

# Summarizing the Structure of the Pizza Ontology: Ontology Development with Partial-area Taxonomies and the Ontology Abstraction Framework

---

**Christopher Ochs, PhD, James Geller, PhD, and Yehoshua Perl, PhD**

Structural Analysis of Biomedical Ontologies Center (SABOC)

Department of Computer Science

New Jersey Institute of Technology (NJIT)

[cro3@njit.edu](mailto:cro3@njit.edu), [geller@njit.edu](mailto:geller@njit.edu), [perl@njit.edu](mailto:perl@njit.edu)

**Version 1.0 (Released: April 5, 2016)**

## 1 Introduction

The Pizza Ontology Tutorial, which was written at Manchester University, is the definitive guide to developing Web Ontology Language (OWL) format ontologies using the Protégé ontology editing tool. The Pizza Ontology Tutorial reviews the various features available in OWL and Protégé. The current version of the Pizza Ontology Tutorial (Version 1.3) is available at:

<http://owl.cs.manchester.ac.uk/publications/talks-and-tutorials/protg-owl-tutorial/> in the file [http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4\\_v1\\_3.pdf](http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf)

In this document, we build on the Pizza ontology tutorial. We introduce the reader to the theory and applications of *Partial-area Taxonomies*. A *Partial-area Taxonomy* is an algorithmically derived summary of an ontology. Specifically, a Partial-area Taxonomy represents major groups of structurally and semantically similar classes of an ontology. Partial-area Taxonomies are usually much smaller than the ontologies that they summarize. Below we list some relevant publications:

Halper, Michael, Huanying Gu, Yehoshua Perl, and Christopher Ochs. "Abstraction networks for terminologies: Supporting management of "big knowledge"." *Artificial intelligence in medicine* 64, no. 1 (2015): 1-16.

Ochs, Christopher, Ankur Agrawal, Yehoshua Perl, Michael Halper, Samson W. Tu, Simona Carini, Ida Sim, Natalya Fridman Noy, Mark A. Musen, and James Geller. "Deriving an abstraction network to support quality assurance in OCR." In *AMIA*. 2012.

Ochs, Christopher, Zhe He, Yehoshua Perl, Sivaram Arabandi, Michael Halper, and James Geller. "Choosing the Granularity of Abstraction Networks for Orientation and Quality Assurance of the Sleep Domain Ontology." In *ICBO*, pp. 84-89. 2013.

He, Zhe, Christopher Ochs, Ankur Agrawal, Yehoshua Perl, Dimitris Zeginis, Konstantinos A. Tarabanis, Gai Elhanan, Michael Halper, Natasha F. Noy, and James Geller. "A family-based framework for supporting quality assurance of biomedical ontologies in BioPortal." In *AMIA*. 2013.

Ochs, Christopher, Zhe He, Ling Zheng, James Geller, Yehoshua Perl, George Hripcsak, and Mark A. Musen. "Utilizing a structural meta-ontology for family-based quality assurance of the BioPortal ontologies." *Journal of Biomedical Informatics* 61 (2016): 63-76.

This guide describes the use of our software tool, called the **Ontology Abstraction Framework (OAF) Protégé Plugin**. The guide follows the Manchester Pizza Ontology Tutorial step by step and illustrates how each step of developing the Pizza Ontology is reflected in the Partial-area Taxonomy created from the Pizza Ontology using our software tool. Images in this guide are screen dumps from the **OAF Protégé Plugin**. For brevity we will often omit the words Protégé Plugin and only talk about the **OAF** or OAF tool.

## 1.1 Required Materials

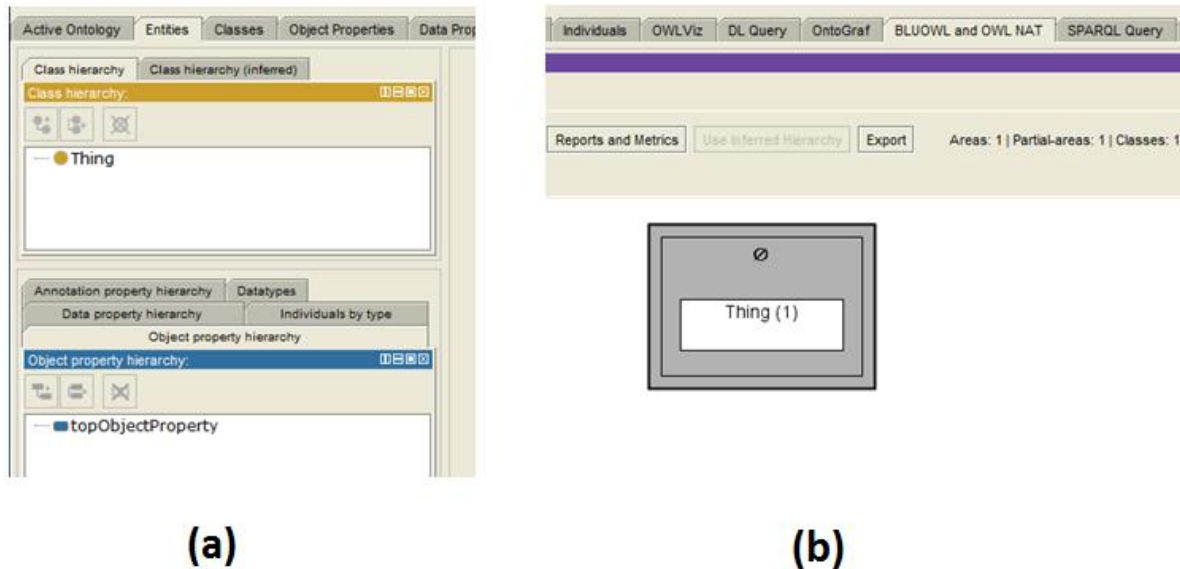
For this tutorial the following materials are required:

- Desktop Protégé, available for download at <http://protege.stanford.edu/>
- The Pizza Ontology Tutorial, available at <http://owl.cs.manchester.ac.uk/publications/talks-and-tutorials/protg-owl-tutorial/>
- The Ontology Abstraction Framework (OAF) Protégé Plugin (a substantial extension and reimplementaton of the BLUOWL tool), currently available from Christopher Ochs ([cro3@njit.edu](mailto:cro3@njit.edu)).

## 2 Pizza Ontology Partial-area Taxonomy

Throughout this Section we follow Chapter 4 and Chapter 5 in the Pizza Ontology Tutorial (henceforth, **POT**). Certain sections from the POT that cover editing operations that do not affect Partial-area Taxonomies are omitted. We explicitly identify which exercise(s) in the POT are being illustrated and how each exercise affects the Pizza Ontology's Partial-area Taxonomy. For illustrative purposes, we will occasionally perform the Pizza Ontology exercises in a different order than the order given in the POT.

### 2.2 Creating a New Ontology



**Figure 2.1 (a)** A new, empty ontology. **(b)** The Partial-area Taxonomy for the empty ontology.

**Exercise 2** in the POT requires the user to create a new, empty ontology in Protégé. (Even an empty ontology contains the root class “Thing.”) Figure 2.1 (a) illustrates the result of creating the empty ontology in Protégé and Figure 2.1(b) illustrates the Partial-area Taxonomy derived from the empty ontology. The ontology consists of only one class, the *Thing* class, and no properties. The Partial-area Taxonomy consists of one *area*. The name of this area is  $\emptyset$  (pronounced “empty”) and it appears in Fig. 2.1(b) as a gray box. Inside of the area is one *partial-area* with the name Thing (1) and it appears as a white box. The number (1) indicates that there is only one class that is summarized by this partial-area, namely the root class *Thing*.

**Exercise 3** in the POT is to create a comment annotation on the ontology. This does not affect the Partial-area Taxonomy, thus, we skip this exercise.

#### Essential Theory

Starting with the most general explanation, an *area* is a group of classes with exactly the same properties (roles, relationships, etc.) in the ontology. In the Partial-area Taxonomy, such a group is represented by one single *area node*. For brevity, whenever we talk about the Partial-area Taxonomy, then “area” means the “area node.”

We distinguish between different kinds of areas. We will now discuss *domain-defined areas*. The

resulting Partial-area Taxonomy is then called the *domain-defined Partial-area Taxonomy*.

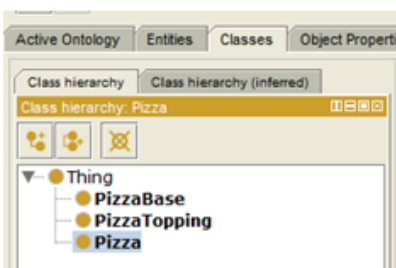
In a *domain-defined Partial-area Taxonomy*, an area summarizes the set of classes that are either explicitly defined as the domain of an object property (via its `rdfs:Domain` axiom) or implicitly defined through inheritance of the object property.

The *name* of an area node is equivalent to the *set* of all object properties of the classes that are summarized by the area. If an area node has no object properties, then its name is the empty set {}, also written as  $\emptyset$ .

