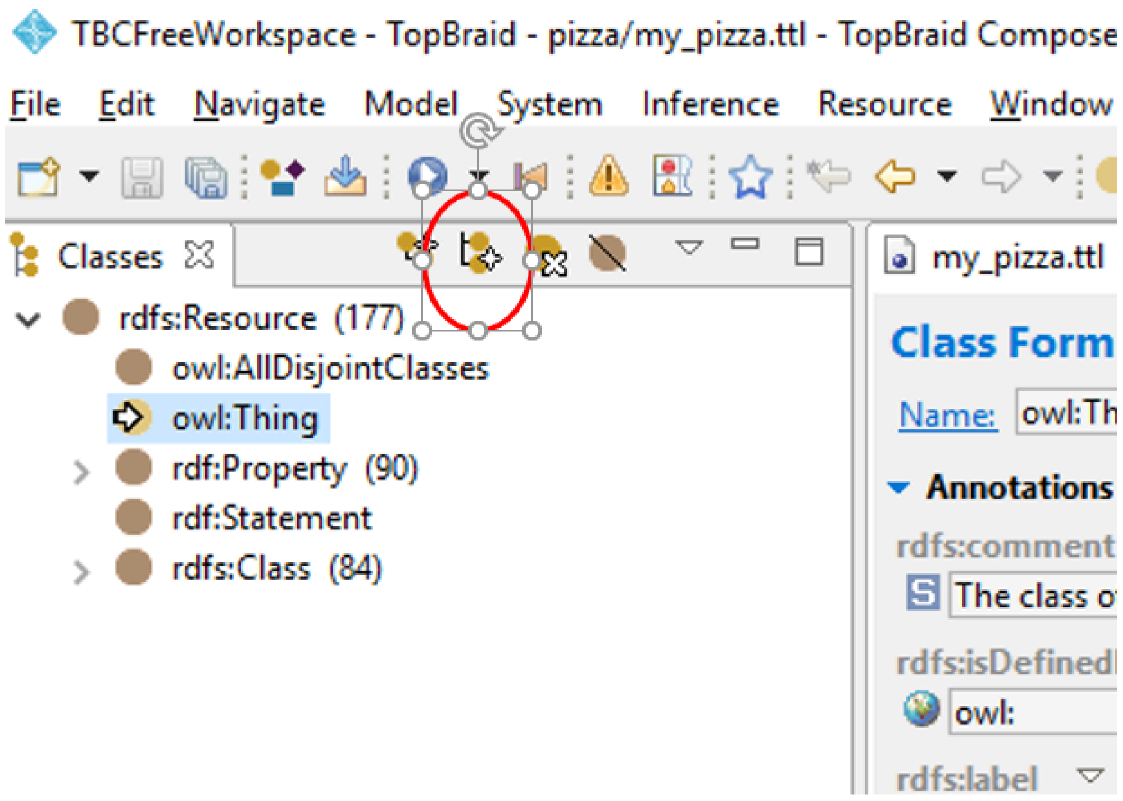
In your new RDF 'mypizza' file, create a new class.

Click on the existing class `owl:Thing` and click on the "Create subclass" button. We subclass `owl:Thing` because in OWL, every user-defined class is a subclass of `owl:Thing`.

