**SADViz User Manual**

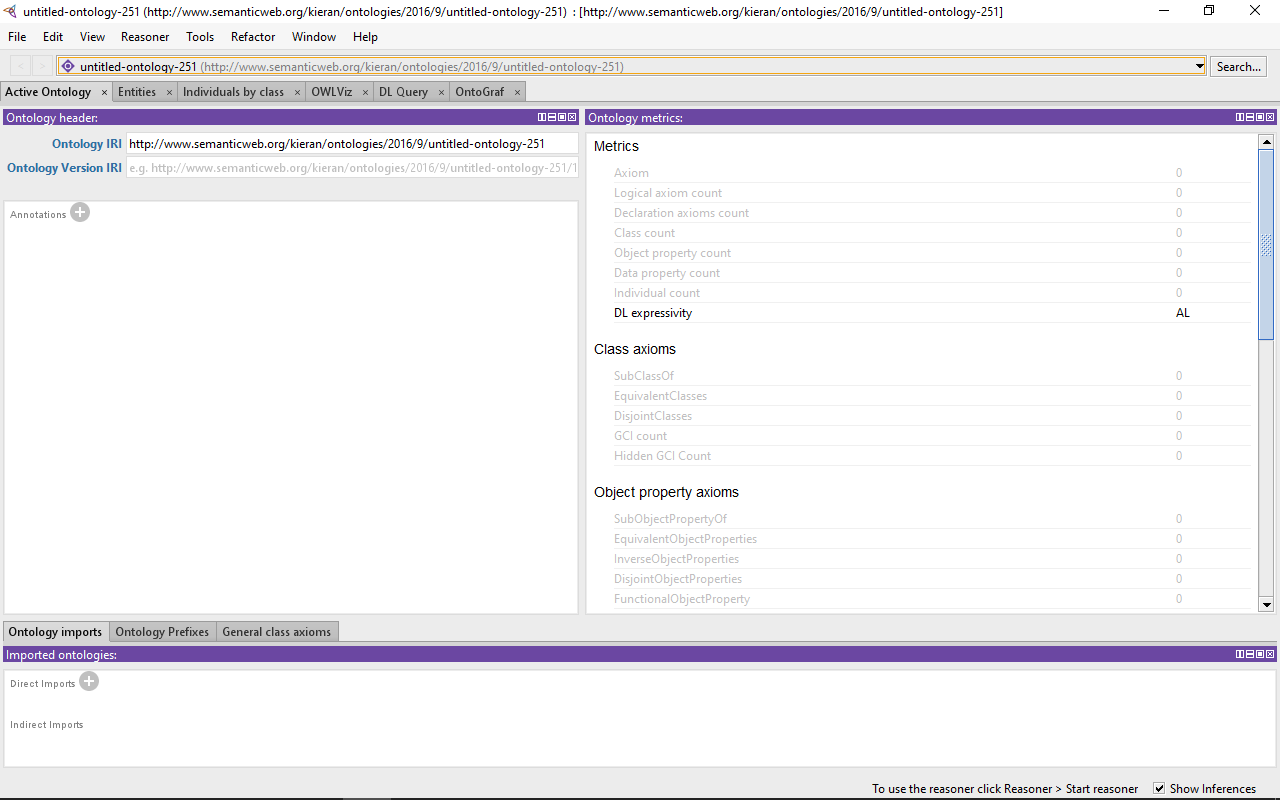
Introduction

The Super Awesome Dynamic Visualisation (SADViz) was developed for Protégé, which is an open-source ontology viewer and editor. The purpose of SADViz is to graphically visualise a loaded ontology in Protégé, showing classes and associated relationships among classes. Included with SADViz are various added functionalities that enable the user to view the graph in various ways, or assist the user in obtaining information from the graphed ontology.

This user manual aims to provide the user with a short guide on what Protégé is, various features of Protégé which are relevant to SADViz, as well as how to install plugins like SADViz. It then goes on to describe how a user can install the SADViz plugin itself. A description of the graphical user interface of SADViz is given. Following on from this is a breakdown of how to populate the visualisation, as well as how to make use of the added functionality.

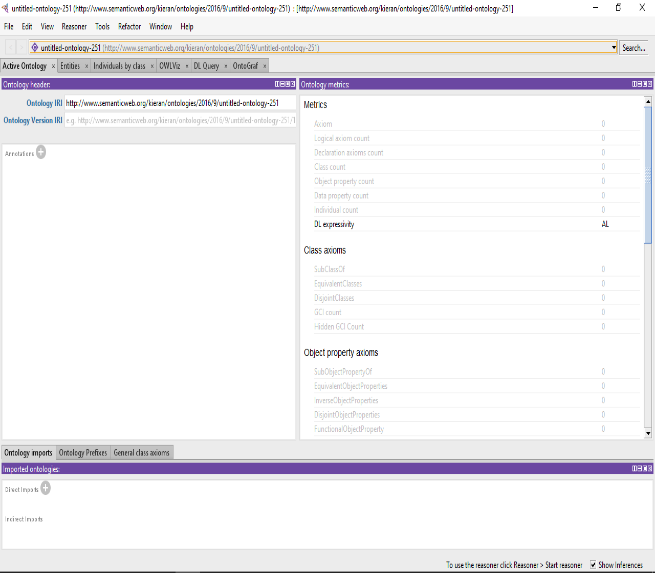
Getting Started with Protégé

As described above, Protégé is an open-source ontology editor and viewer. The latest version of Protégé can be downloaded from: http://Protégé.stanford.edu/. Once downloaded, installed, and run, you will be greeted by this screen:

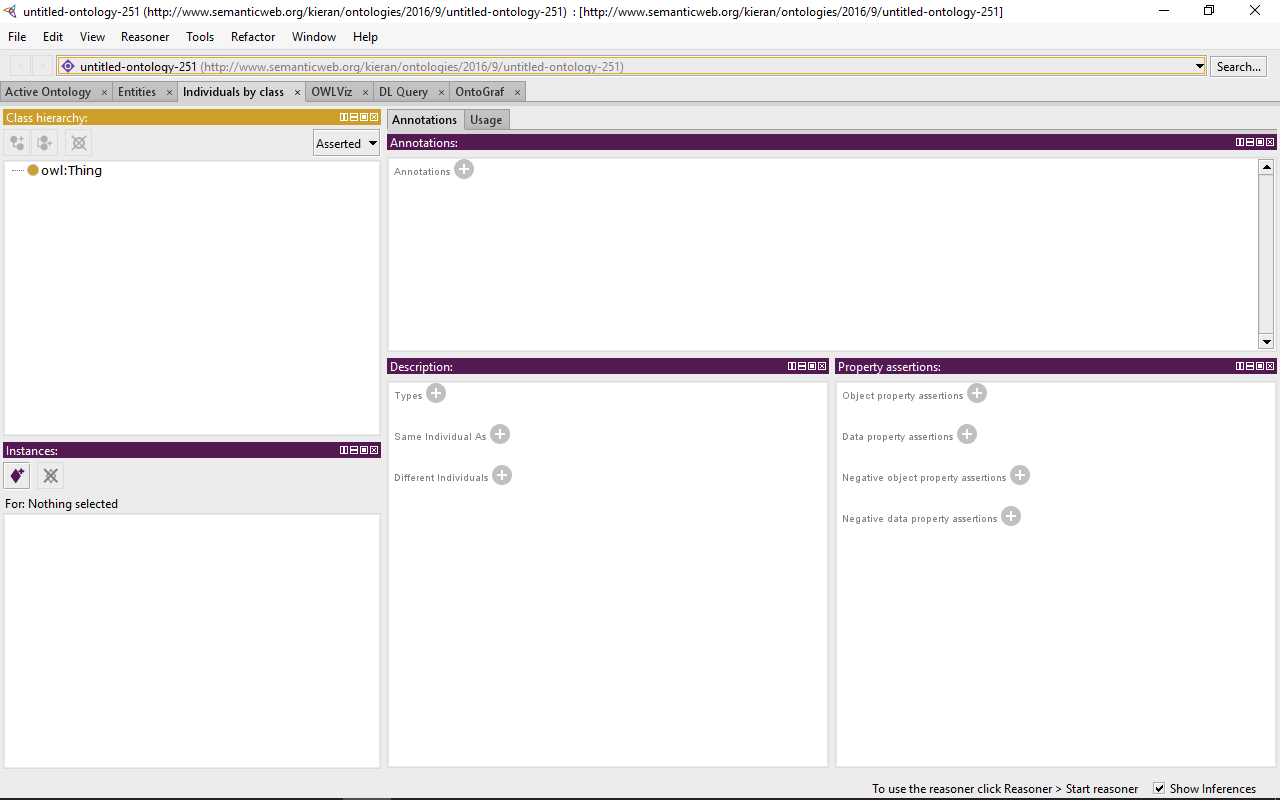


*Figure 1: Protégé 5’s Initial Screen*

From this initial screen you will notice various tabs, and views. Each tab contains its own set of views, and each view is used to represent the information of a loaded ontology in some way.



*Figure 2: Example of a view*



*Figure 3: Example of a tab*

There are various ontologies you could load into Protégé, and some of these can be found at: http://Protégéwiki.stanford.edu/wiki/Protégé\_Ontology\_Library. You can either load an ontology through “File > Open…”: opening a downloaded ontology, “File > Open from URL…”: opening an ontology using a given URL, or “File > Open Recent”: opening a recently loaded ontology.

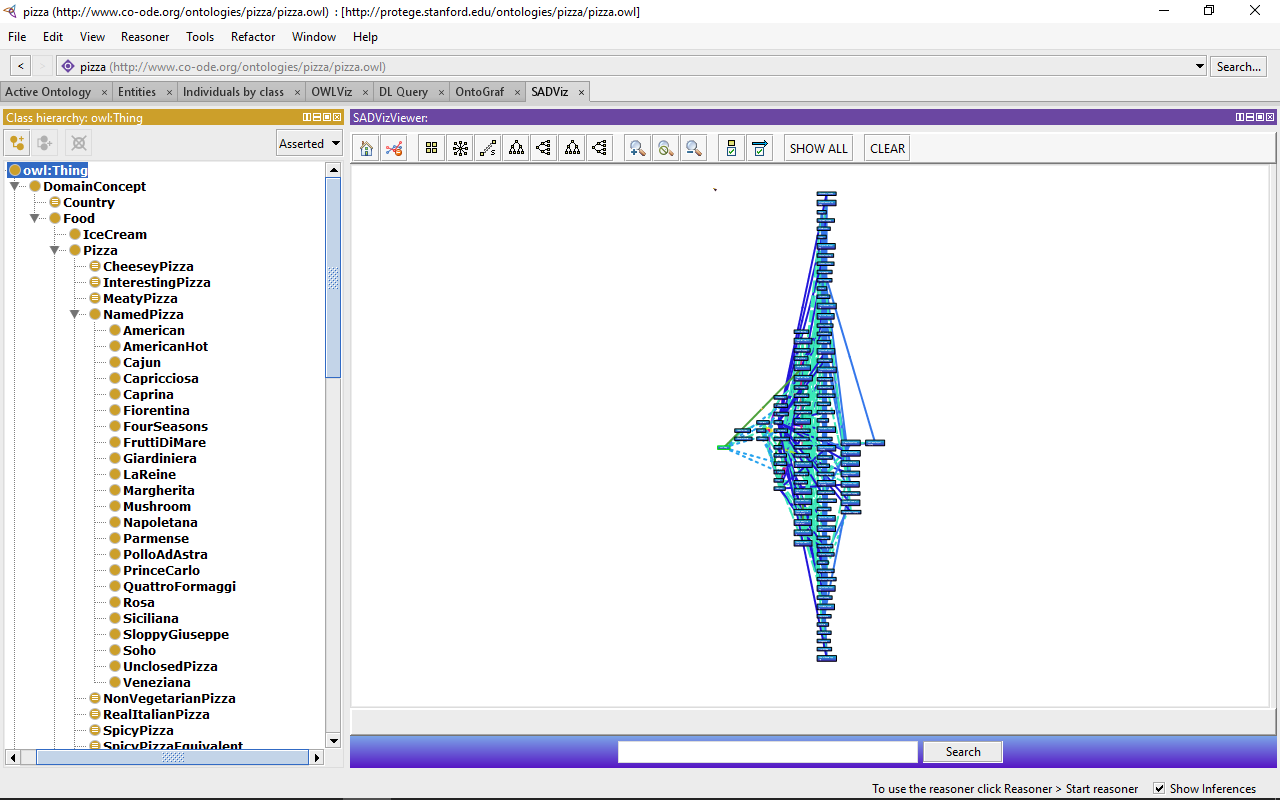
Protégé enables developers to create plug-ins which allow added functionality, or different views in which a user can view the information of a loaded ontology. You can view a list of developed plug-ins at: http://Protégéwiki.stanford.edu/wiki/Protégé\_Plugin\_Library. In order to install a given plugin, navigate to the Protégé file, open the plugins directory, and place the JAR file of a downloaded plugin within this directory.