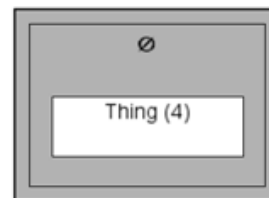
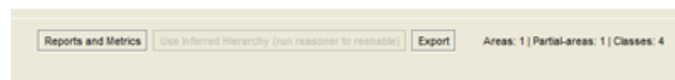
The theory of partial-areas will be explained at an appropriate point, as it requires more background knowledge.

- In Figure 2.1(a) there are no object properties defined in the ontology. Thus, the one class in the ontology, *Thing*, is summarized by the area named  $\emptyset$ , representing all classes that are not in the domain of any object property.
- The white partial-area box embedded within the gray  $\emptyset$  box, named Thing (1), represents the subhierarchy of classes rooted at Thing which are not the domain of any object property. The subhierarchy in Figure 2.1(b) consists only of the root class. The (1) indicates the cardinality of the partial-area.

## 2.3 Named Classes



(a)



(b)

**Figure 2.2 (a)** Three classes, *Pizza*, *PizzaBase*, and *PizzaTopping*, have been added to the ontology. **(b)** These three classes are summarized by the Thing (4) partial-area since no properties have yet been defined within the ontology.

**Exercise 4** in the POT focuses on creating a new class in the ontology, *Pizza*, and then adding two sibling classes, *PizzaTopping* and *PizzaBase*. The result of this is illustrated in Figure 2.2(a).

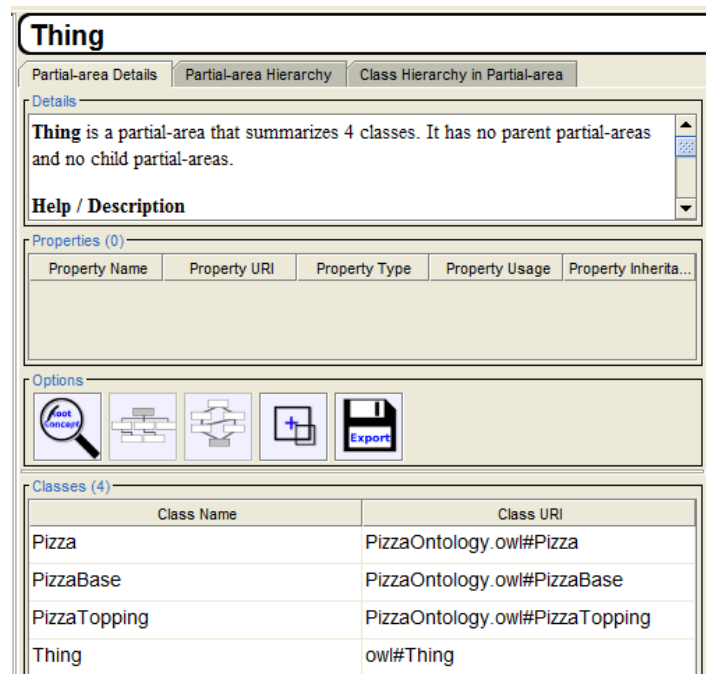
### Theory

The classes *Pizza*, *PizzaBase*, and *PizzaTopping* were defined as subclasses of *Thing*. No object properties were defined for the ontology in this step.

- The Thing (4) partial-area, represented as a white box within the gray  $\emptyset$  area, summarizes the subhierarchy of classes that are not in the domain of any object property. It is in the nature of the summarization that the three new classes are only represented by changing Thing (1) to Thing (4).
- Thus, the three newly added classes are summarized by the Thing (4) partial-area. For brevity, we are omitting the cardinality (...) from the name of a partial-area when it does not contribute to the discussion.

## Software

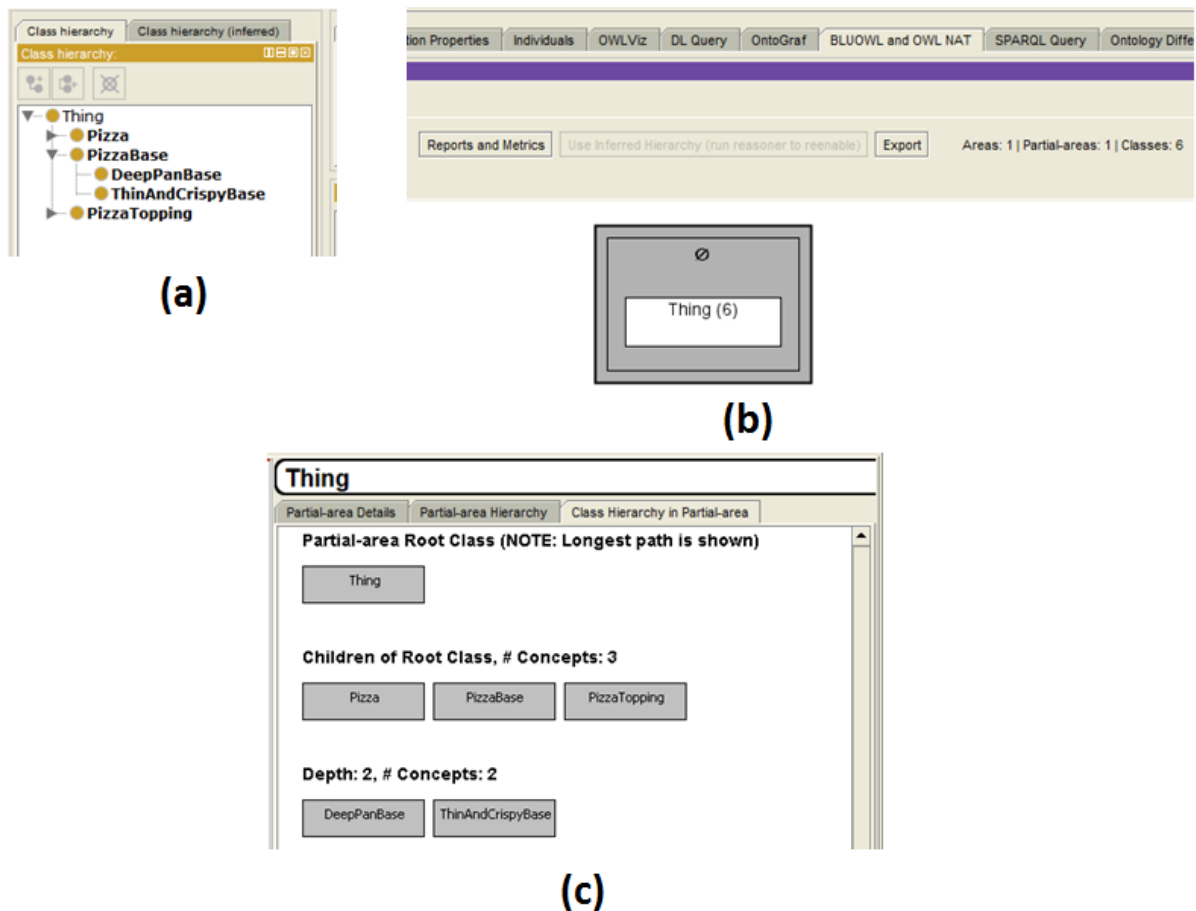
Within the **Ontology Abstraction Framework (OAF) Protégé Plugin**, clicking on a partial-area displays information about the selected partial-area in the Details Panel, on the right side of the screen. Figure 2.3 illustrates the information that is displayed when the Thing partial-area is selected. Detailed English language information about the partial-area is shown at the top of the box under **Details**. The list of classes summarized by the Thing (4) partial-area is shown in alphabetical order at the bottom under **Classes (4)**. Double clicking on a class in this list takes the user to the class definition/editing tab in Protégé.



**Figure 2.3** The Thing Partial-area Details panel.

**Exercise 5** in the POT makes the three Pizza Ontology classes created in Exercise 4 disjoint from one another. This step has no effect on the Partial-area Taxonomy. Thus, we skip the result of this exercise.

## 2.4 Create Class Hierarchy Tool



**Figure 2.4** (a) *DeepPanBase* and *ThinAndCrispyBase* were added to the ontology using the Create Class Hierarchy tool in Protégé. (b) The two new classes are summarized by changing Thing (4) to Thing (6). This is sufficient, since they are children of *PizzaBase* and no properties have been introduced into the ontology yet. (c) The hierarchy of classes summarized by the Thing partial-area, as displayed in the Ontology Abstraction Framework tool when the Thing partial-area is selected and the “Class Hierarchy in Partial-area” tab is chosen. This is the third tab from the left in Figure 2.3. Only the levels of the concepts are shown. Connection lines are omitted in Figure 2.4(c), because they quickly lead to overwhelming figures. However, parents and children of a class can be dynamically determined by clicking on it.

**Exercise 6** in the POT introduces the Create Class Hierarchy tool, which simplifies the process of creating new classes in an ontology. Two new subclasses of *PizzaBase*, *ThinAndCrispyBase* and *DeepPanBase*, are added to the ontology using this tool. The result is reflected in Figure 2.4(a).

