

OWLFCAVIEW Protege Tab Plug-in Manual

OWLFCAVIEW Protégée Tab plug-in Manual

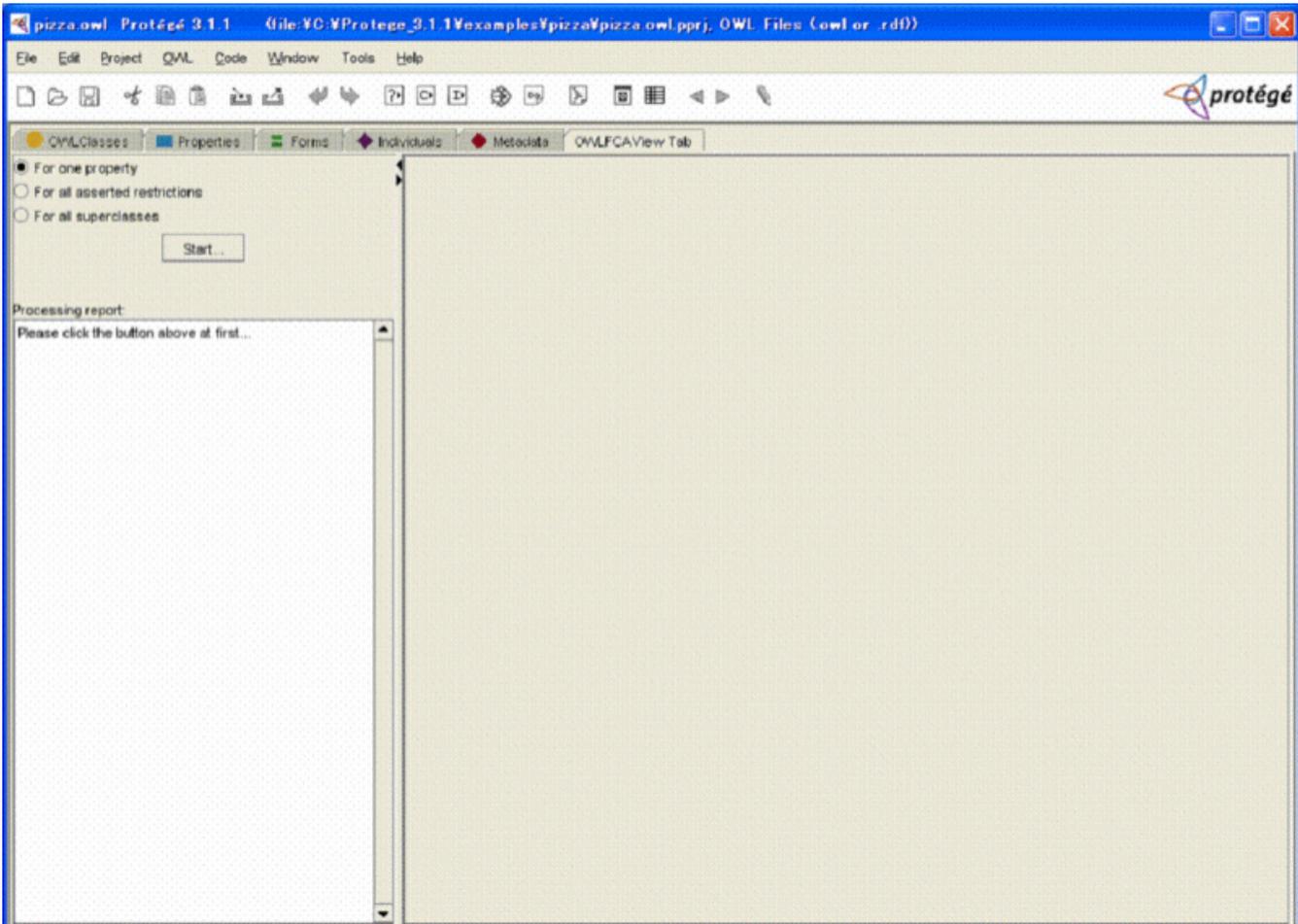
Contents of this Page

- [Specification 1: For one property](#)
- [For all asserted restrictions](#)
- [Specification 3: For all superclasses](#)

 This manual explains how to use OWLFCAVIEW Tab plug-in For Protégé OWL Ontology.

The OWLFCAVIEW Tab plug-in provides Protégé-2000 users a novel way to visualize their OWL ontologies using the formal concept analysis (FCA) technique. In current specifications, we developed a property-oriented way for the visualization of OWL ontology in Protégé-2000. First of all, please make sure that you are using Protégé-3.1 version. We assume that you are familiar with how to configure a tab plug-in in Protégé platform.