Installing SADViz

As mentioned above, SADViz was developed to graphically visualise the classes and relationships between these classes. Once downloaded, extract contents, and navigate to the target directory. Within this directory you will find the SADViz-1.0.0.jar file. Copy the jar file into the Protégé plugins directory. At this point SADViz should be installed in Protégé and ready to execute. In order to activate the SADViz tab, you simply open Protégé, select Window > Tabs > SADViz. If SADViz is not under the list of tabs, then there was some error during the installation process.

The Graphic User Interface

The graphical user interface of SADViz plays an essential role in the function of the plug-in, due to it being such a visual tool by nature. Included below is a figure of the SADViz plug-in tab with the pizza.owl ontology loaded.

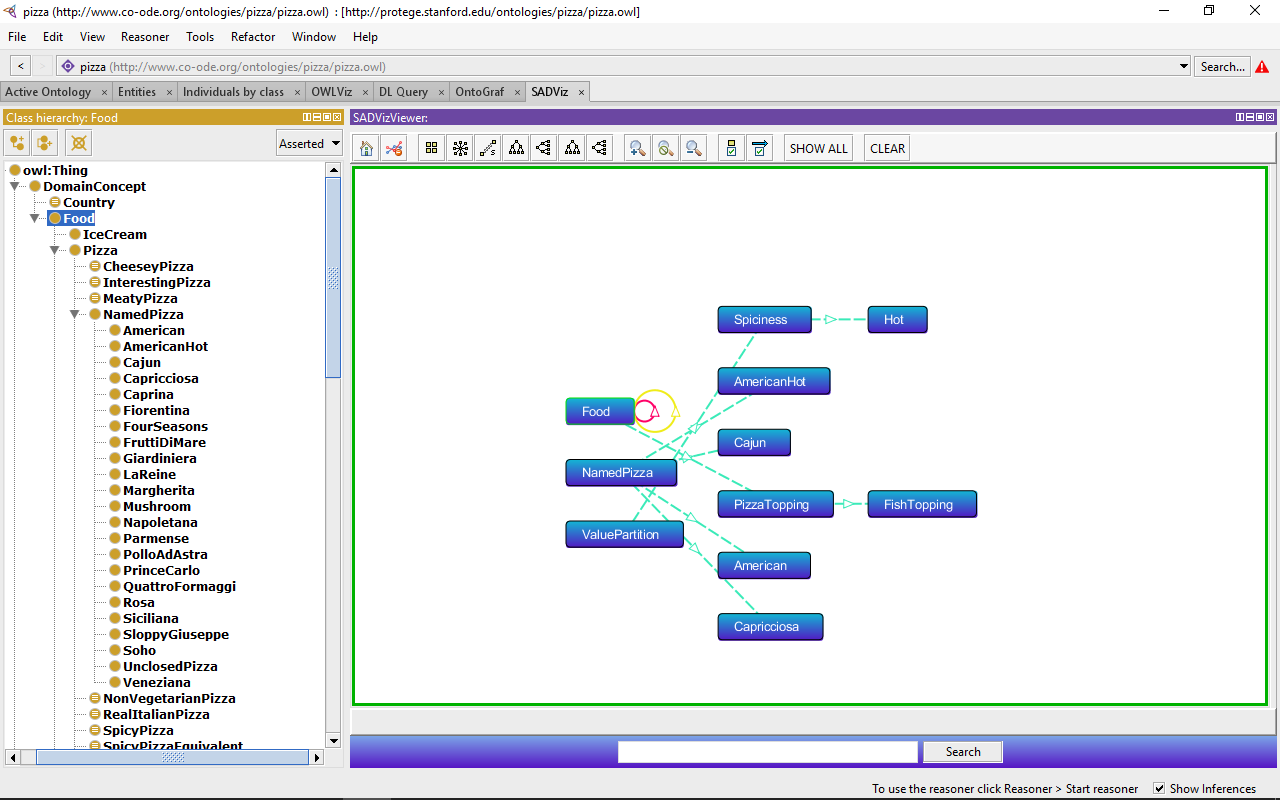


*Figure 4: The SADViz Tab showing*

Note how all nodes and relationships in the pizza ontology are displayed immediately within the SADViz View. On the left-hand side of the tab is a Class Hierarchy Viewer which displays the classes and their subclasses in a given ontology, in a text-based format. On the right-hand side of the tab is the SADViz View with all the classes and their relationships of the loaded ontology graphed. The user is also provided with various buttons which enable added functionality that can be used to enhance the information obtained by the graph, more on these features later.