### Theory

Again, in this step no properties have been added to the ontology. Thus, the two new classes added in this step, *DeepPanBase* and *ThinAndCrispyBase*, which were defined as

subclasses of *PizzaBase*, are also summarized by the Thing partial-area in the  $\emptyset$  area, as reflected in Figure 2.4(b), by changing the cardinality in parentheses to Thing (6).

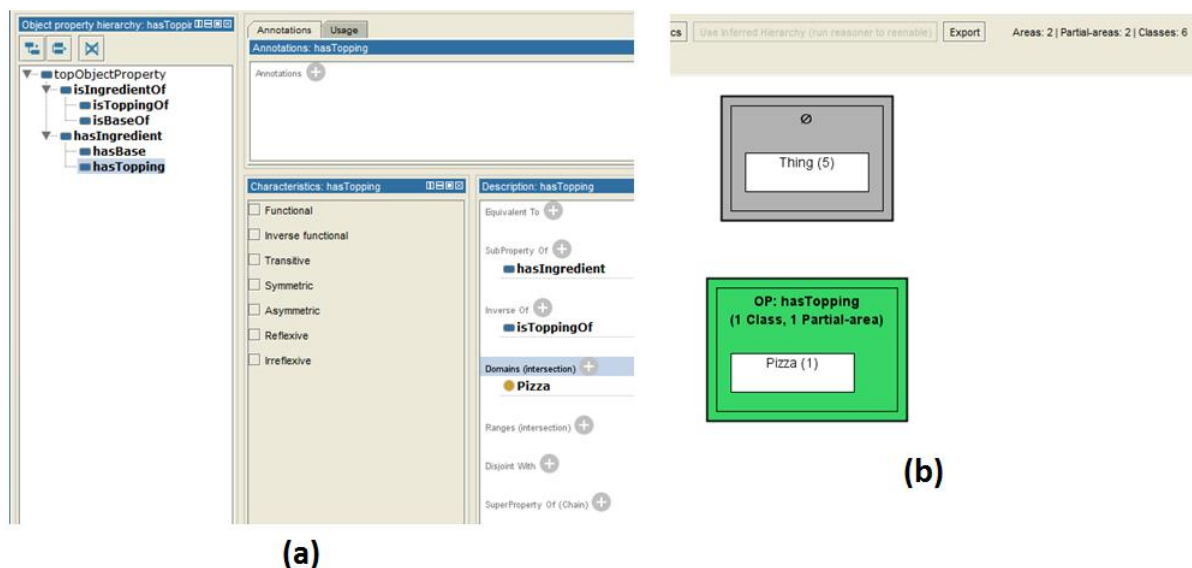
## Software

After selecting a partial-area in the **Ontology Abstraction Network Protégé Plugin**, the user can click on the “Class Hierarchy in Partial-area” tab to view the subhierarchy of classes summarized by the selected partial-area. Classes are organized in longest-path order. Single clicking on a class will highlight its parents and children (by changing their colors). Double clicking on a class will open its definition in Protégé’s class editor.

In this guide, we will perform **Exercise 7** after **Exercises 8, 9, and 10**, as this order better illustrates the functionality of the Partial-area Taxonomy display.

## 2.5 Object Property Domains

Domain-defined partial-area taxonomies (see Section 2.2 Theory) are derived according to the domains of the properties in an ontology. Thus, we draw special attention to the effects of the editing operations described in *Section 4.4 – Section 4.7* of the POT. In this section, we focus on what effects defining property domains in an ontology has on its *domain-defined Partial-area Taxonomy*.



**Figure 2.5 (a)** Six object properties have been added to the ontology. The *Pizza* class has been set as the domain of the *hasTopping* object property. **(b)** The Partial-area Taxonomy, based on the *rdfs:Domain* axioms of the object properties in the ontology.

In **Exercise 8** the reader creates the first object property in the Pizza Ontology: *hasIngredient*. In **Exercise 9** the reader creates two sub properties of *hasIngredient*: *hasTopping* and *hasBase*. In **Exercise 10** the reader creates the inverse properties of these three object properties. The results of performing these three exercises are shown in Figure 2.5(a).

Section 4.6 of the POT focuses on the characteristics of properties, which do not affect partial-area taxonomies, and thus we skip to Section 4.7 and **Exercises 13** and **14**.

In **Exercise 13** the user sets the range of *hasTopping* to *PizzaTopping*. Since partial-area taxonomies are derived according to the domain of a property, this exercise has no effect on the Partial-area Taxonomy.

**Exercise 14** instructs the reader to set the *domain* of *hasTopping* as the class *Pizza*, as illustrated in Figure 2.5(a). After **Exercise 14** has been completed *the Partial-area Taxonomy undergoes its first significant change*.

The Partial-area Taxonomy now consists of **two areas**,  $\emptyset$  and  $\{hasTopping\}$ . The new area named  $\{hasTopping\}$  summarizes exactly the classes with the object property *hasTopping*. Each of the two areas contains exactly one partial-area. The new partial-area, *Pizza* (1), embedded within the  $\{hasTopping\}$  area, summarizes one class, namely *Pizza*. Note that the class *Pizza* has migrated down from the partial-area *Thing* (6), which has now become *Thing* (5).

### Theory

In this step, several object properties were added to the ontology. However, in this set of Exercises only the *hasTopping* object property was assigned a domain. Specifically, the class *Pizza* was explicitly defined as being the domain of *hasTopping*. Thus, there exists one class which is in the domain of the *hasTopping* object property. This information is reflected in the Partial-area Taxonomy by the new area, colored green, named  $\{hasTopping\}$ .

Partial-areas are consistently displayed as white. The assignment of colors to areas is arbitrary, except that all areas with the same number of object properties must have the same color. There can be only one area with no properties, namely  $\emptyset$ .

- The area node  $\{hasTopping\}$  is displayed below the area node  $\emptyset$ , because area nodes are organized into levels according to their numbers of object properties.
- There is one partial-area in the  $\{hasTopping\}$  area named *Pizza* (1), because the class *Pizza* is the explicit domain of *hasTopping*.
- All classes that are descendants of the class *Pizza*, and not in the domain of any other object properties, will be summarized by the *Pizza* partial-area.
- As of this step, all other classes in the ontology are not assigned as a property's domain and are not the descendant of a class that is assigned a domain, thus they are still summarized by the *Thing* (5) partial-area node in the  $\emptyset$  area node.

An area may contain (a) classes that have no parents in this area and (b) classes that have parents in this area. A class that has no parents in its area is called a *root of the area* (briefly: root). Every root has its own semantics and is displayed as its own partial-area node.

A partial-area node summarizes a root class and all its descendant classes that are strictly in the same area as the root class. A class may belong to more than one partial-area.

## Software

Clicking on the Pizza (1) partial-area node displays a screen similar to that of Figure 2.6. However, the classes summarized by the Pizza partial-area (only one class, *Pizza*, as of this step) are in the domain of one object property, *hasTopping*. This information would be displayed in the middle of the screen, under the [Properties](#) title. The information displayed here includes

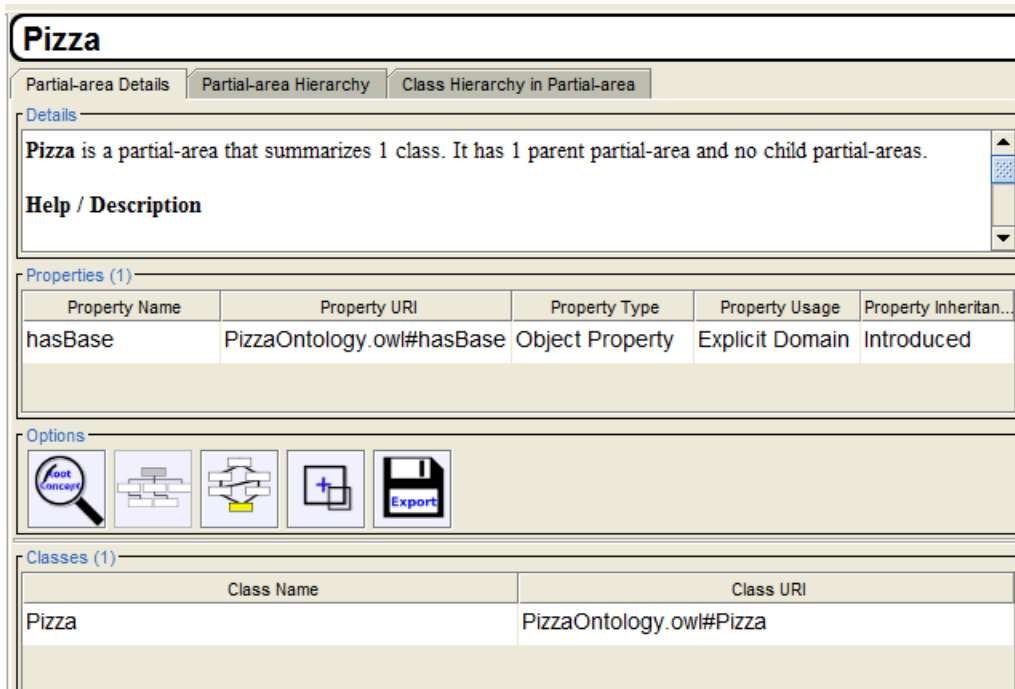
- the name of the property,
- the unique identifier of the property,
- the type of the property (see below this bullet list),
- how the property was used (assigned a specific domain or used in a restriction),
- and if the property is explicitly defined at the root of the partial-area or if it was inherited from a higher ancestor.