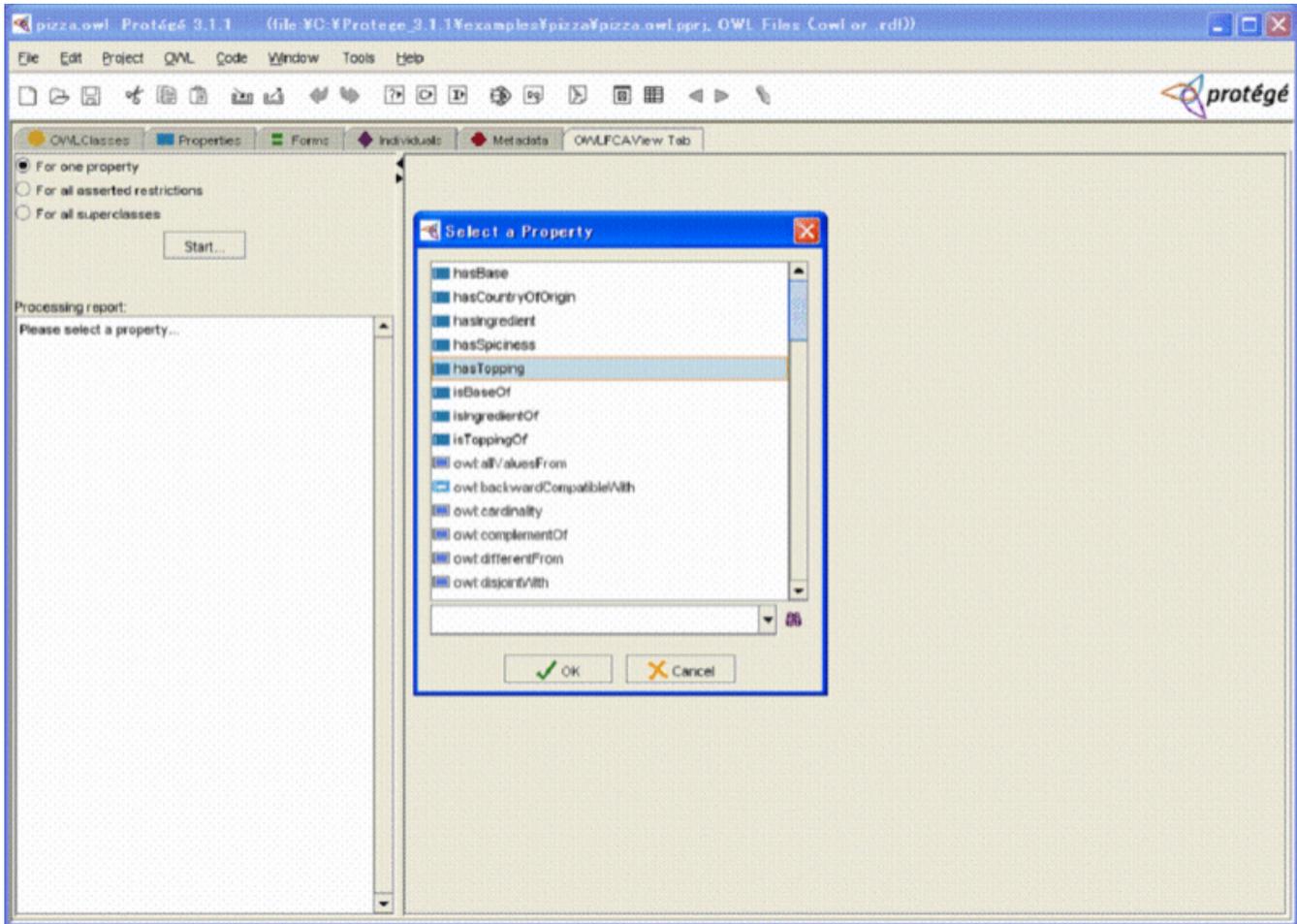
We use an OWL ontology, gpizza.owl, that is also example project used in the Protégé OWL plug-in to demonstrate how to use OWLFCAVIEW Tab plug-in. After opening the example project and configuring the OWLFCAVIEW Tab, you would see the following.



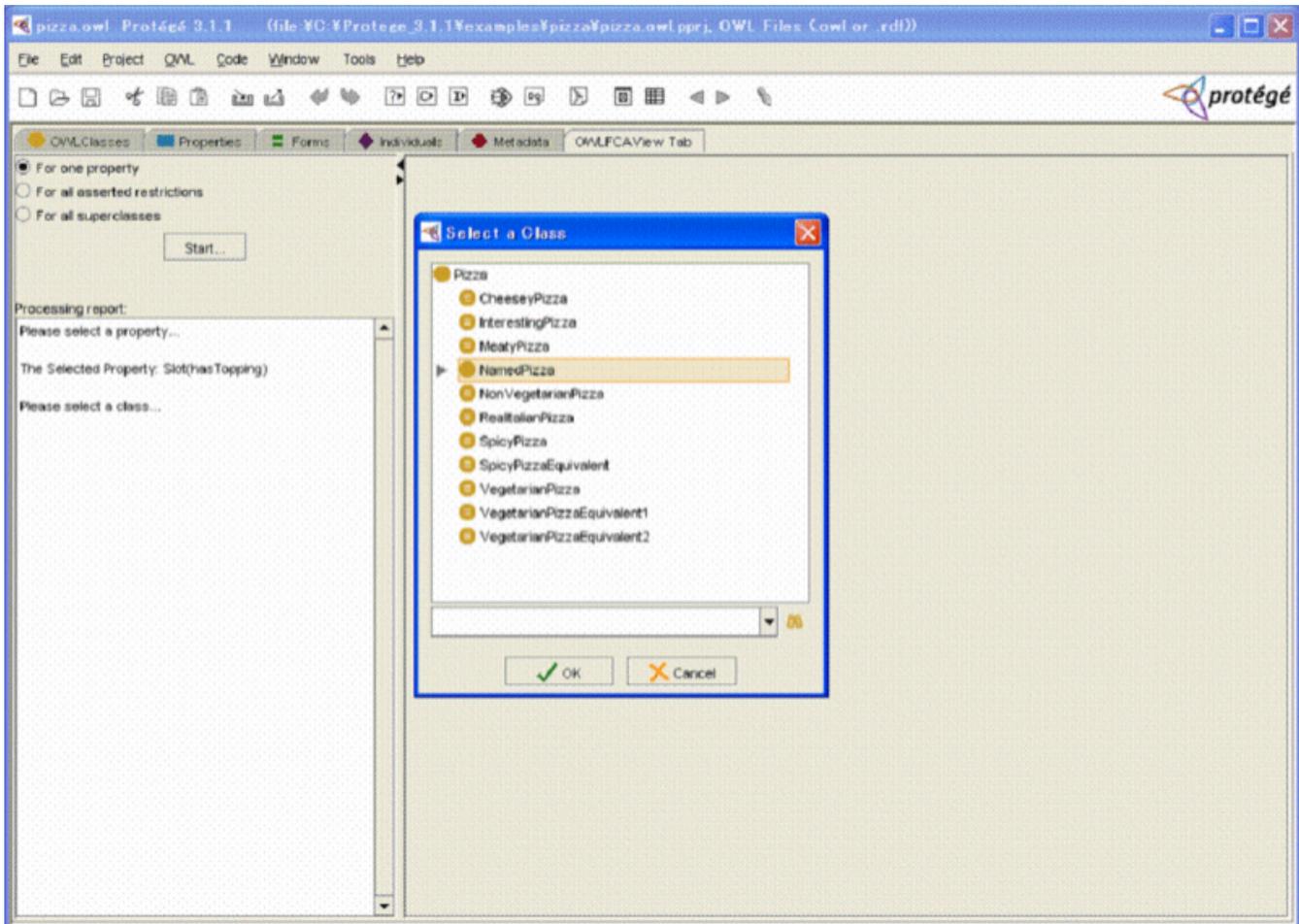
There are three basic specifications implemented in the current version of our tool. The first one is for one property, the second one for all asserted restrictions and the third one for all superclasses.