Populating a Visualisation

When you first start up Protégé and open the SADViz tab without a ontology loaded, the only node that will be displayed is the “owl:Thing” node. Once you have loaded an ontology, all classes and relationships are displayed within the SADViz View. If the user clicks the “CLEAR” button, located at the top of view, all classes and relationships are cleared from the view. The user can then use the Class Hierarchy Viewer to select classes to be graphed in the SADViz View. As the user navigates through the text-based hierarchy of classes, these classes are graphed within the SADViz View, with their corresponding relationships.

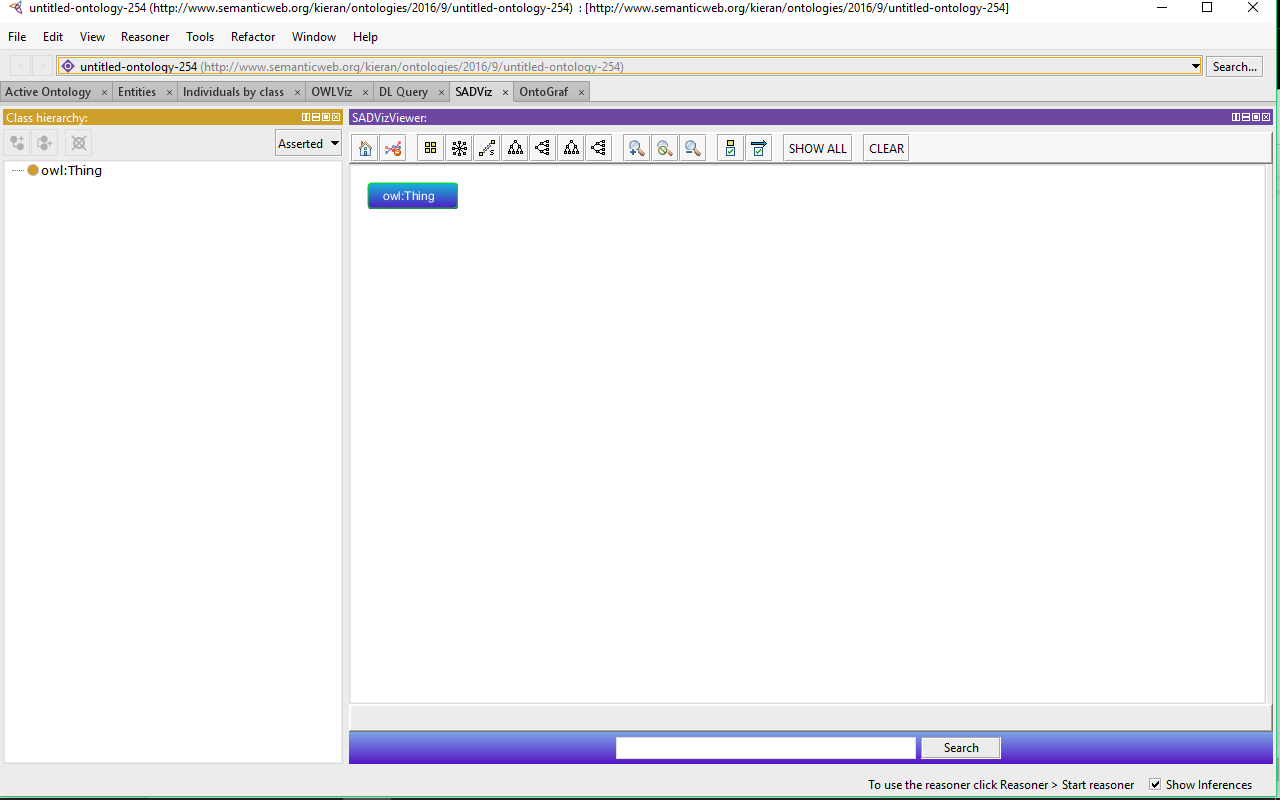


*Figure 5: Selected nodes graphed using the Class Hierarchy Viewer*

If the user has cleared the view, populated the visualisation with various classes, and would now like to view the entire the ontology again, they may do so by selecting the “SHOW ALL” button which will re-populate the visualisation with all classes and associated relationships.