The type of a property may be

- an object property
- a data property

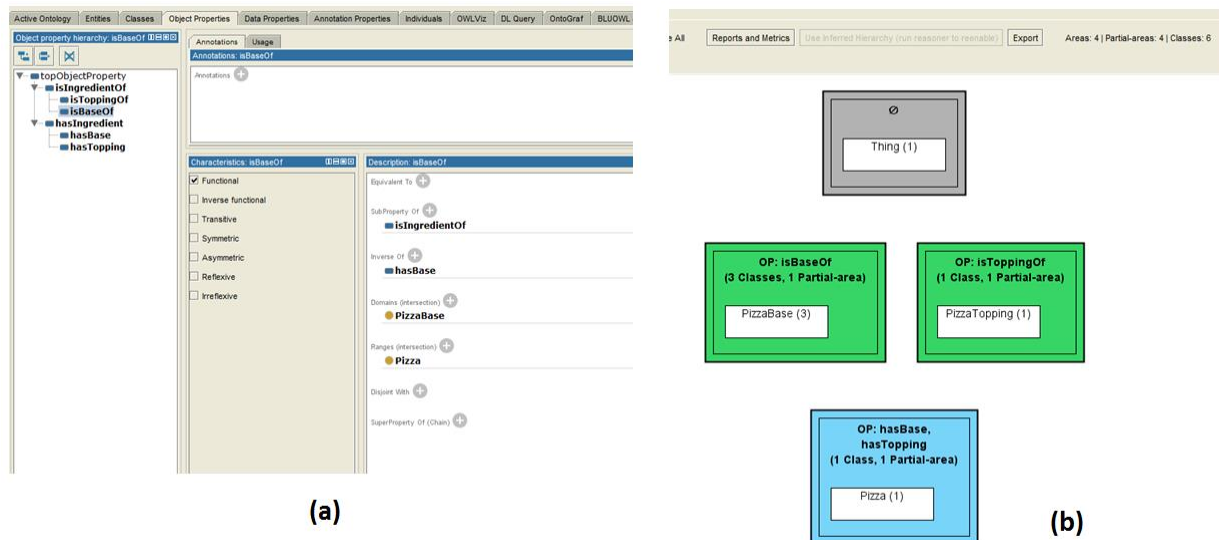
In essence, a data property is “local” data of the class, typically a string or a number, while an object property refers to another class (“a smart pointer”).

Double clicking on a property in the list takes the user to the object property (or data property) definition/editing tab in Protégé. This shows the tight integration between the original Protégé functionality and the **Partial-area Taxonomy Protégé Plugin** described in this guide.



**Figure 2.6** Information about the Pizza partial-area.

**Exercise 15** has the reader set the domains of the *hasBase*, *isBaseOf*, and *isToppingOf* object properties as *Pizza*, *PizzaBase*, and *PizzaTopping*, respectively.



**Figure 2.7 (a)** The domains and ranges have been set for the various properties introduced in Figure 2.5(a). **(b)** The Partial-area Taxonomy after all of the object property domains have been set.

Like **Exercise 14**, the editing operations in **Exercise 15** have a significant effect on the Partial-area Taxonomy. The Partial-area Taxonomy now consists of four area nodes:  $\emptyset$ ,  $\{isBaseOf\}$ ,  $\{isToppingOf\}$ , and  $\{hasBase, hasTopping\}$ .

### Theory

In this step several object properties were assigned domains. Let us describe the effect of each change.

- *PizzaTopping* was explicitly assigned as the domain of the *isToppingOf* object property. Thus, a new area  $\{isToppingOf\}$  is introduced into the Partial-area Taxonomy. There is one partial-area, *PizzaTopping*. This partial-area summarizes only one class in the area  $\{isToppingOf\}$ , since only *PizzaTopping* is defined as the domain of *isToppingOf*.
- *PizzaBase* was explicitly assigned as the domain of *isBaseOf*. Thus, a new area,  $\{isBaseOf\}$  is introduced to the Partial-area Taxonomy. The two subclasses of *PizzaBase*, namely *ThinAndCrispyBase* and *DeepPanBase*, are implicitly in the domain of *isBaseOf* and are not in the domain (either explicitly or implicitly) of any other object property. Thus, the *PizzaBase* (3) partial-area in  $\{isBaseOf\}$  summarizes these three classes.
- Both areas  $\{isToppingOf\}$  and  $\{isBaseOf\}$  are colored green and are shown at the same level, because these areas summarize classes that are in the domain of exactly one object property.
- *Pizza* was explicitly defined as the domain of *hasBase*, in addition to still being the domain of *hasTopping*. Thus, the *Pizza* class is now summarized by an area named  $\{hasBase, hasTopping\}$ . This area is colored blue and shown below  $\{isToppingOf\}$  and  $\{isBaseOf\}$  because this area has two object properties, compared to  $\{isToppingOf\}$  and  $\{isBaseOf\}$ , which have only one. There is one partial-area, *Pizza* (1), which still summarizes only one class *Pizza* because no subclasses of *Pizza* have been added.
- In diagrams of Partial-area Taxonomies we omit the curly brackets to save space. Therefore, the name of  $\{hasBase, hasTopping\}$  appears as *hasBase, hasTopping*.

Figure 2.7(b) hints at the fact that the area nodes themselves are forming a hierarchy. This is shown by laying out the area nodes in levels. Once again, to avoid overwhelming pictures, the links between area nodes and partial-area nodes are normally not shown.

Thus, an area node may have parent area nodes and child area nodes. Similarly, a partial-area may have parent partial-areas and child partial-areas.

### Software

Figures 2.8 and 2.9 illustrate the kinds of information displayed about the Partial-area Taxonomy within the Ontology Abstraction Framework tool. Figure 2.8 illustrates the “Areas in Partial-area Taxonomy” tab, which is accessible when no partial-areas and no areas have been selected within the ontology.

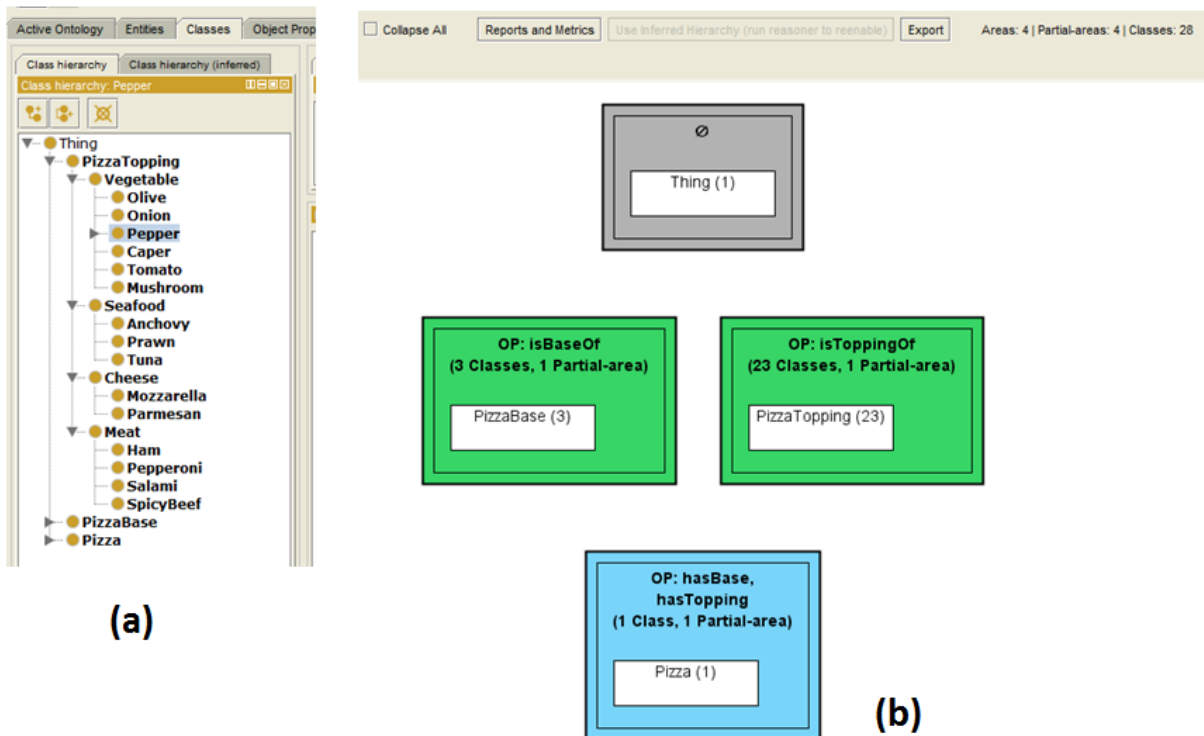
Summary Partial-area Taxonomy Levels Areas in Partial-area Taxonomy			
Areas (4 total)			
Area	# Partial-areas	# Classes	# Overlapping Classes
root area (no propertie	1	1	0
{isBaseOf}	1	3	0
{isToppingOf}	1	1	0
{hasBase hasTopping}	1	1	0

**Figure 2.8** Within the Ontology Abstraction Framework tool one can view which areas exist and how many classes (and partial-areas) are in each area.