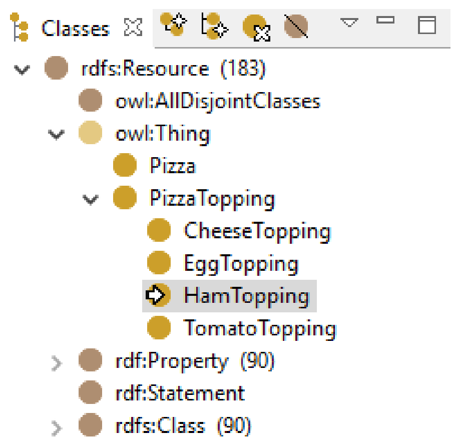


3.1.2. Create more classes

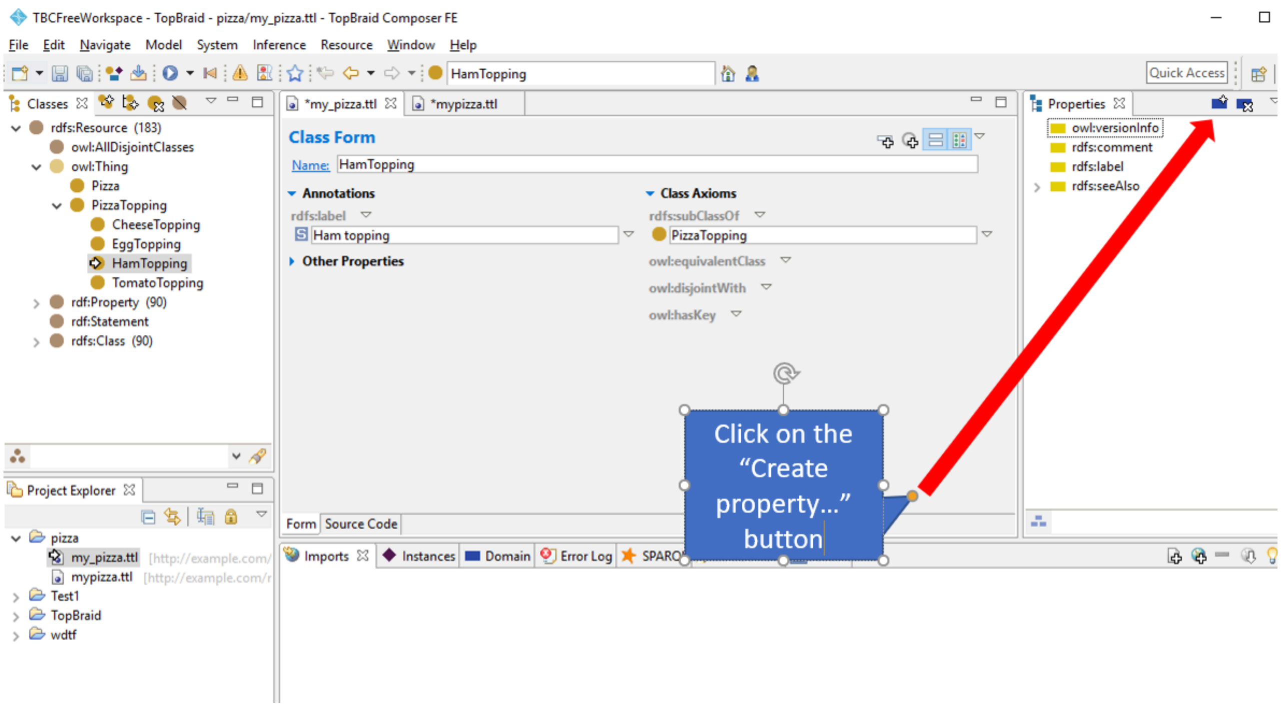
Create a sibling class called `PizzaTopping` and the remaining topping subclasses, e.g. CheeseTopping, HamTopping, TomatoTopping, EggTopping



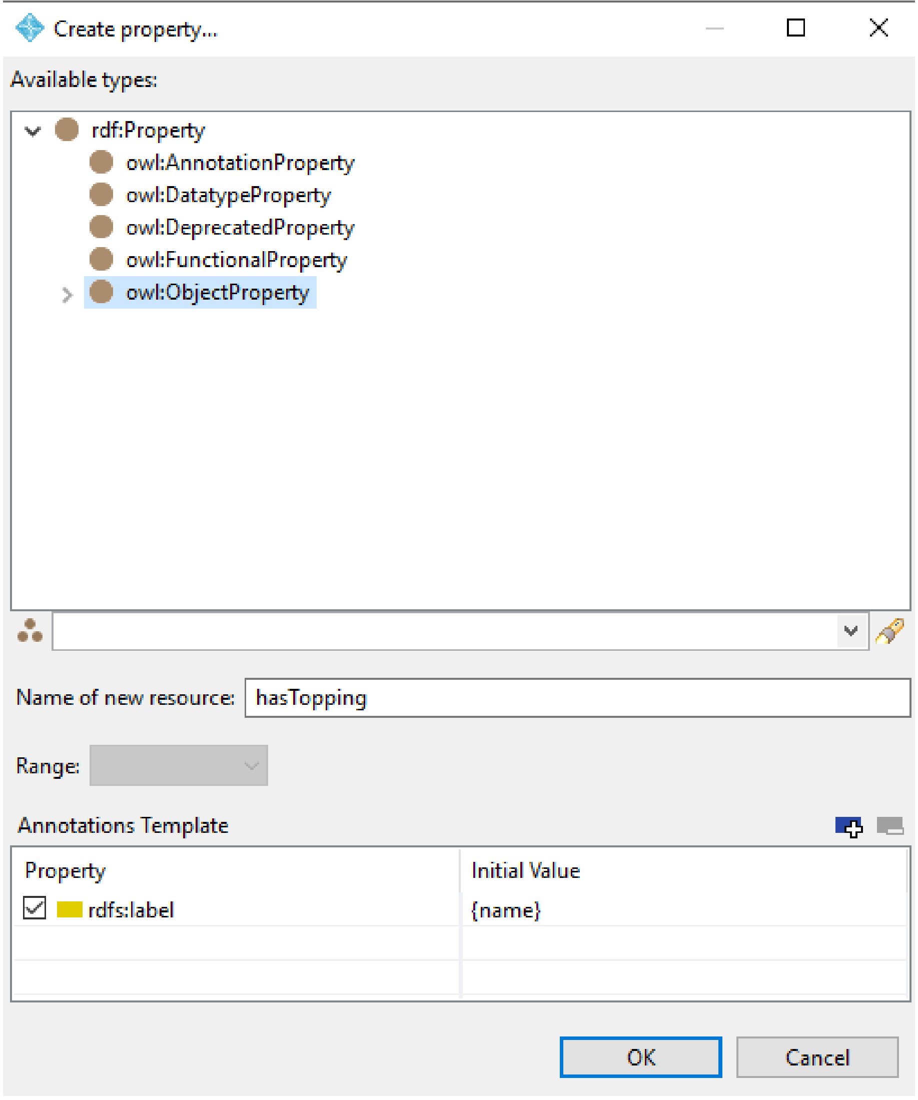
It should look like this:


Classes
rdfs:Resource (183)
owl:AllDisjointClasses
owl:Thing
Pizza
PizzaTopping
CheeseTopping
EggTopping
HamTopping
TomatoTopping
rdf:Property (90)
rdf:Statement
rdfs:Class (90)

3.1.3. Create hasTopping object property Pizza subclasses



Select `owl:ObjectProperty` in the in the `Create property` panel. Add the name of the property in the text field, i.e. hasTopping.


Create property...

Available types:

- rdf:Property
  - owl:AnnotationProperty
  - owl:DatatypeProperty
  - owl:DeprecatedProperty
  - owl:FunctionalProperty
  - owl:ObjectProperty

Name of new resource: hasTopping

Range:

Annotations Template

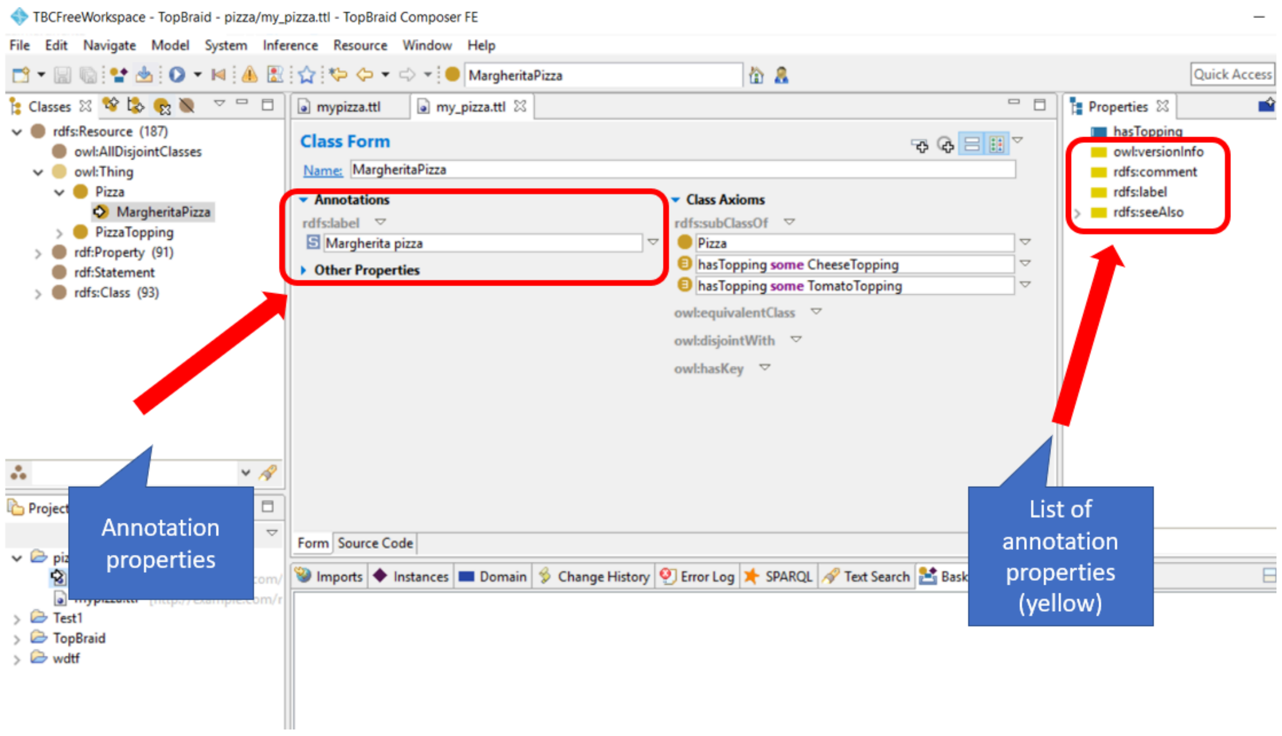
Property	Initial Value
✓ rdfs:label	{name}

This creates an OWL `ObjectProperty` which can be used to relate instances of two classes. In OWL, there are 2 different property types `ObjectProperty` and `DatatypeProperty`. The latter can be used to relate an instance of a class and RDF literals and XML Schema datatypes, or *Datatypes*.

3.1.4. Create the `MargheritaPizza` class and add an annotation

Create a new subclass of the `Pizza` class as you did earlier with the other classes, called `MargheritaPizza`.

Often it is useful to add an annotation to the class. You will notice, there is a field called `rdf:label` in the "Annotations" panel. Also, there is a list of annotation `DatatypeProperty` items in the "Properties" panel (yellow icon).