Specification 1: For one property

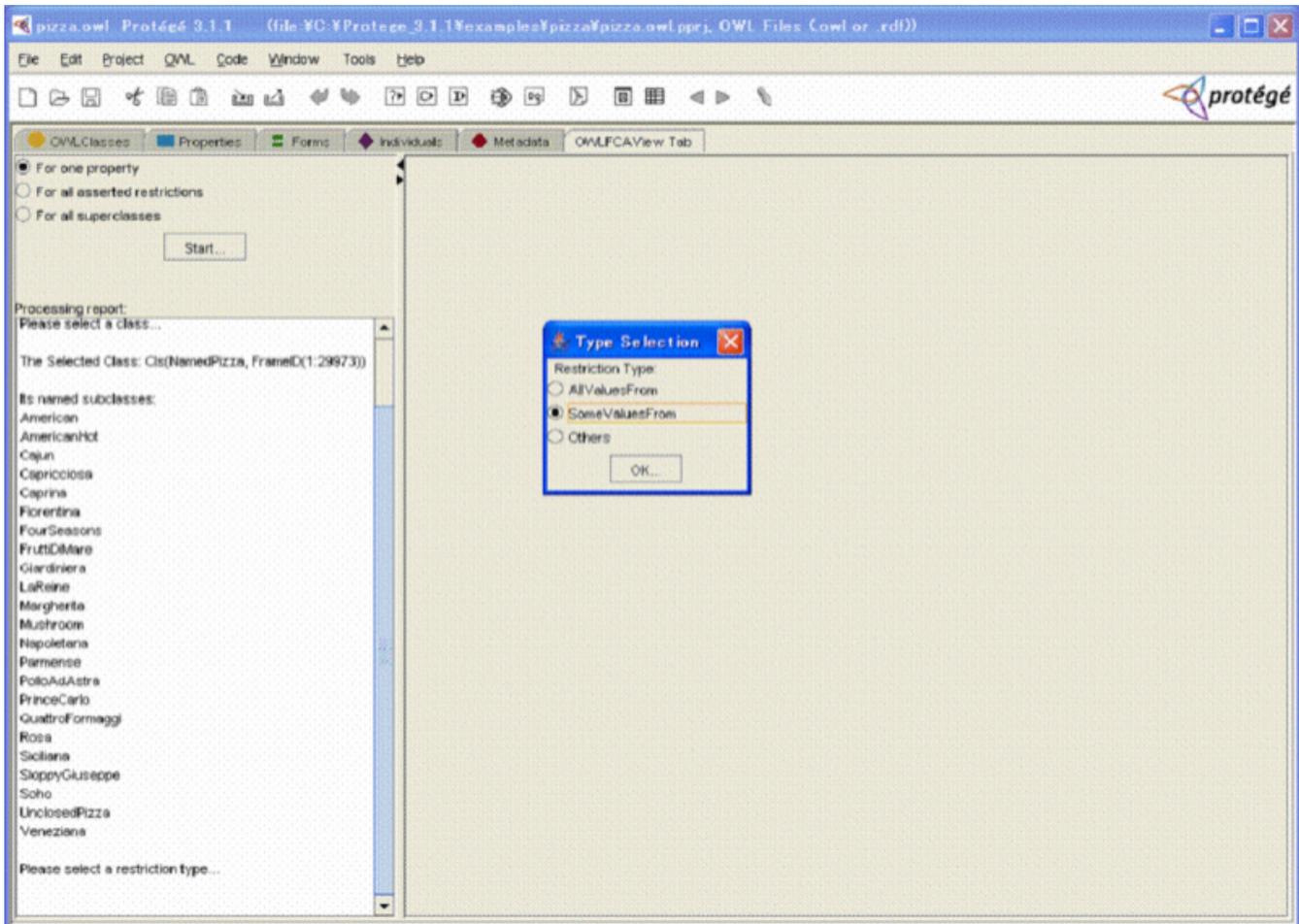
In left panel, there are three radio buttons for specification selection, a Start button, and a text area for processing guidance. To start processing, please select the first radio button For one property, then click the button Start and then you see the following.