Thing				
Partial-area Details Partial-area Hierarchy Class Hierarchy in Partial-area				
Root's Parent Classes Partial-areas				
Parent Class	Parent Class URI	Parent Partial-area	# Classes in Pare...	Area
Child Partial-areas				
Child Partial-area	# Classes		Area	
Pizza	1		hasBase hasTopping	
PizzaBase	3		isBaseOf	
PizzaTopping	1		isToppingOf	

**Figure 2.9** Clicking on the Thing partial-area and selecting the “Partial-area Hierarchy” tab displays information about the selected partial-area’s child partial-areas, each of which are in an area that has one or more additional properties.

Figure 2.9 illustrates the “Partial-area Hierarchy” tab, which is accessible when a partial-area is selected with a mouse click. In this example, the Thing (1) partial-area was selected. There are three partial-areas which are child partial-areas of Thing (1): Pizza (1), PizzaBase (3), and PizzaTopping (3).



**Figure 2.10 (a)** The new subhierarchy of classes added under *PizzaTopping*. Twenty-two kinds of pizza toppings were added to the ontology. **(b)** The Partial-area Taxonomy after the pizza topping classes were added. All of the new classes are summarized by the *PizzaTopping (23)* partial-area, since they are implicitly in the domain of *isToppingOf*. At this point it is becoming, for the first time in this guide, clear that the Partial-area Taxonomy can substantially simplify the display of an ontology. All classes “hidden” inside of *PizzaTopping (23)* have the same object properties and are semantically similar, as they are all pizza toppings. However, if desired, the user can drill down to display those pizza toppings.

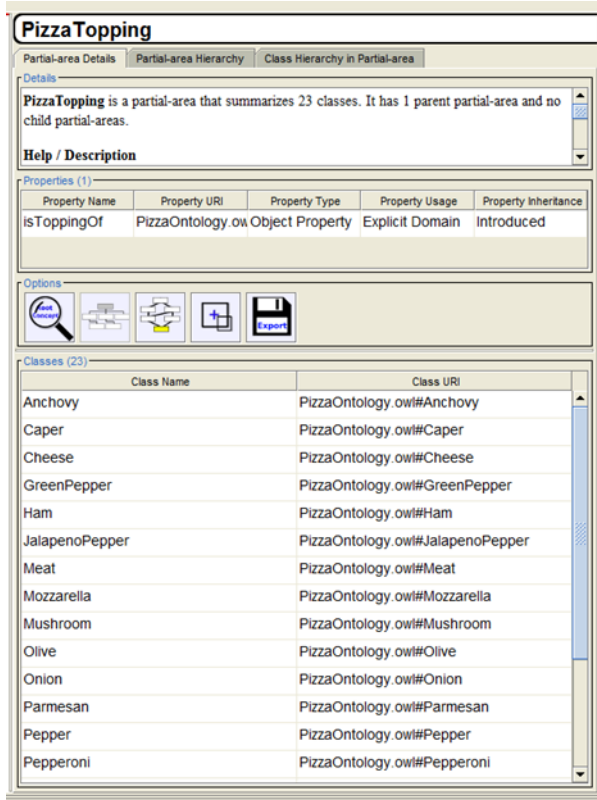
We now return to **Exercise 7**, where the reader uses the Create Class Hierarchy tool to add various kinds of pizza toppings to the Pizza Ontology. Figure 2.10 (a) illustrates the new subhierarchy of classes. Figure 2.10 (b) illustrates the Partial-area Taxonomy after all of the topping classes have been added to the ontology.

### Theory

All of the topping classes that were added to the ontology are defined as descendants of the *PizzaTopping* class. Thus, they are implicitly in the domain of the *isToppingOf* object property. Additionally, as of this step, these new classes are not the domain of any additional object properties. Thus, all of the new pizza topping classes are summarized by the *PizzaTopping (23)* partial-area in the *{isToppingOf}* area. This is reflected by the cardinality (23) shown after “*PizzaTopping*.” Figure 2.11(b) below shows that the summary that is achieved by the Partial-area Taxonomy goes well beyond a summary based on “levels of descendants,” because it takes structure and semantics of the ontology into account.

## Software

Figure 2.11(a) shows the information which is displayed when clicking on the PizzaTopping (23) partial-area. Additionally, the hierarchy of classes summarized by the PizzaTopping (23) partial-area is shown in Figure 2.11(b).



**PizzaTopping**

Partial-area Details | Partial-area Hierarchy | Class Hierarchy in Partial-area

**Details**

PizzaTopping is a partial-area that summarizes 23 classes. It has 1 parent partial-area and no child partial-areas.

**Help / Description**

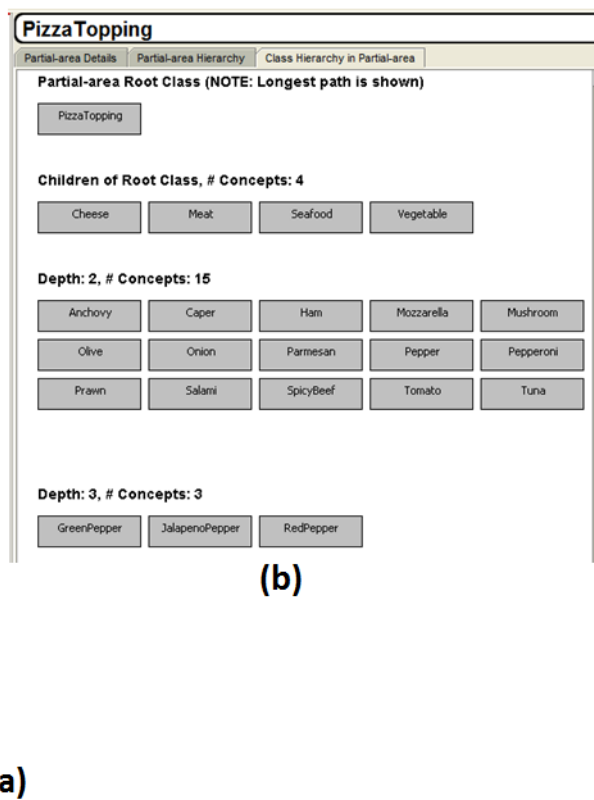
**Properties (1)**

Property Name	Property URI	Property Type	Property Usage	Property Inheritance
isToppingOf	PizzaOntology.owl	Object Property	Explicit Domain	Introduced

**Options**

**Classes (23)**

Class Name	Class URI
Anchovy	PizzaOntology.owl#Anchovy
Caper	PizzaOntology.owl#Caper
Cheese	PizzaOntology.owl#Cheese
GreenPepper	PizzaOntology.owl#GreenPepper
Ham	PizzaOntology.owl#Ham
JalapenoPepper	PizzaOntology.owl#JalapenoPepper
Meat	PizzaOntology.owl#Meat
Mozzarella	PizzaOntology.owl#Mozzarella
Mushroom	PizzaOntology.owl#Mushroom
Olive	PizzaOntology.owl#Olive
Onion	PizzaOntology.owl#Onion
Parmesan	PizzaOntology.owl#Parmesan
Pepper	PizzaOntology.owl#Pepper
Pepperoni	PizzaOntology.owl#Pepperoni



**PizzaTopping**

Partial-area Details | Partial-area Hierarchy | Class Hierarchy in Partial-area

**Partial-area Root Class (NOTE: Longest path is shown)**

PizzaTopping

**Children of Root Class, # Concepts: 4**

Cheese | Meat | Seafood | Vegetable

**Depth: 2, # Concepts: 15**

Anchovy | Caper | Ham | Mozzarella | Mushroom  
Olive | Onion | Parmesan | Pepper | Pepperoni  
Prawn | Salami | SpicyBeef | Tomato | Tuna

**Depth: 3, # Concepts: 3**

GreenPepper | JalapenoPepper | RedPepper

(b)

(a)

**Figure 2.11 (a)** The partial-area Details Panel with the PizzaTopping partial-area selected. **(b)** The “Class Hierarchy in Partial-area” tab for the PizzaTopping partial-area. As before, links are omitted but may be recovered for one class at a time by clicking on the class.