Add an new annotation property, called `rdfs:comment`. Select `rdfs:comment` from the "Properties" panel, drag-and-drop it from the "Properties" panel to the "Annotations" panel. Edit the `rdfs:comment` field and enter in some text about the `MargheritaPizza` like so:

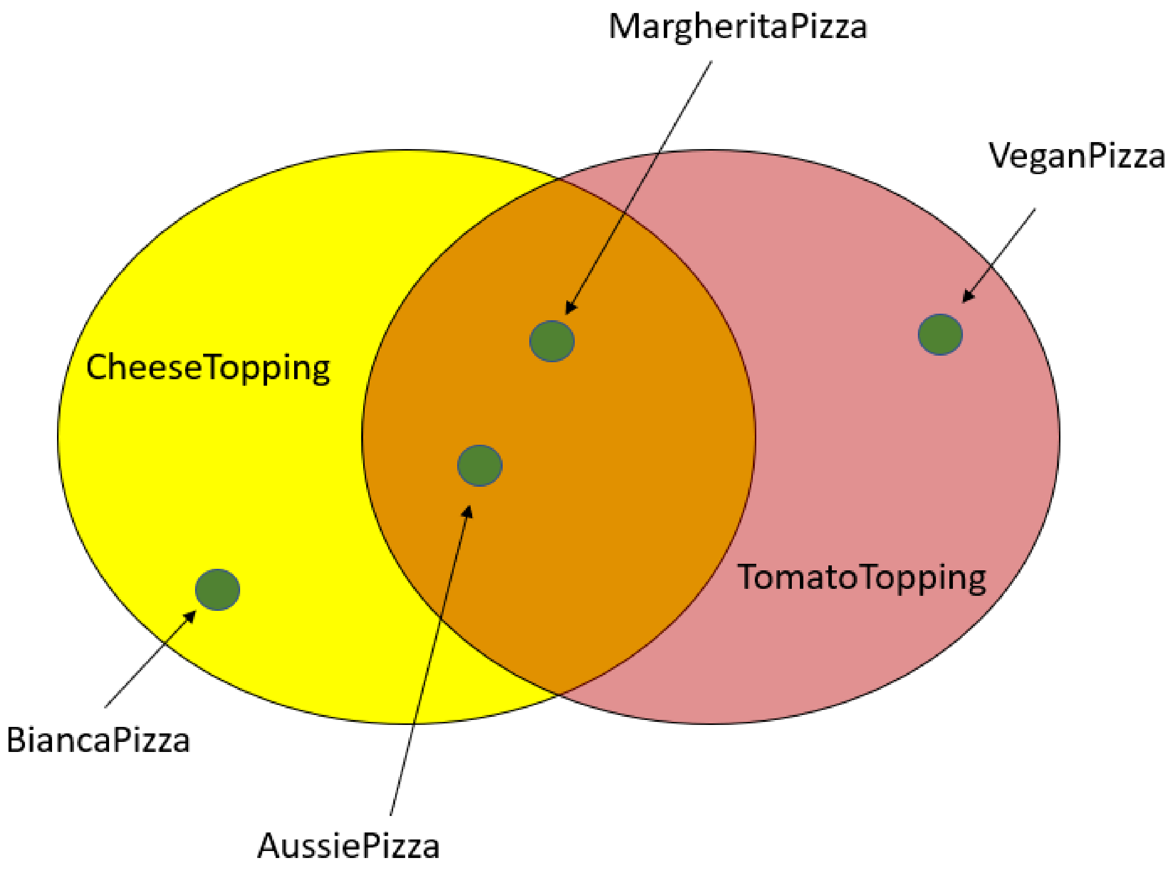
> *According to legend, in 1889, the Margherita Pizza was created and named after Queen Margherita.*

3.1.5. Add semantics to the `MargheritaPizza` class

We now want to add more semantics to the `MargheritaPizza` class to express that it has a relationship with the `TomatoTopping` and the `CheeseTopping`. To do so, we need to introduce the idea of a *Class Restriction*.

Background to Class Restrictions

To understand *Class Restrictions*, it's useful to think about it in terms of a Venn diagram. See below:



In OWL, we use Description Logic to capture class semantics. We use class restrictions to narrow down the possible logical statements about that class and their set of instances. Using the MargheritaPizza example, we know that it is a pizza that has cheese and tomato toppings. To express this, we create a restriction on the subClassOf property for MargheritaPizza with the following:

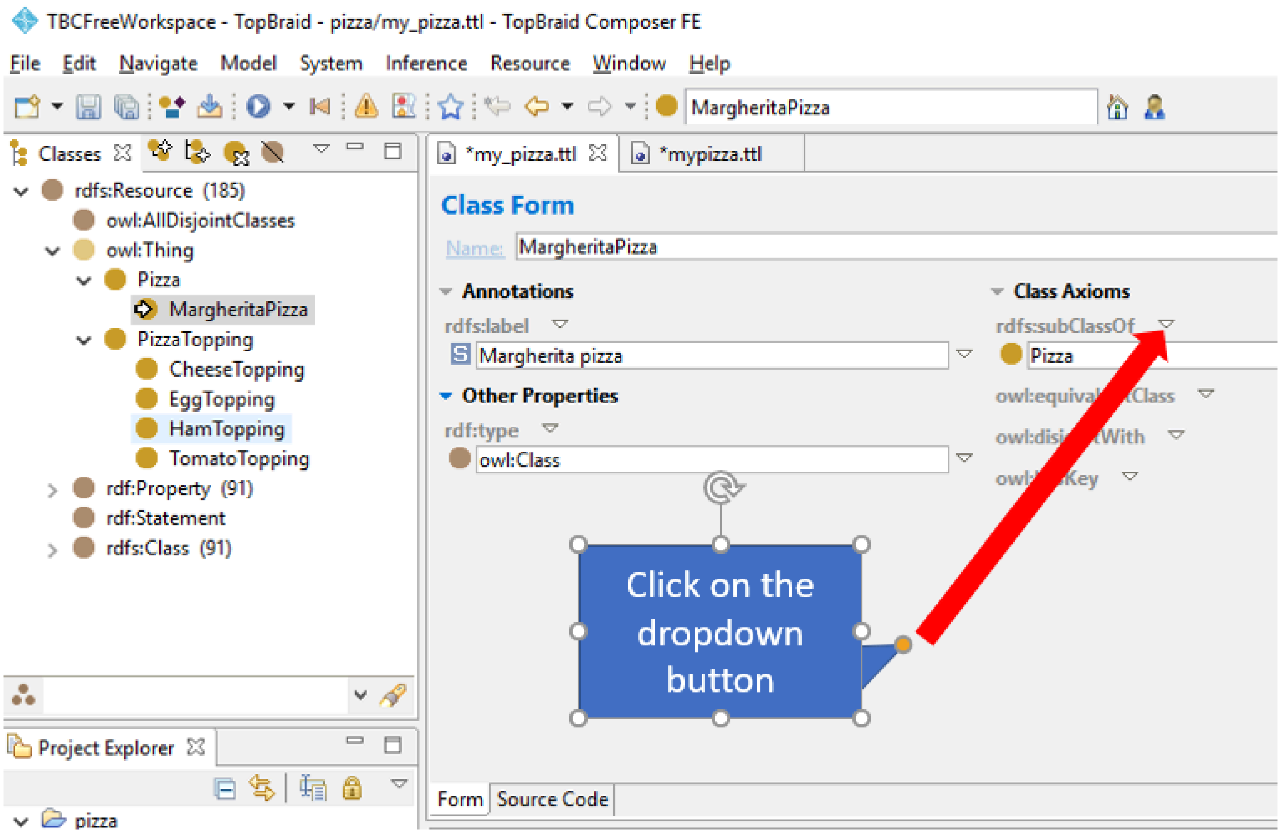
- There exists instances of Pizza where the `hasTopping` property is `CheeseTopping`
- There exists instances of Pizza where the `hasTopping` property is `TomatoTopping`

The other example in the diagram show that, the `BiancaPizza` is a pizza that has CheeseTopping but no TomatoTopping, and we can express that using subClassOf restrictions.

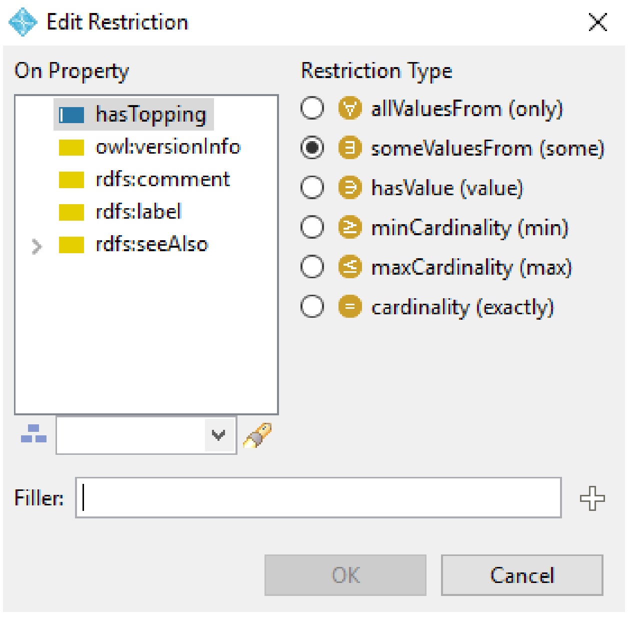
These restrictions can then be used in software reasoners to infer a list of pizzas which has no TomatoToppings, which would include the BiancaPizza. Similarly, we can use software reasoners to infer a list of pizzas which has CheeseToppings, which would include the BiancaPizza, AussiePizza and the MargheritaPizza.

Create the Pizza class restrictions

To create a subClassOf restriction, click on the dropdown button on the `rdfs:subClassOf` field, and select “Create restriction”.



Select the “hasTopping” property, and the “someValuesFrom” Restriction Type.



We now need to specify the PizzaTopping classes for MargheritaPizza.