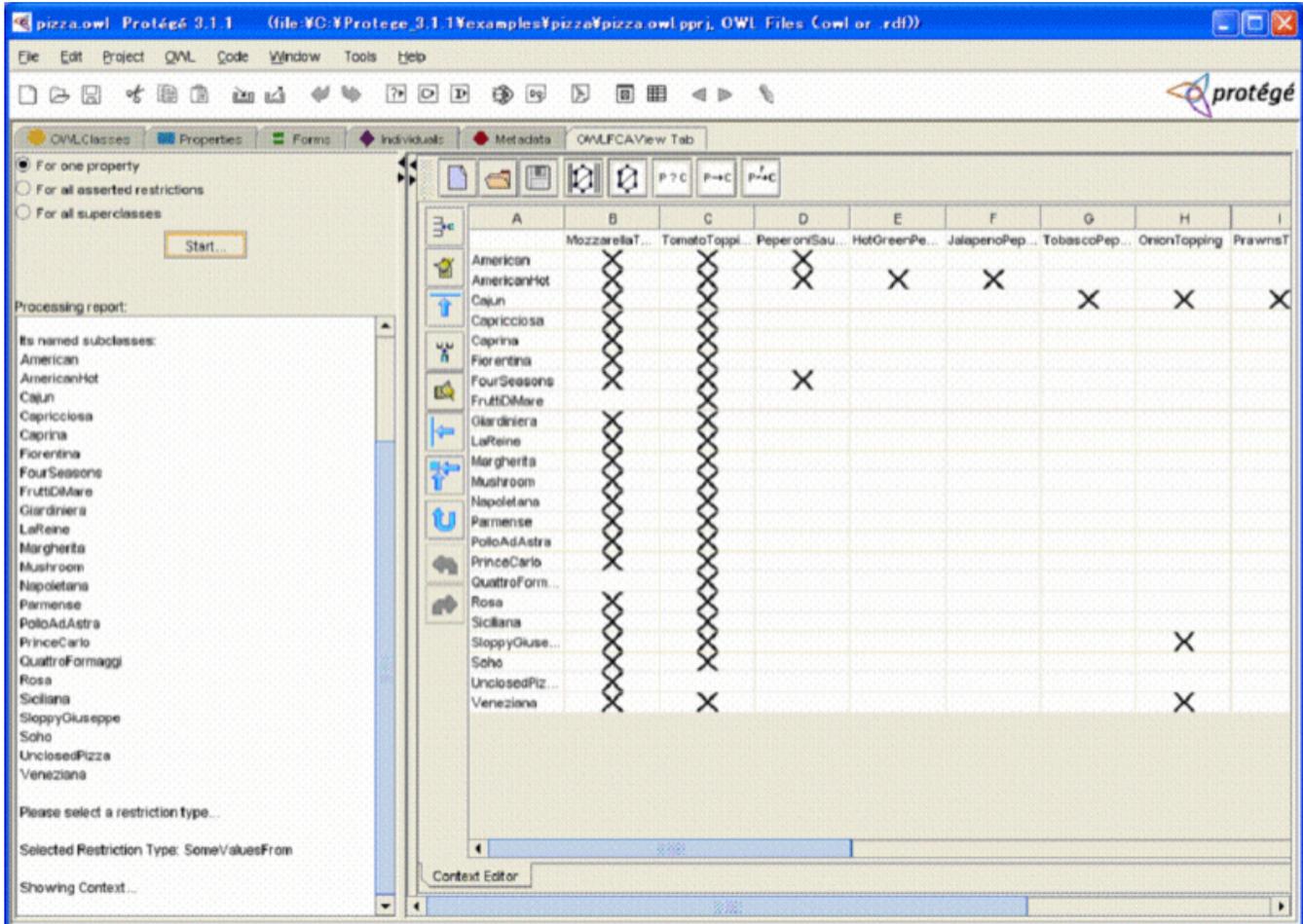
A list of the properties of the Pizza ontology is displayed in a panel for selection. Assume that we are interested in the property hasTopping, and select it. After clicking the OK button you see the following.



A list of the domain classes of property hasTopping is displayed in a panel for selection. Assume that we are interested in the class NamedPizza, and select it. Then after clicking OK button, you would see the following.