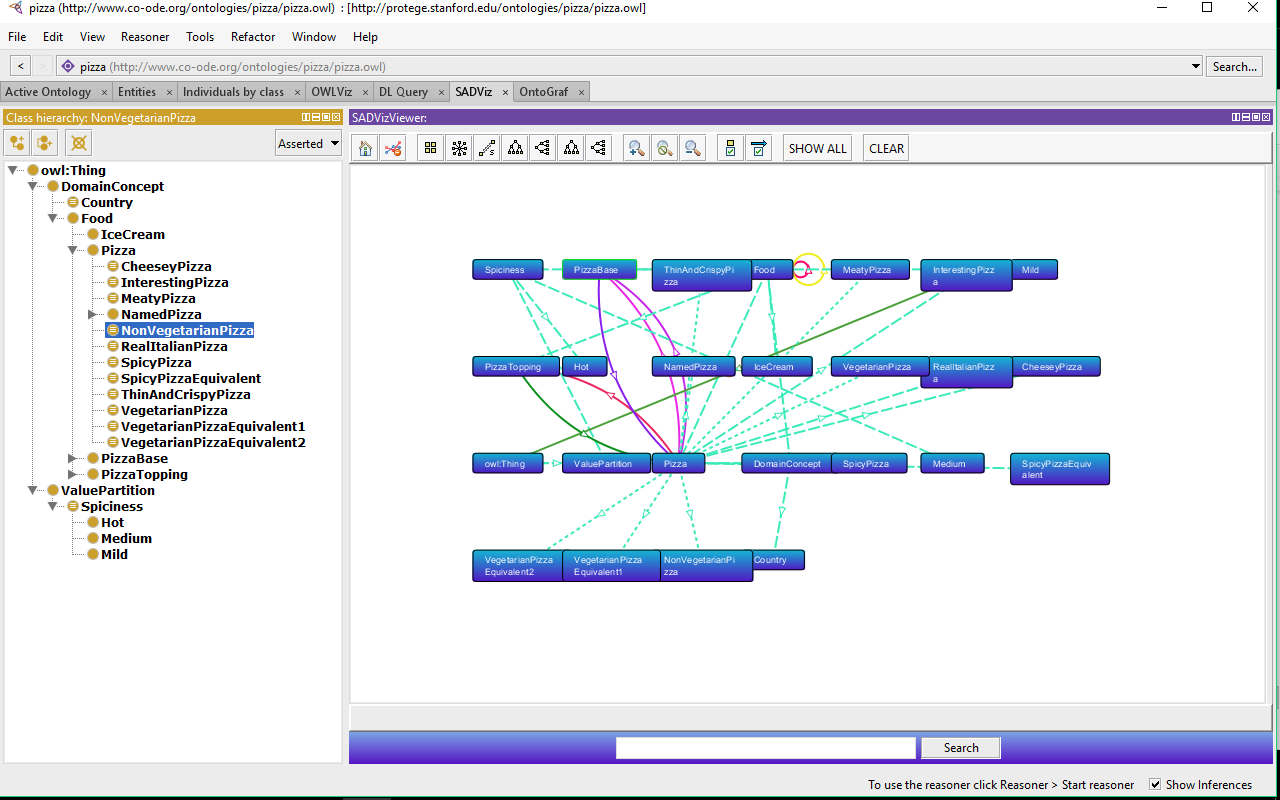
Making Use of Added Functionality

Built into the SADViz plugin are various added functionalities which you can make use of to manipulate the graphed components and enhance the information extracted from loaded ontologies. In Figure 6 below you will notice a close up of the various buttons located at the top of the SADViz view.