Click on the “+” button in the “Filler” field and select “CheeseTopping”.

Repeat for “TomatoTopping”.

3.1.6. Extra exercise

Create a new class called `AussiePizza` and create subClassOf restrictions of `hasTopping` with `CheeseTopping`, `TomatoTopping`, `EggTopping` and `HamTopping`

4. Part 2. Import an existing ontology into the Topbraid editor (15-20mins)

Often you won't be creating an ontology from scratch, but rather importing this into your workspace. In this part of the tutorial, we will import the *Pizza ontology* created by the University of Manchester which was developed for learning OWL.

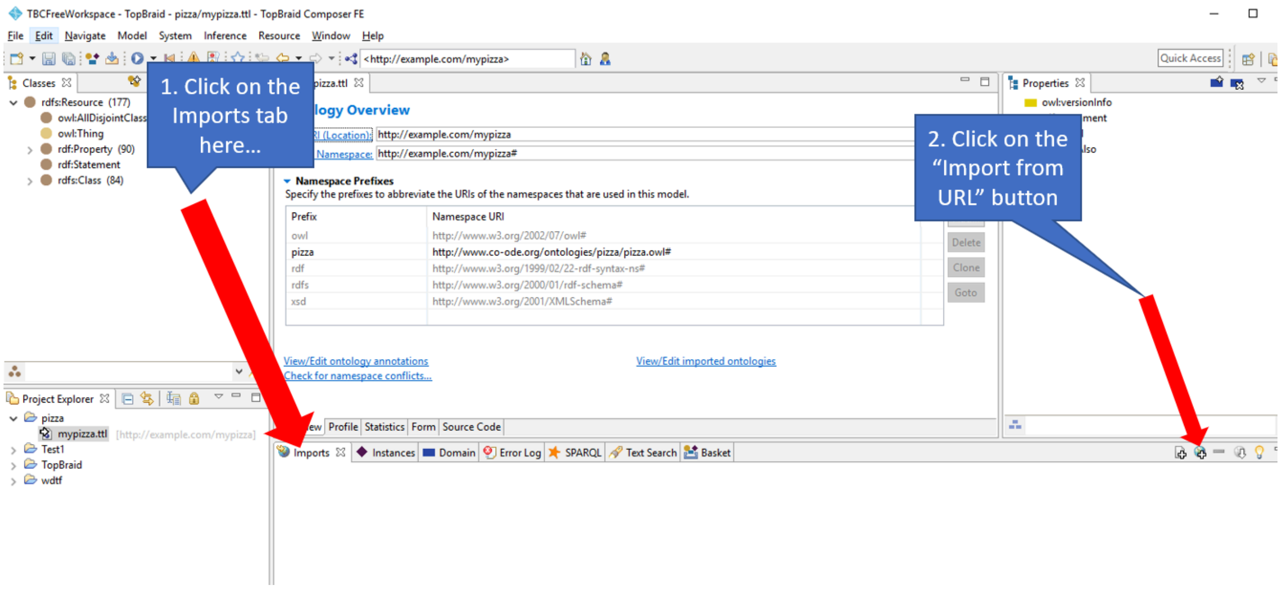
4.1. Exercise 2. Import the Pizza ontology and explore its features

In the current project, create a new RDF file, called "pizza"

4.1.1. Import the Pizza ontology

In your new RDF 'pizza' file, import the Pizza ontology from this URL:
https://protege.stanford.edu/ontologies/pizza/pizza.owl

To import the Pizza ontology, navigate to the `Imports` tab and click on the "`Import from URL`" button.

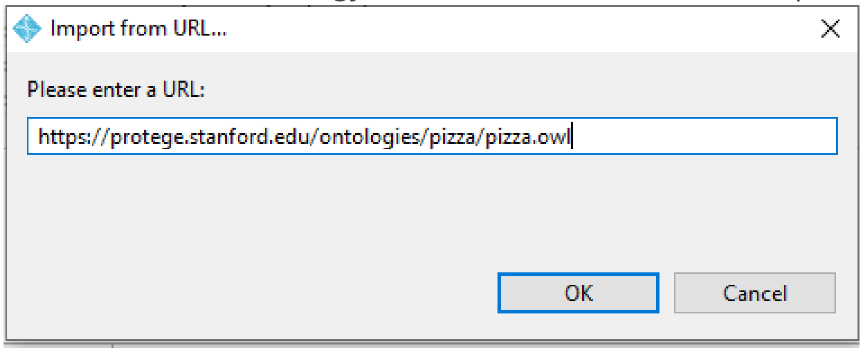

1. Click on the Imports tab here...
2. Click on the "Import from URL" button

URI (Location): http://example.com/mypizza
Namespace: http://example.com/mypizza#

Namespace Prefixes
Specify the prefixes to abbreviate the URIs of the namespaces that are used in this model.