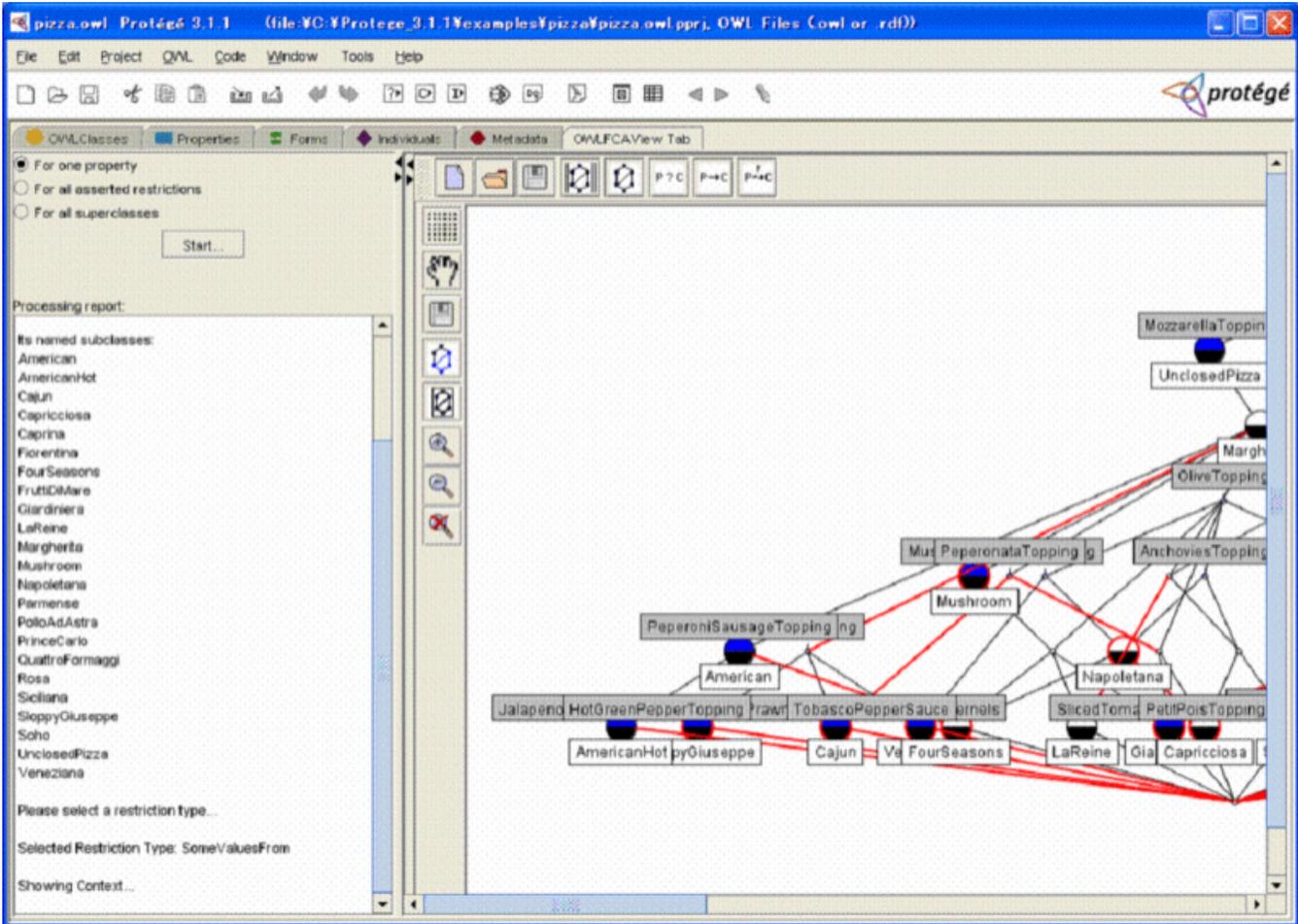


A panel for restriction type selection is displayed. In the text area in the left panel, you could also see some message about all subclasses of the class NamedPizza that is used as the formal objects. You could select one of three types including AllValuesFrom, SomeValuesFrom and Others. Assume that we are interested in the type SomeValuesFrom and select it. Then after clicking the OK button, you see a formal context was formed in a cross table as follows.

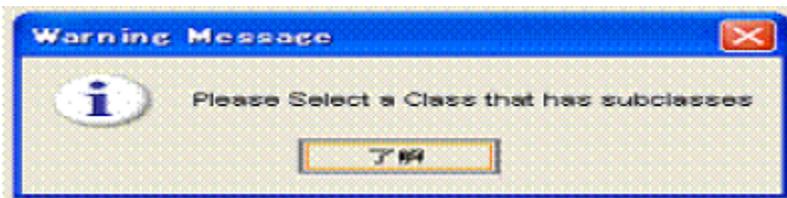


In this formal context, all subclasses of selected domain class NamedPizza are used as formal objects and all fillers of the restriction SomeValuesFrom of the property hasTopping are used as formal attributes. The cross markers in the table show the binary relations between objects and attributes.

By clicking the fifth button above the cross table (Build Lattice), you see a lattice diagram of the formal context as follows.



For selection of the domain class, the users should select a class that has subclasses because we use the subclasses of a selected class as the formal objects. When a class without the subclasses is selected, a warning panel is displayed to warn the users to select another class.

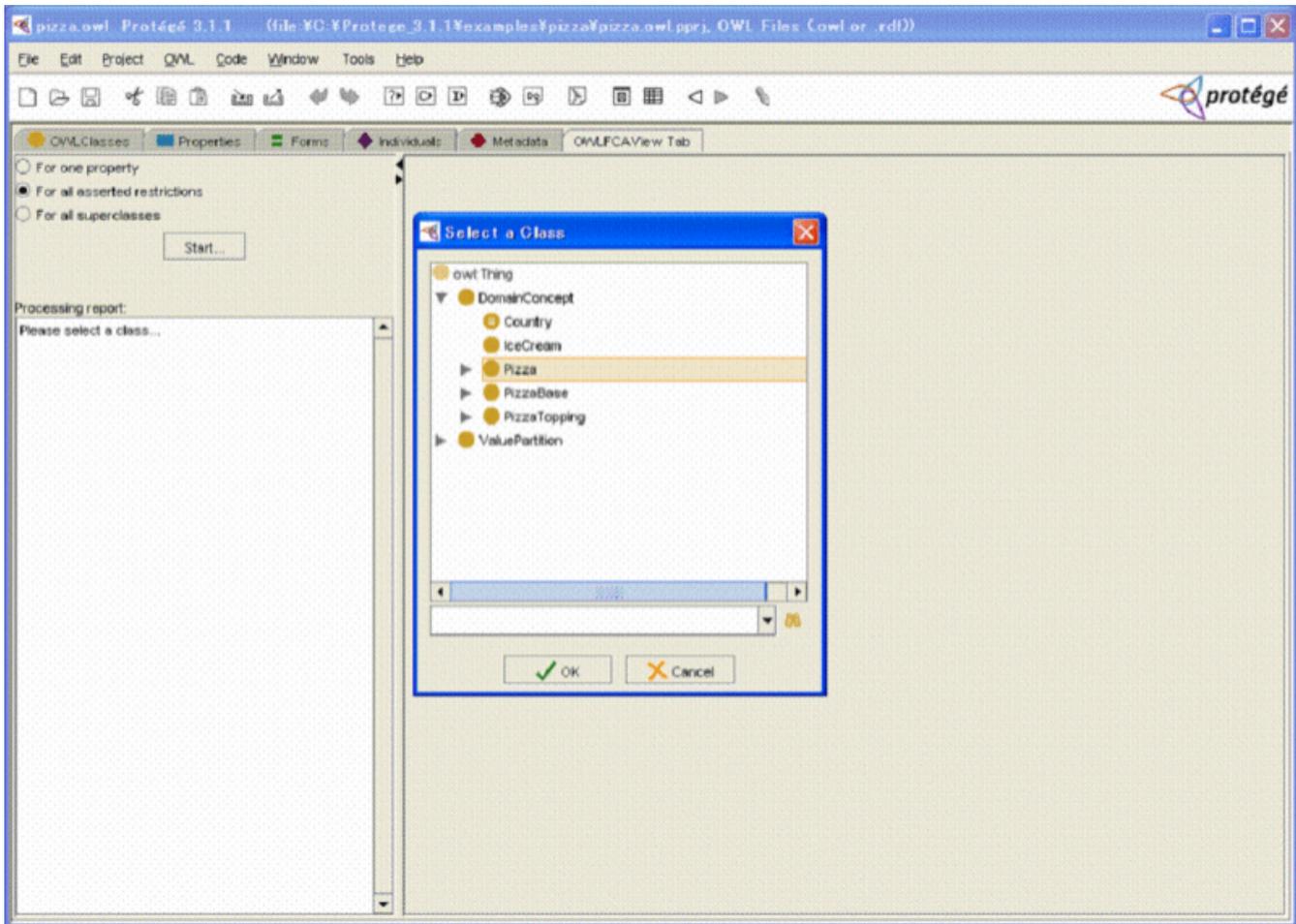


For the restriction type AllValuesFrom selection, the unionRangeClasses of the selected property are used as the formal attributes.