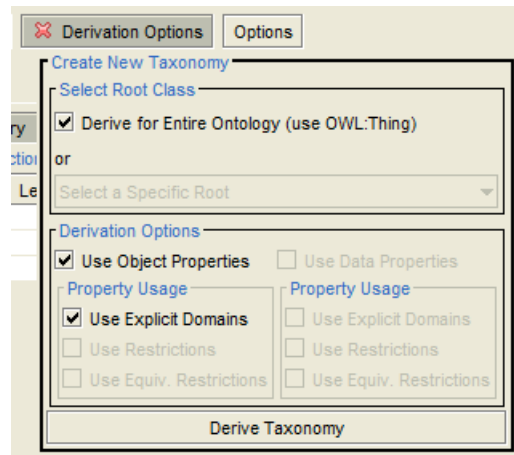
Figure 2.12 shows the “Partial-area Taxonomy Levels” tab, which is available when no partial-areas are selected within the OAF. In this tab, information is displayed about the levels of the Partial-area Taxonomy. For example, there are 2 areas at level 1, and 1 area at level 3. The total number of classes at each level is also displayed.

Summary   Partial-area Taxonomy Levels   Areas in Partial-area Taxonomy				
Abstraction Network Levels (3 total)				
Level	# Areas	# Partial-areas	# Classes	# Overlapping Class...
0	1	1	1	0
1	2	2	26	0
2	1	1	1	0

**Figure 2.12** The Partial-area Taxonomy Levels tab, displaying metrics for the Pizza Ontology after the current set of Exercises had been completed.

## Partial-area Taxonomy Derivation Options

Up to this point, we have focused on a Partial-area Taxonomy derived according to `rdfs:Domain` axioms of the object properties in the Pizza ontology. However, it is possible to derive other kinds of partial-area taxonomies, which are derived according to different knowledge elements in the ontology.



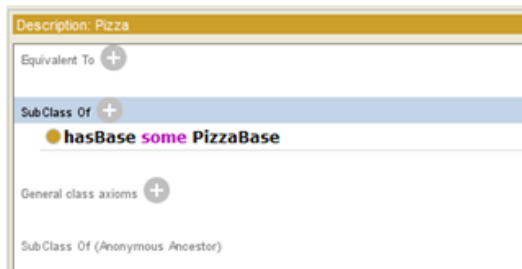
**Figure 2.13** The Partial-area Taxonomy derivation options panel in the OAF. By default, the explicitly defined domains of object properties (i.e., their `rdfs:Domain` axiom) are used for derivation.

Figure 2.13 illustrates the Derivation Options panel in the OAF. Since only object properties with domains have been defined within the Pizza ontology (so far), only this option is available. In Section 2.5 we will describe how restriction-defined Partial-area Taxonomies are created. Later we will look at partial-area taxonomies derived using data properties.

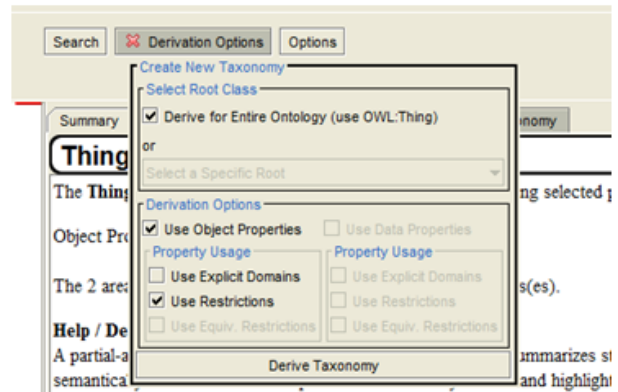
## 2.5 Object Properties in Restrictions

We now will illustrate how object properties used in class restrictions can be used to derive a Partial-area Taxonomy for the Pizza Ontology. Specifically, we focus on Sections 4.8.1 – 4.8.2 of the Pizza Ontology tutorial, “Property Restrictions.” We will now illustrate how each of the Exercises in this section of the POT affects a *restriction-defined Partial-area Taxonomy*.

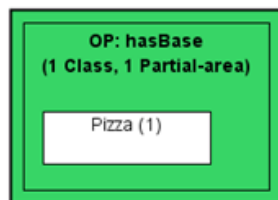
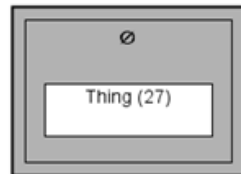
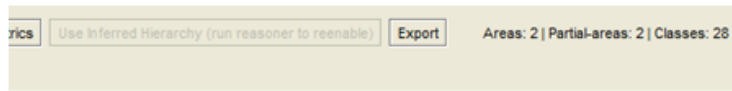
The first exercises in this section are **Exercises 16 and 17**, which instruct the reader to create a restriction on the Pizza class which states that a Pizza must have a PizzaBase. Figure 2.14 (a) shows this restriction in Protégé’s display of the definition of Pizza. Figure 2.14(b) shows that the “Use Object Properties in Restrictions” option becomes available after this restriction was added to the Pizza class. (Previously the option “Use Restrictions” appeared in gray and was not clickable. Now this option appears in black and it has become clickable.)



(a)



(b)



(c)

**Figure 2.14** (a) The “*hasBase some PizzaBase*” restriction added to the Pizza class. (b) The addition of a restriction does not affect the Partial-area Taxonomy derived using object property domains, which was illustrated throughout Section 2.4 of this tutorial. Since there now exists an object property used in a restriction the “Use Restrictions” option under “Use Object Properties” becomes available and the user selected it. (c) The restriction-defined Partial-area Taxonomy for the Pizza Ontology as of Exercise 17.

### Theory

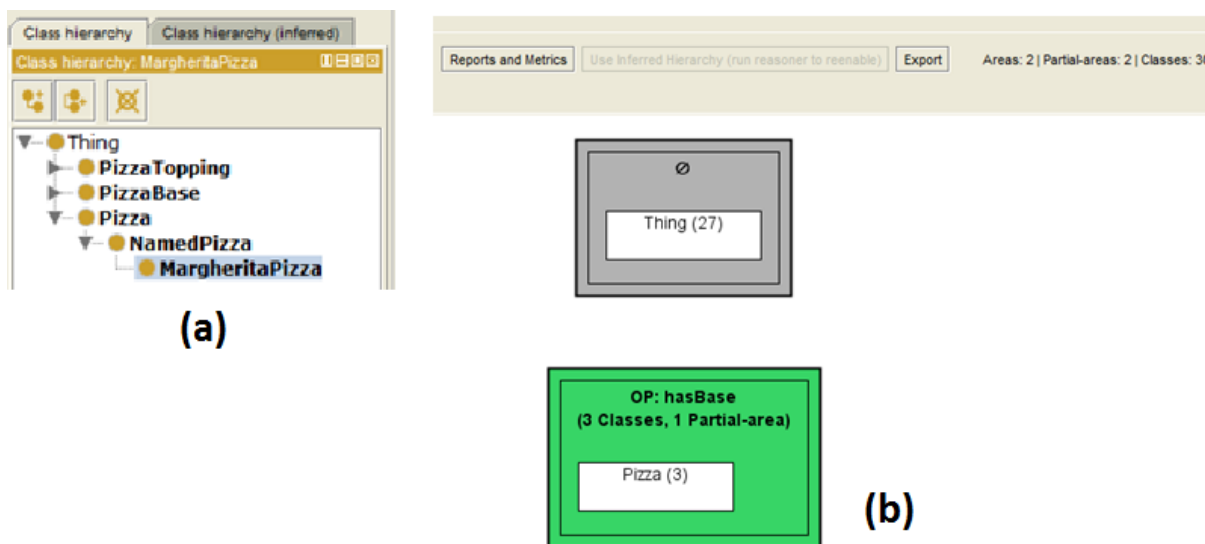
Different kinds of partial-area taxonomies summarize the content of the Pizza Ontology differently. A *restriction-defined Partial-area Taxonomy* summarizes the Pizza Ontology according to the object properties used as necessary conditions on classes within the Pizza Ontology. Areas in the *restriction-defined Partial-area Taxonomy* summarize sets of classes that have a specific set of object properties used in restrictions. Let us describe what is shown in Figure 2.14(c).

- The object property *hasBase* was used in a restriction on the class *Pizza*. In simple words, something can only be a pizza if it has a pizza base.<sup>1</sup> Thus, in the *restriction-defined Partial-area Taxonomy* an area  $\{hasBase\}$  exists. There is only one partial-area in this area, *Pizza* (1), because it is the only class in the ontology that has a restriction that uses the *hasBase* object property (and **only** the *hasBase* object property).
- The other classes in the *Pizza Ontology* are not under any restrictions, as of Exercise 17, thus they are summarized by the *Thing* (27) partial-area in the  $\emptyset$  area, indicating that these 27 classes have no restrictions.

## Software

Within the OAF, *restriction-defined Partial-area Taxonomies* become derivable only when an ontology contains an object property used in a restriction. Using the “Derivation Options” tab, a user can quickly switch back and forth between different summaries of the same ontology. The decision which summary is “the best” depends on the intricate details of the ontology, on the specific task of the user, and in some cases on the personal taste of the user.