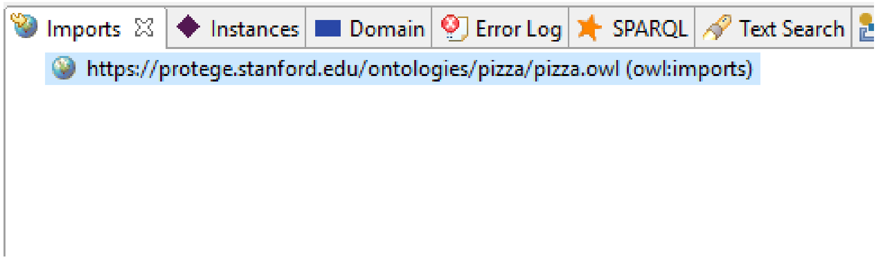
Prefix	Namespace URI
owl	http://www.w3.org/2002/07/owl#
pizza	http://www.co-ode.org/ontologies/pizza/pizza.owl#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
xsd	http://www.w3.org/2001/XMLSchema#

Enter in the Pizza ontology URL (see above) into the text input box like so:



Please enter a URL:
https://protege.stanford.edu/ontologies/pizza/pizza.owl

The imported Pizza ontology will appear in the Imports tab like so:



https://protege.stanford.edu/ontologies/pizza/pizza.owl (owl:imports)

4.1.2. For discussion

- Take a few moments to navigate around the Pizza ontology
- What do you notice about the Margherita Pizza definition?

4.1.3. Extra exercise

Create a new class called `AussiePizza` by extending the framework in the imported pizza ontology

5. Part 3. Query the RDF data using SPARQL in Topbraid (15-20mins)

Objective: Learn how to write simple SPARQL queries to understand RDF data using Topbraid Composer

The part of the tutorial aims to provide a very quick overview of SPARQL and write some simple queries. We will focus on SPARQL SELECT queries.

5.1. Introducing SPARQL

SPARQL = SPARQL Protocol and RDF Query Language (pronounced “spar-kle”)

SPARQL is a structured query language that is used to query RDF data, much like SQL is used to query relational databases. There are 4 query forms:

- SELECT - used to get RDF values in a tabular result form
- CONSTRUCT - create a RDF graph based on returned RDF graph values
- ASK - returns a simple boolean (true/false) result
- DESCRIBE - get a descriptive RDF graph (usually up to the query engine to define returned result)

For the purposes of this tutorial, we’ll explore using SPARQL SELECT query

5.1.1. Anatomy of a SPARQL SELECT query

The SPARQL query takes the form

```
SELECT [ list of variables delimited by a space ]
WHERE
{
    [
          list triple statements separated by '.'
    ]
}
```

Example:

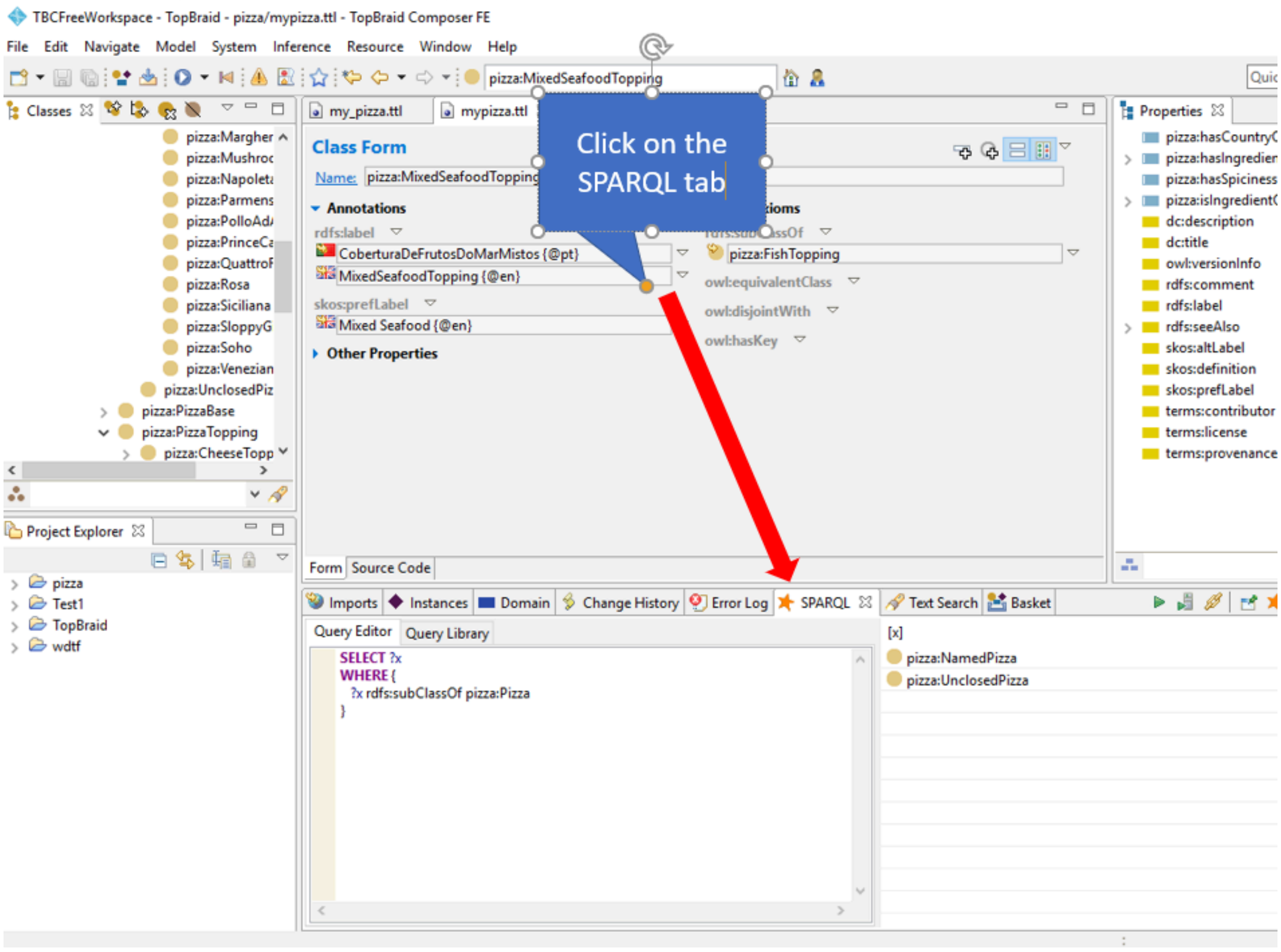
```
SELECT ?subject1 ?subject2
WHERE
{
    ?subject1 ?predicate1 ?object1 .
    ?subject2 ?predicate2 ?object2 .
}
```