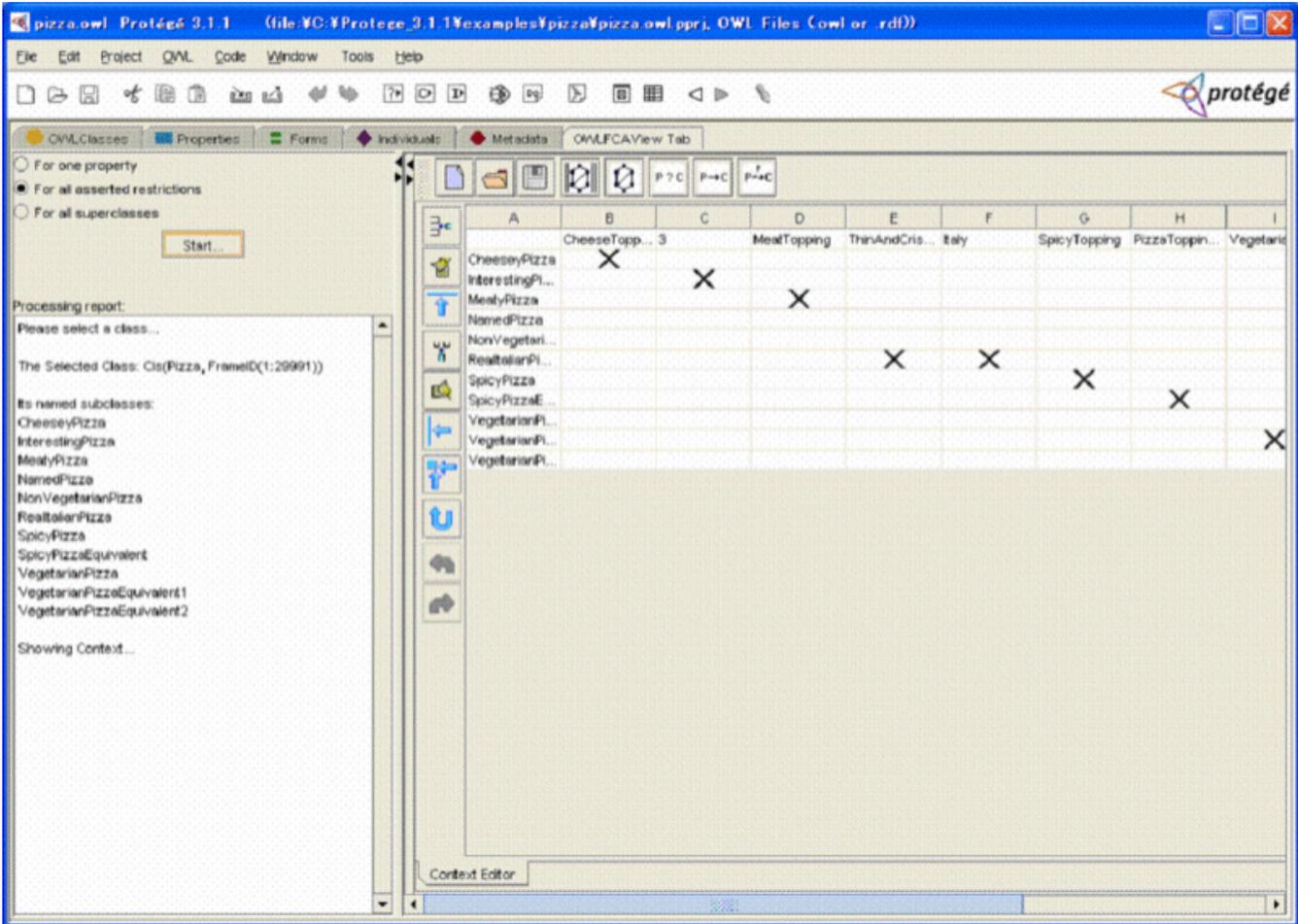
For the restriction type Others selection, the fillers of the restrictions that are not AllValuesFrom type and SomeValuesFrom type of the selected property are used as the formal attributes.