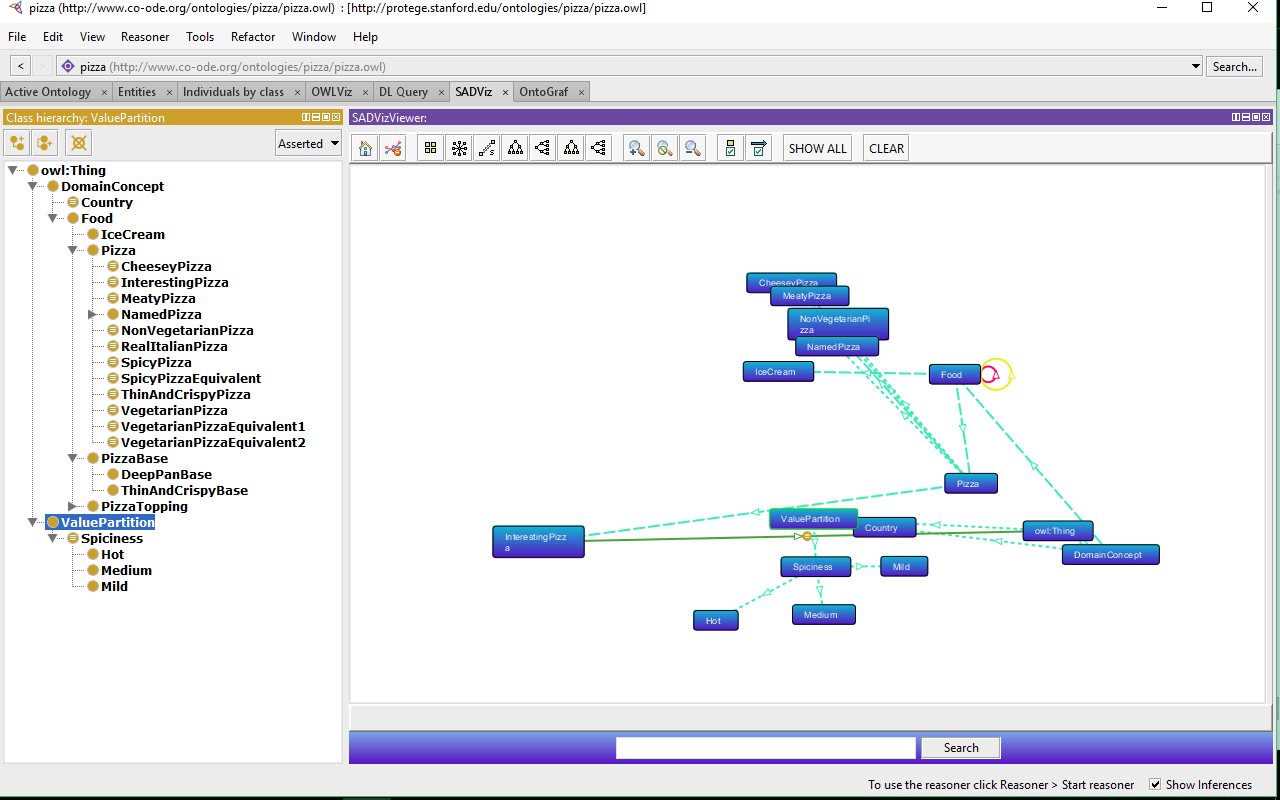


*Figure 6: Close up of the added functionality buttons*

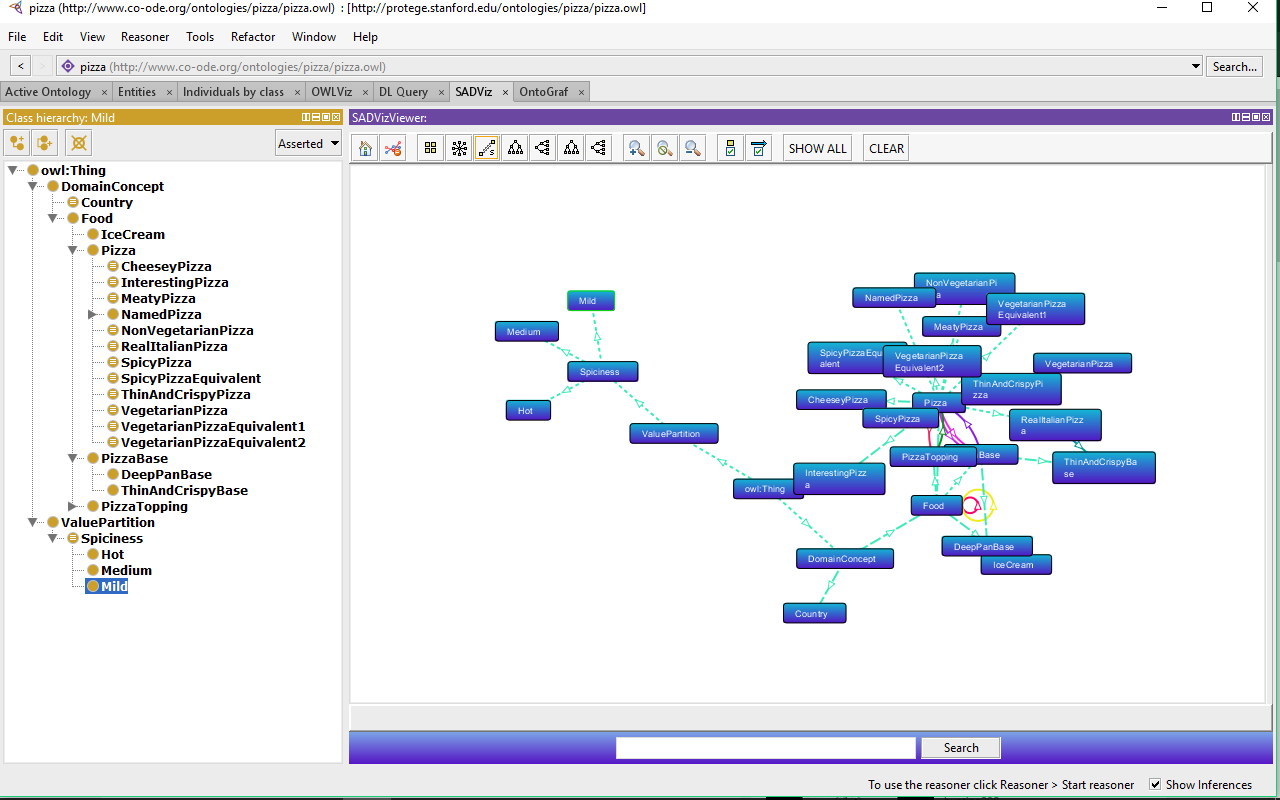
Moving across the bar from left to right, a breakdown is given for button and their function:

* “Focus on Home” button: provides an overview of the classes and their relationships.
* “Remove Orphan Nodes” button: removes any nodes which do not have relationships with any other classes.
* “Grid” button: places the classes of a visualised graph into a grid like form.

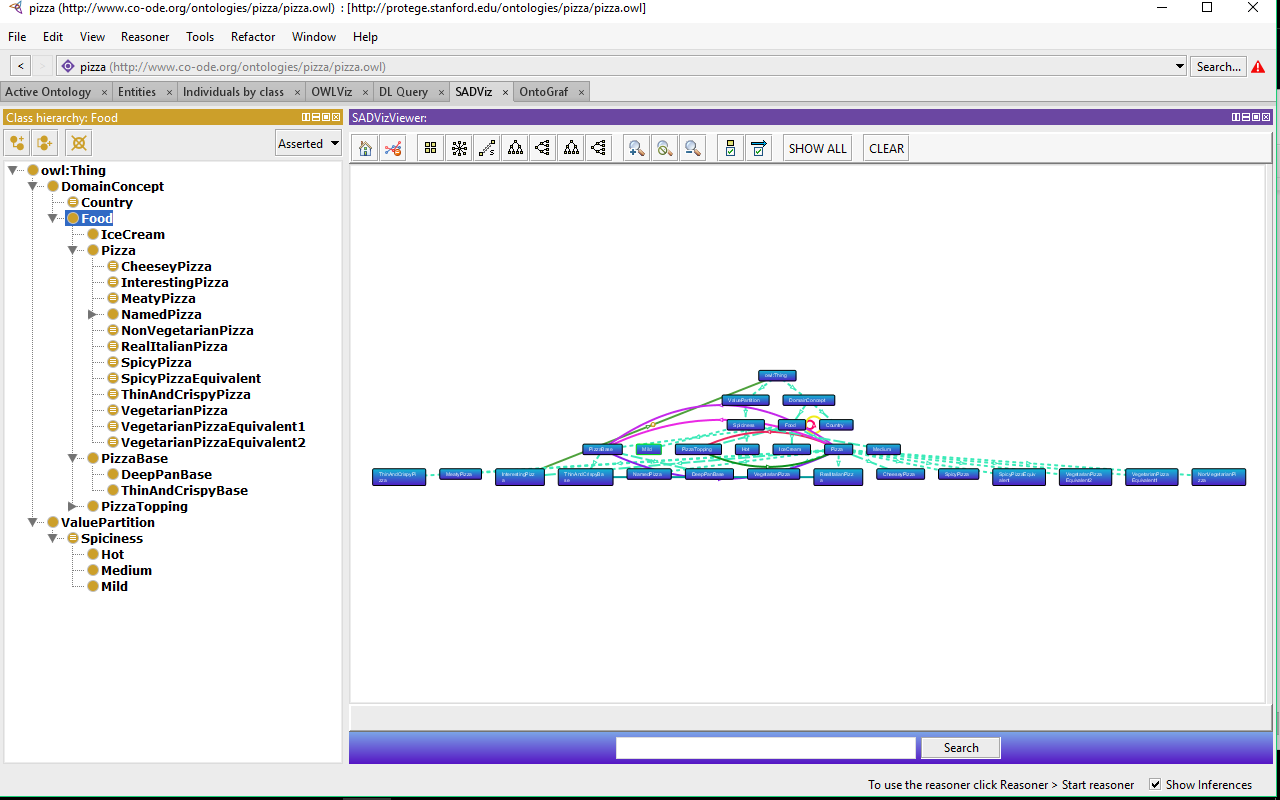
*Figure 7: An example of the layout of classes using “GRID” button*

* “Radial” button: places the classes into a radial like form.