A list of rows will be returned based on the list of variables in the SELECT line. In the example above, rows with columns of `?subject1 ?subject2` will be returned. e.g.

```
| ?subject1 | ?subject2 |
-------------------------
| "foo"     |  'bar"    |
-------------------------
```

5.2. Exercise 3: Querying the Pizza ontology using SPARQL

Using the imported Pizza ontology from Part 2, select the SPARQL tab like so:



```
SELECT ?x
WHERE {
  ?x rdfs:subClassOf pizza:Pizza
}
```

```
[x]
pizza:NamedPizza
pizza:UnclosedPizza
```

Use the queries below to get hands-on with SPARQL SELECT queries with the Pizza ontology.

5.2.1. Query the Pizza ontology and list the direct subclasses of `pizza:Pizza`

```
SELECT ?x
WHERE {
   ?x rdfs:subClassOf pizza:Pizza
}
```

Discuss: What's happening here?

5.2.2. Query the Pizza ontology and list all subclasses of `pizza:Pizza` (direct and indirect)

```
SELECT ?x
WHERE {
   ?x rdfs:subClassOf+ pizza:Pizza
}
```

Discuss: What's happening here?

5.2.3. Query the Pizza ontology and list all subclasses of `pizza:Pizza` by its label

```
SELECT ?x
WHERE {
   ?x rdfs:subClassOf+ pizza:Pizza .
   ?x rdfs:label ?label
   FILTER  (regex(?label, "margherita", "i")) .
}
```

Discuss: What's happening here?

5.2.4. Find all Pizza classes that have `TomatoTopping`

```
SELECT ?x
WHERE {
   ?x rdfs:subClassOf+ pizza:Pizza .
   ?x rdfs:subClassOf [
             a owl:Restriction ;
             owl:onProperty pizza:hasTopping;
        owl:someValuesFrom pizza:TomatoTopping
        ]
}
```

Discuss: What's happening here?

5.2.5. Try to create a SPARQL query to find Pizzas that has a `GarlicTopping`

5.2.6. Advanced querying

Find all Pizza classes that have `TomatoTopping` and all types of classes of `FishTopping`

```
SELECT ?x
WHERE {
   ?x rdfs:subClassOf+ pizza:Pizza .
   ?x rdfs:subClassOf [
                a owl:Restriction ;
                owl:onProperty pizza:hasTopping;
        owl:someValuesFrom pizza:TomatoTopping
        ] .
          ?x rdfs:subClassOf [
                a owl:Restriction ;
                owl:onProperty pizza:hasTopping;
        owl:someValuesFrom ?fishClasses
        ] .
    ?fishClasses rdfs:subClassOf+ pizza:FishTopping
}
```

Find all Pizza classes that have a topping with “seafood” in its label

```
SELECT ?x
WHERE {
   ?x rdfs:subClassOf+ pizza:Pizza .
   ?x rdfs:subClassOf [
                a owl:Restriction ;
                owl:onProperty pizza:hasTopping;
        owl:someValuesFrom ?topping
        ] .
   ?topping rdfs:subClassOf+ pizza:PizzaTopping .
   ?topping rdfs:label ?label .
   FILTER  (regex(?label, "seafood", "i")) .
}
```

Discuss: What’s happening here?

5.3. Going deeper

If you would like to explore more about SPARQL, we’d recommend the following tutorial: SPARQL Tutorial by Apache Jena

6. References

Check out the Learning resources section for more material.