For all asserted restrictions

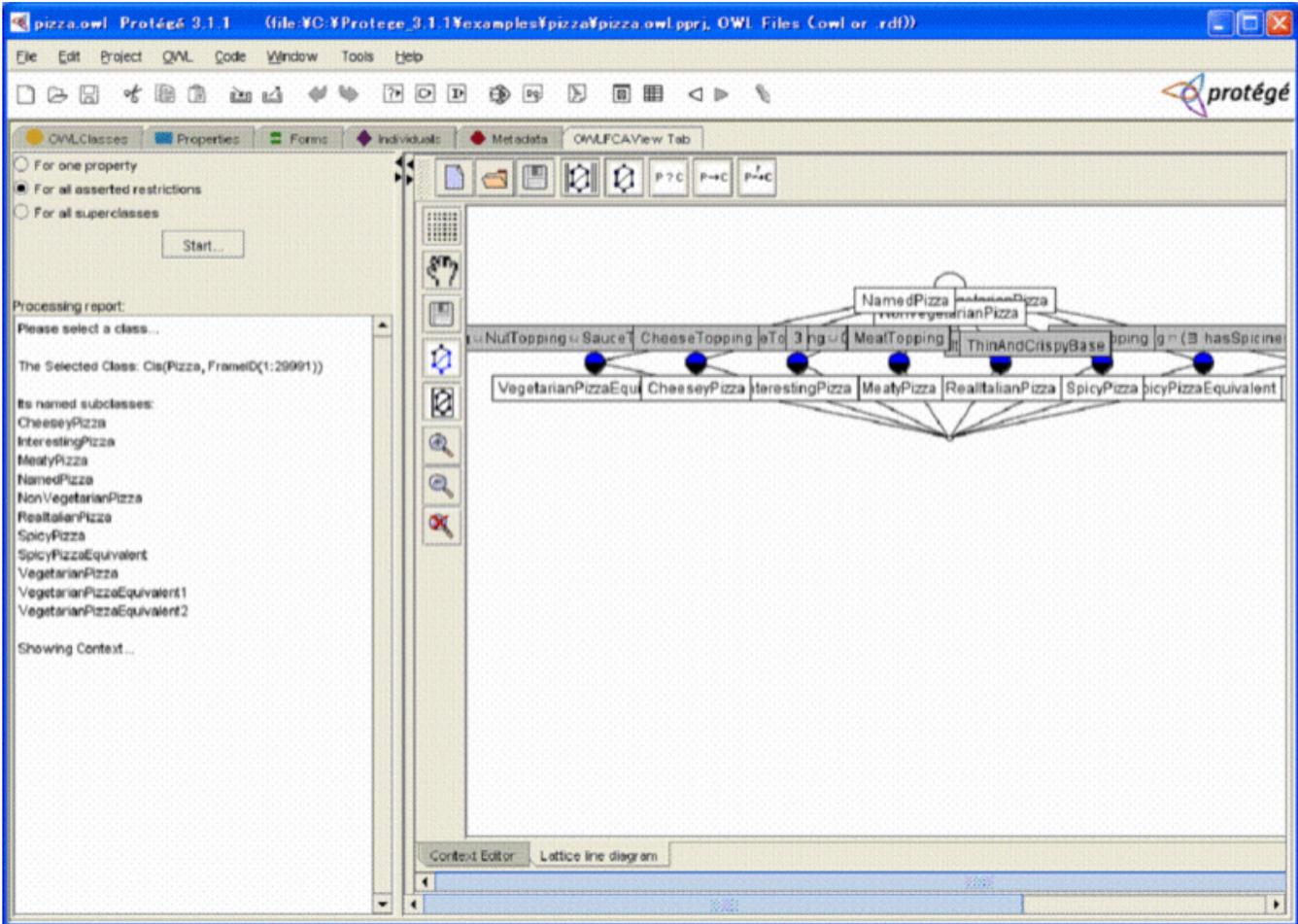
After selecting the radio button for all asserted restrictions and clicking the button Start, you see a screen like the following.



After selecting a class, e.g. Pizza here, and clicking the OK button, you see a screen like the following. The figure displays a cross table representing a formal context. The formal context is formed by all subclasses of Pizza class as the formal objects and the fillers of all asserted restrictions defined for the classes as the formal attributes.

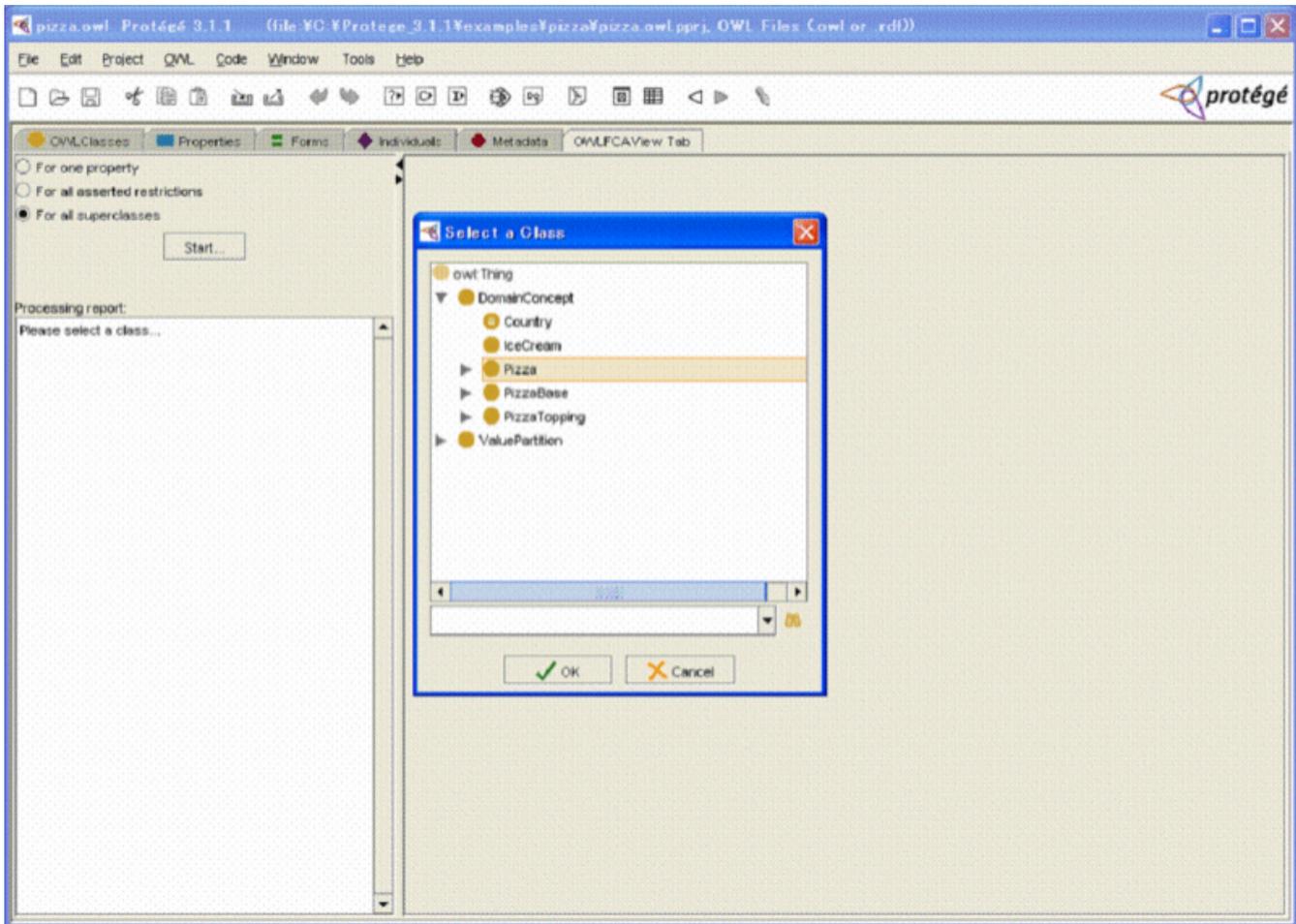


After clicking the Build Lattice button, you could see a lattice diagram, based on the formal context shown in the cross table.