*Figure 8: Example of layout of classes using “Radial” button*

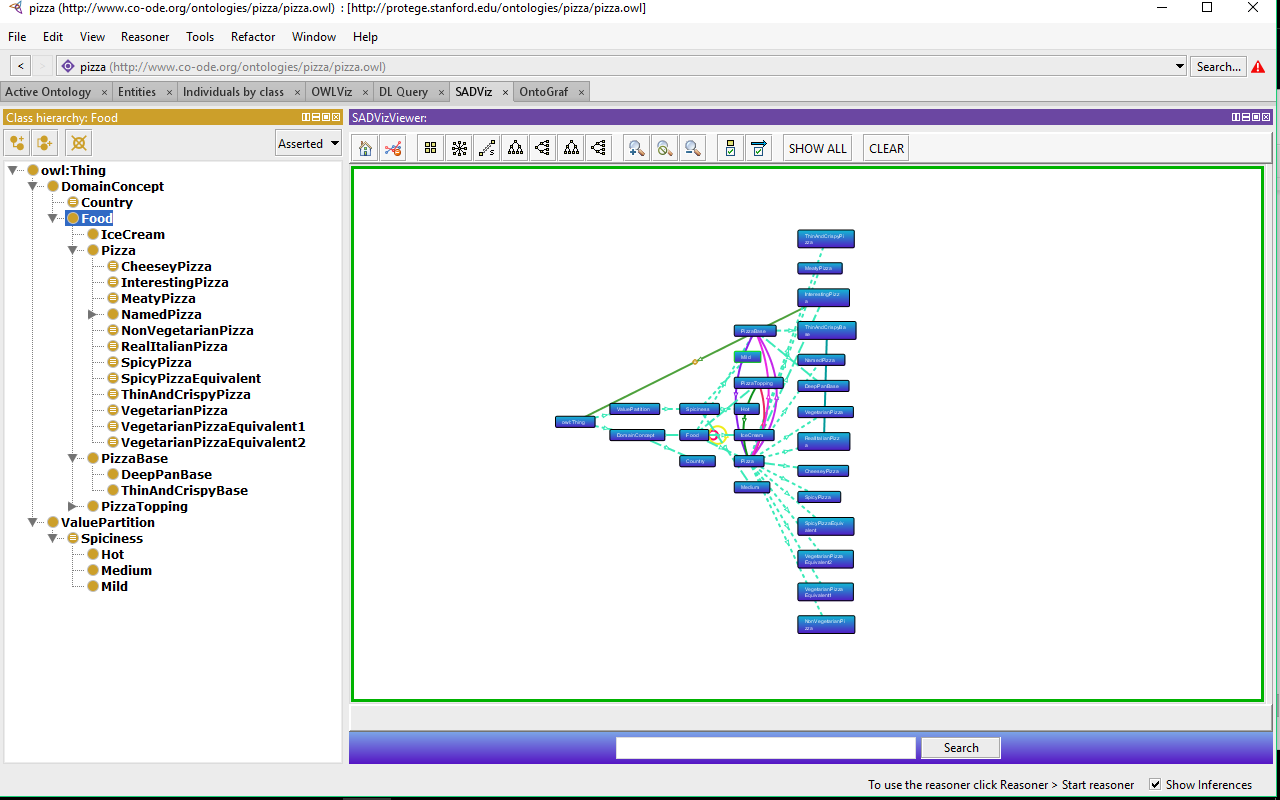
* “Spring” button: places the classes into a form where relationships are a certain set length.

*Figure 9: Example of the layout when the “Spring” button is used*

* “Tree - Vertical” button: places classes in a tree-like layout, in a vertical orientation.

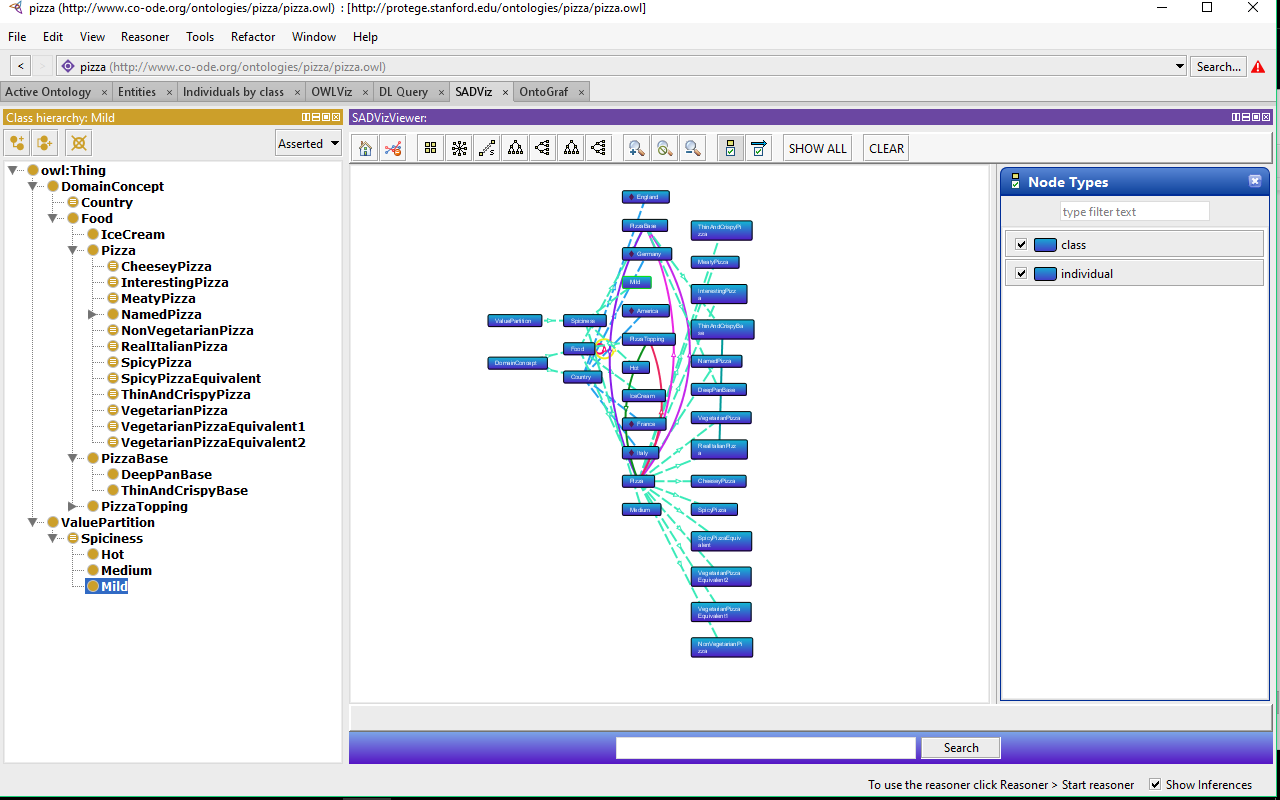
*Figure 10: Layout of classes using “Tree - Vertical” button*