In **Exercise 18** the reader creates two new descendant classes of *Pizza*: *NamedPizza*, a direct subclass, and *MargheritaPizza*, a subclass of *NamedPizza*. Figure 2.15(a) illustrates this subhierarchy of classes and Figure 2.15(b) shows the effect on the *restriction-defined Partial-area Taxonomy* for the *PizzaOntology*.



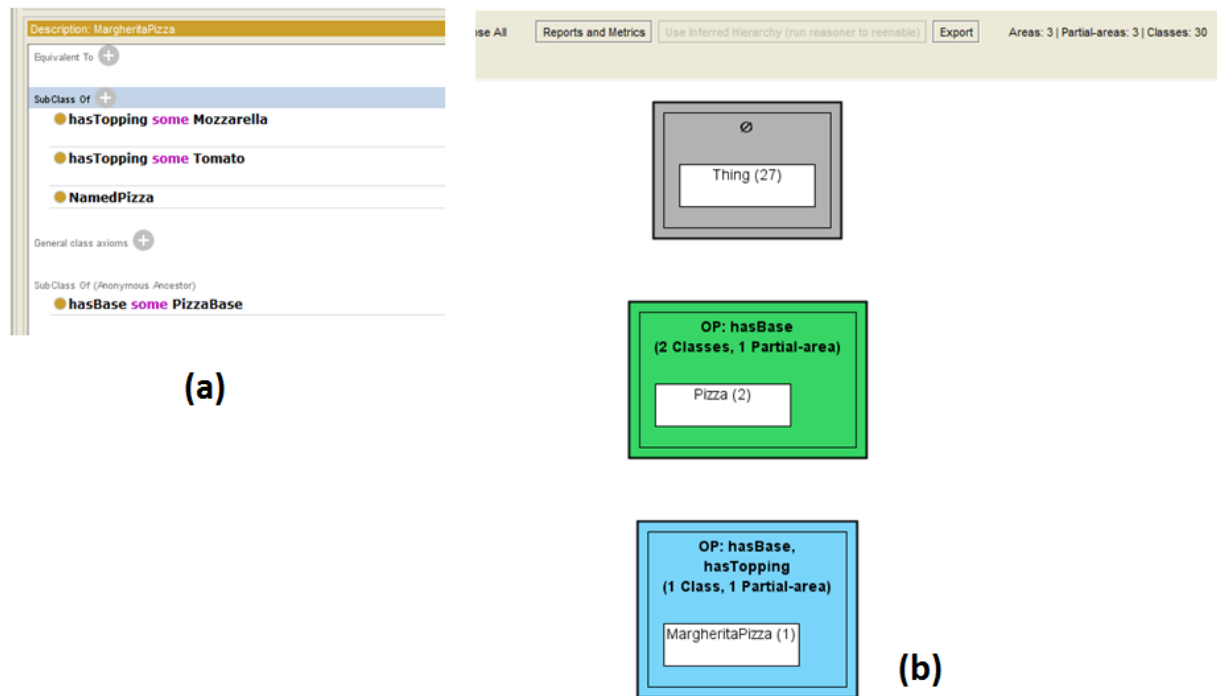
**Figure 2.15 (a)** The subhierarchy under *Pizza* after Exercise 18. **(b)** The restriction-defined Partial-area Taxonomy for the *Pizza Ontology* after two new classes were added that inherit the same restriction from *Pizza*.

<sup>1</sup> In American English the correct term would be “crust” but we will continue to use “base.”

## Theory

The two new classes, *NamedPizza* and *MargheritaPizza*, were added as child and descendant of the *Pizza* class, respectively. The class *Pizza* is under a restriction that uses the *hasBase* object property. Thus, the new classes *NamedPizza* and *MargheritaPizza* inherit this restriction. Therefore, they are now summarized by the *Pizza* (3) partial-area in the *{hasBase}* area, since they are descendants of *Pizza* and they implicitly have a restriction that includes *hasBase* (by inheritance). The range of a restriction **is not** considered when creating a *restriction-defined Partial-area Taxonomy*, just its use.

In **Exercises 19** and **20** the reader creates two existential restrictions using the *hasTopping* object property on the *MargheritaPizza* class. Figure 2.16(a) shows these restrictions in Protégé. Note that *MargheritaPizza* inherits the “*hasBase some PizzaBase*” restriction from *Pizza*.



**Figure 2.16 (a)** The existential restrictions defined for the *MargheritaPizza* class defined in Exercises 19 and 20. **(b)** The restriction-defined Partial-area Taxonomy after these restrictions have been added.

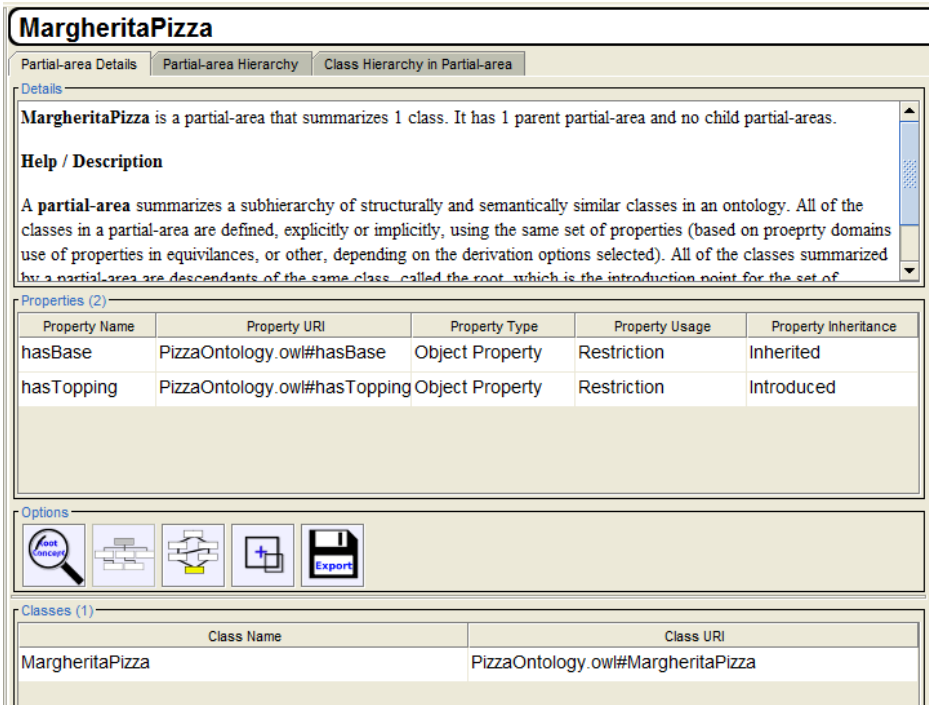
## Theory

This step illustrates the effects of inheritance in a *restriction-defined Partial-area Taxonomy*. The class *MargheritaPizza* introduces two restrictions into the ontology, “*hasTopping some Mozzarella*” and “*hasTopping some Tomato*.” Additionally, it inherits the “*hasBase some PizzaBase*” restriction from *Pizza*. Thus, the following major changes occurred:

- The class *MargheritaPizza* is now summarized by a partial-area named MargheritaPizza (1) in an area named {*hasBase*, *hasTopping*}. This indicates that the class *MargheritaPizza* is under restrictions that use the object properties *hasBase* and *hasTopping* (and no other object properties). Currently, only *MargheritaPizza* is under restrictions with both of these object properties; thus, the MargheritaPizza (1) partial-area summarizes only the one eponymous class.
- The *number* of restrictions that utilize a given object property do not affect the derivation of a *restriction-defined Partial-area Taxonomy*. Only the existence of an object property used in a restriction is considered.
- For example, the fact that there are two restrictions on *MargheritaPizza* that use *hasTopping* does not affect the derivation of the *restriction-defined Partial-area Taxonomy*. If *MargheritaPizza* had only one restriction that uses *hasTopping*, it would still be summarized by the {*hasBase*, *hasTopping*} area. This will be illustrated further in the next step.

## Software

Selecting a partial-area in the OAF provides details about the inheritance of restrictions within the Pizza Ontology. Figure 2.17 illustrates the inheritance of properties for the MargheritaPizza (1) partial-area.



**MargheritaPizza**

Partial-area Details | Partial-area Hierarchy | Class Hierarchy in Partial-area

**Details**

MargheritaPizza is a partial-area that summarizes 1 class. It has 1 parent partial-area and no child partial-areas.