Specification 3: For all superclasses

After selecting the radio button For all superclasses and clicking the button Start, you see a screen like the following.



After selecting a class, e.g. Pizza here, and clicking the OK button, you see a screen like the following. The illustration shows a cross table representing a formal context. The formal context is formed by all subclasses of Pizza class as the formal objects and all superclasses of each selected class as the formal attributes.

protégé 3.1.1 (file: YC:\Protege_3.1.1\examples\pizza\pizza.owl.pprj, OWL Files (owl or .rdf))

File Edit Project OWL Code Window Tools Help

OWLClasses Properties Forms Individuals Metadata OWLFCATView Tab

For one property
 For all asserted restrictions
 For all superclasses

Start...

Processing report:
Please select a class...

The Selected Class: `Class(Pizza, FrameID(1:29991))`

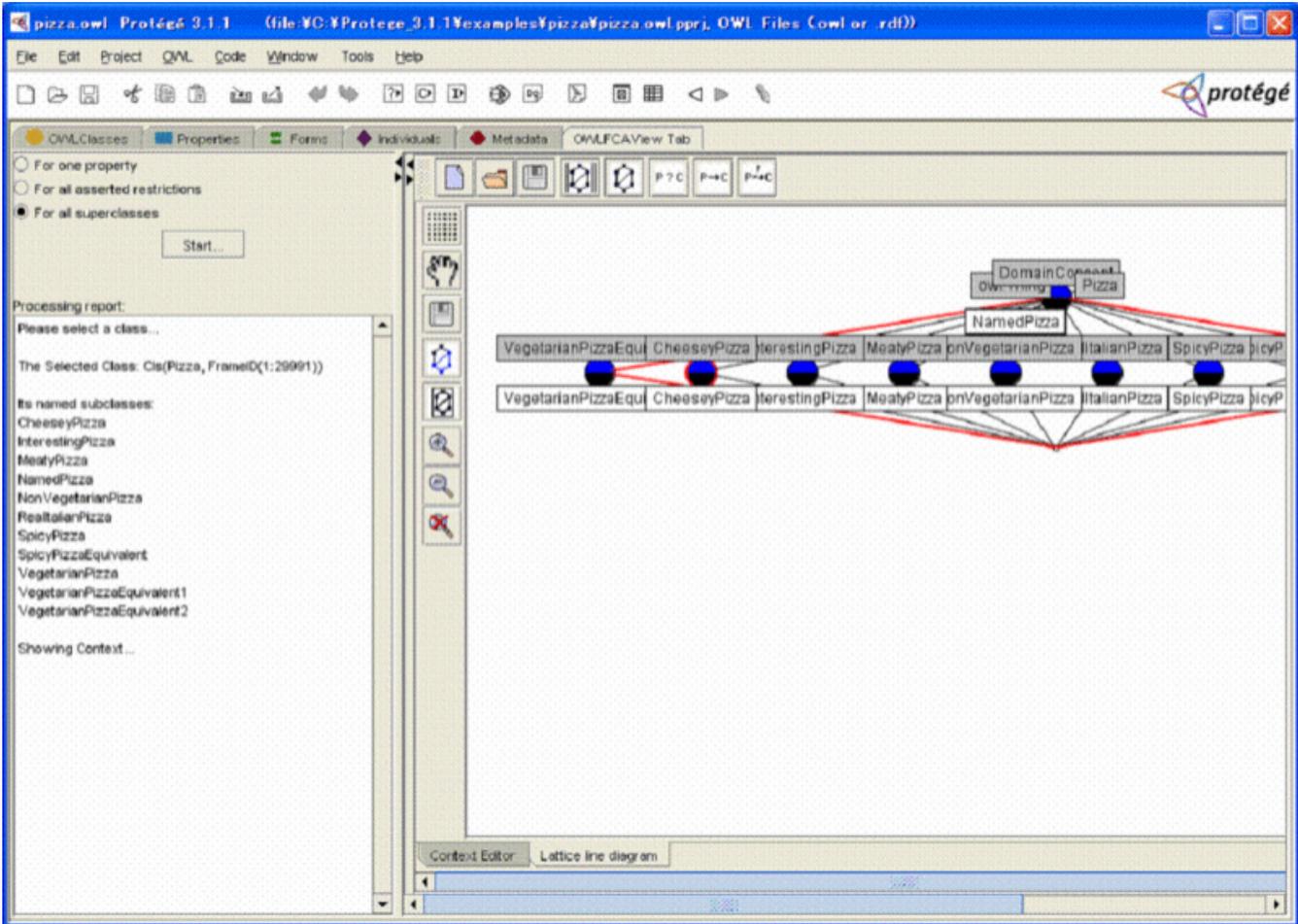
Its named subclasses:
 CheesyPizza
 InterestingPizza
 MeatyPizza
 NamedPizza
 NonVegetarianPizza
 RealItalianPizza
 SpicyPizza
 SpicyPizzaEquivalent
 VegetarianPizza
 VegetarianPizzaEquivalent1
 VegetarianPizzaEquivalent2

Showing Context...

	A	B	C	D	E	F	G	H	I
CheesyPizza		X							
InterestingPi...			X	X	X	X			
MeatyPizza			X	X	X		X		
NamedPizza			X	X	X				
NonVegetari...			X	X	X			X	
RealItalianPi...			X	X	X				X
SpicyPizza			X	X	X				
SpicyPizzaE...			X	X	X				
VegetarianPi...			X	X	X				
VegetarianPi...			X	X	X				

Context Editor

After clicking the Build Lattice button, you could see a lattice diagram is displayed based on the formal context shown in the cross table.



In current step, we think we only developed a basic model. Several advanced models are under active development. Any questions, please contact Guoqian Jiang (Jiang.Guoqian@mayo.edu) or Harold R. Solbrig (Solbrig.Harold@mayo.edu).
 May 12, 2006