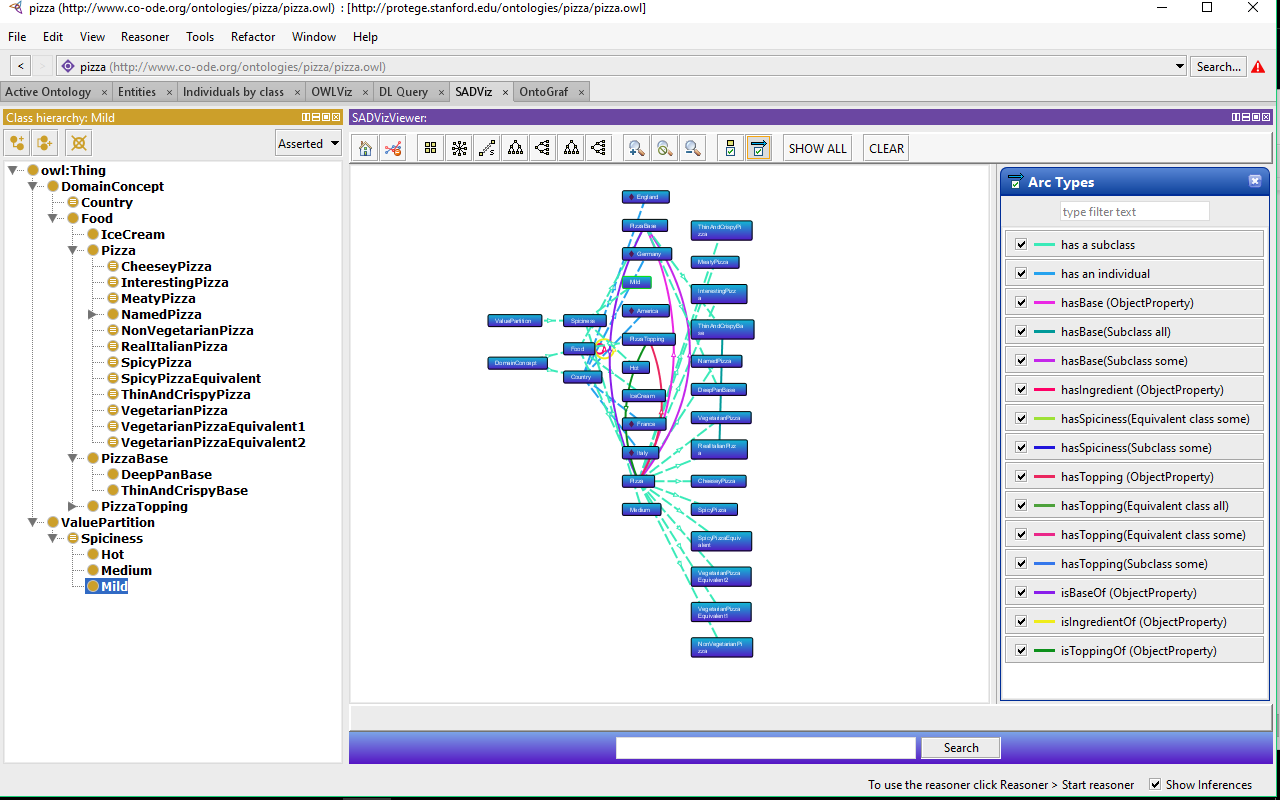
* “Tree - Horizontal” button: places classes in a tree-like layout, in a horizontal orientation.

*Figure 11: Layout of classes using the “Tree - Horizontal” button*

* “Zoom In” button: enlarges the classes and their relationships.
* “No Zoom” button: restores the visualisation to the default orientation.
* “Zoom Out” button: shrinks the classes and their relationships.
* “Node Types”: button: opens a panel to the right of the view with a list of the types of nodes currently being displayed within the visualisation. The user can select/deselect these node types to hide/show the different node types of a loaded ontology.

*Figure 12: Node Types panel displayed with types of nodes currently displayed*

* “Arc Types” button:  opens a panel to the right of the view with a list of the types of relationships currently being displayed within the visualisation. The user can select/deselect these relationships types to hide/show the different relationship types of a loaded ontology.

*Figure 13: Relationship types panel showing the different relationships within the ontology*

* “SHOW ALL” button: displays all classes and their respective relationships.
* “CLEAR” button: clears any classes and their respective relationships.
* Search functionality: located at the button of the view is a search bar and a button “Search”. A user can enter a desired class name to search through the ontology for. If there are matching results then all classes relating to what was searched for as well as their nearest related classes are shown. If there are no matches, then nothing is displayed within the visualisation.

On top of the functionality mentioned above, there is functionality embedded in the visualisation screen itself. The user is able to navigate around the ontology graph by left-clicking an empty part of the view and move the mouse. To zoom in or out, the user can scroll down or up using a mouse wheel. Nodes can be moved around by clicking and dragging them on screen. Double clicking a node will display it, as well as its closest related classes.