**Help / Description**

A **partial-area** summarizes a subhierarchy of structurally and semantically similar classes in an ontology. All of the classes in a partial-area are defined, explicitly or implicitly, using the same set of properties (based on property domains use of properties in equivalences, or other, depending on the derivation options selected). All of the classes summarized by a partial-area are descendants of the same class, called the *root*, which is the introduction point for the set of

**Properties (2)**

Property Name	Property URI	Property Type	Property Usage	Property Inheritance
hasBase	PizzaOntology.owl#hasBase	Object Property	Restriction	Inherited
hasTopping	PizzaOntology.owl#hasTopping	Object Property	Restriction	Introduced

**Options**

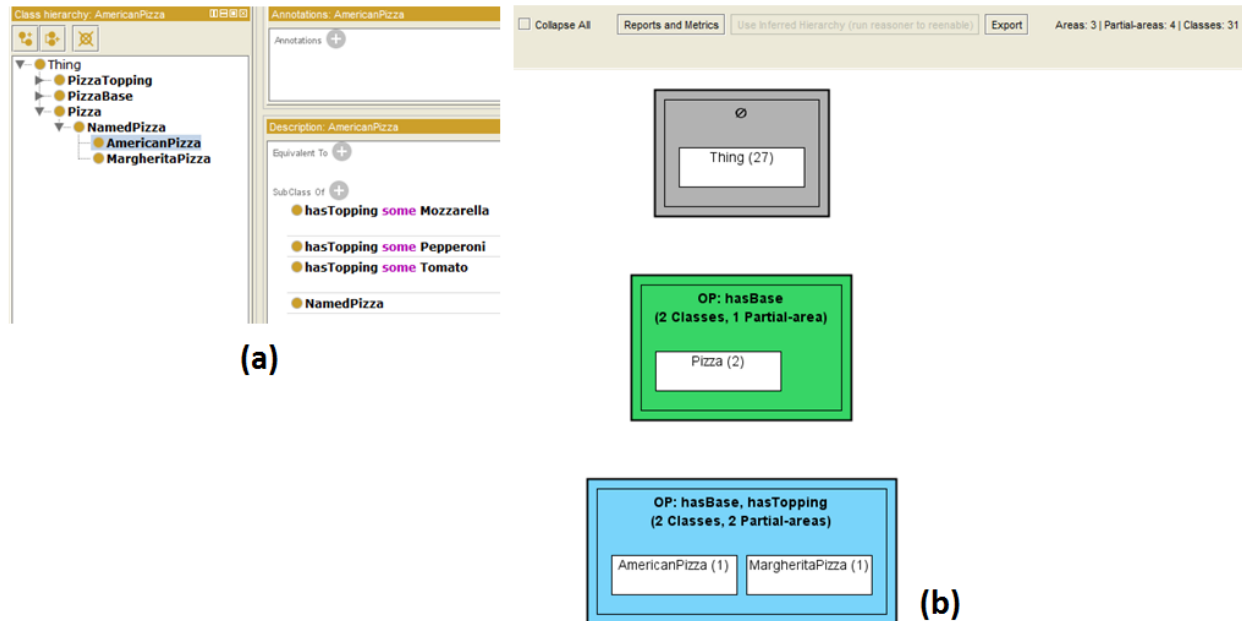
Find Concept | [Icon] | [Icon] | [Icon] | Export

**Classes (1)**

Class Name	Class URI
MargheritaPizza	PizzaOntology.owl#MargheritaPizza

**Figure 2.17** The partial-area Details Panel for the MargheritaPizza (1) partial-area. Note that the *hasBase* property used in a restriction is inherited while the *hasTopping* property used in a restriction is introduced at *MargheritaPizza*. Where a given property is inherited from can be seen in the “Partial-area Hierarchy” tab.

In **Exercise 21** the reader creates a new class *AmericanPizza* by cloning the *MargheritaPizza* class. Additional restrictions using *hasTopping* are also defined on the *AmericanPizza* class. This process is illustrated in Figure 2.18(a). The *restriction-defined Partial-area Taxonomy* after this class was added is shown in Figure 2.18(b).



**Figure 2.18 (a)** The class *AmericanPizza* and its restrictions, as displayed in Protégé. **(b)** The *restriction-defined Partial-area Taxonomy* after the addition of the *AmericanPizza* class. Note that the  $\{hasBase, hasTopping\}$  area now has two partial-areas.

## Theory

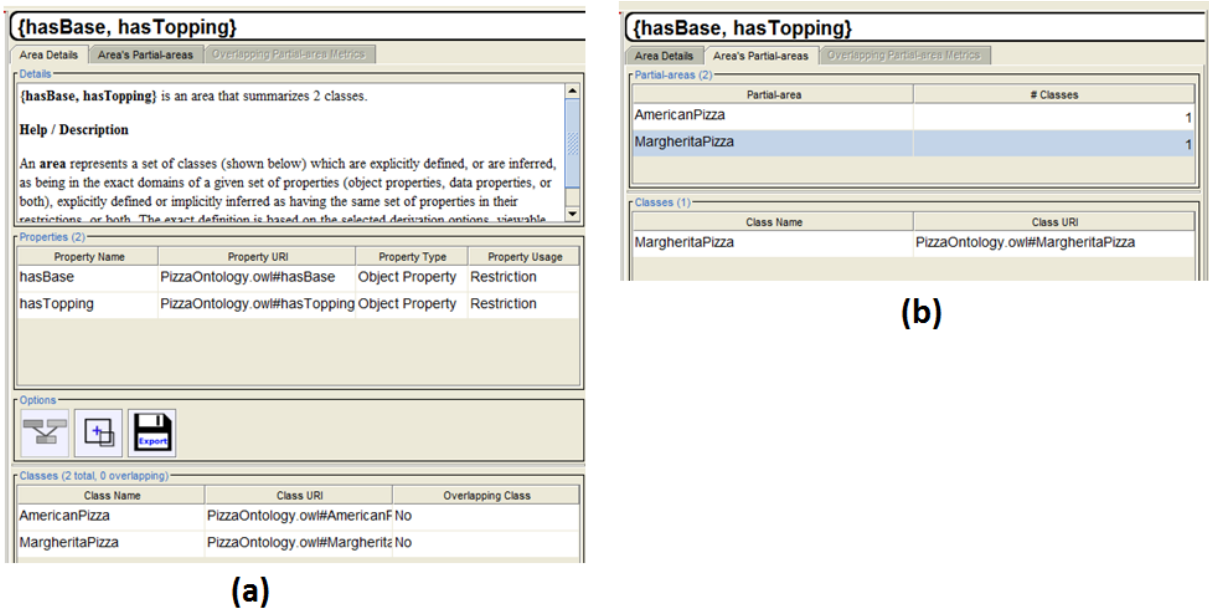
This step illustrates important new features of the Partial-area Taxonomy. Both classes *AmericanPizza* and *MargheritaPizza* do not have any parents in the area  $\{hasBase, hasTopping\}$ . Therefore, both of them are roots in the area  $\{hasBase, hasTopping\}$ , and therefore each of them initiates its own partial-area. This is the first example of an area with more than one partial-area.

- As described in the previous step, the number of restrictions with a given object property do not affect the derivation. The class *AmericanPizza*, which has three restrictions that use *hasTopping* and inherits the one restriction that uses *hasBase* from *Pizza*, is summarized by the  $\{hasBase, hasTopping\}$  area just as the class *MargheritaPizza*.
- The class *AmericanPizza* is a different introduction point for a restriction that uses *hasTopping*, since *AmericanPizza* is not hierarchically connected to *MargheritaPizza* (it is its sibling class). In other words, the two classes *AmericanPizza* and *MargheritaPizza* have the same restrictions, **but they received them in different ways**. This information is reflected by placing them into two separate partial-areas

in the  $\{hasBase, hasTopping\}$  area. The AmericanPizza (1) partial-area summarizes only one class, *AmericanPizza*, as of Exercise 21.

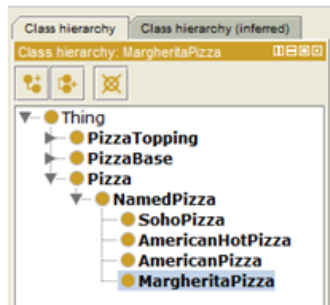
## Software

In this step we introduce the Area Details panel, which is displayed when an area is selected from the Partial-area Taxonomy. The Area Details panel displays information similar to the Partial-area Details panel, displayed, e.g., in Figure 2.17. Figure 2.19(a) illustrates this view for the  $\{hasBase, hasTopping\}$  area. Figure 2.18(b) illustrates how classes can be browsed by partial-area.

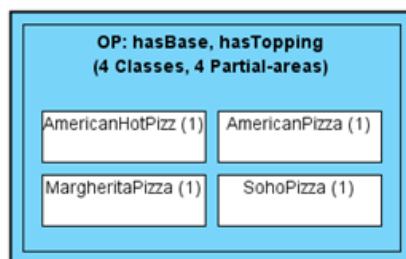
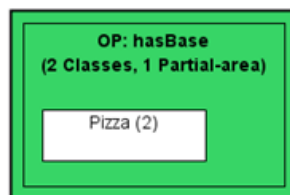
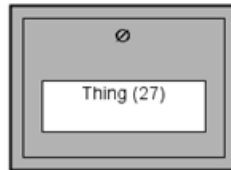
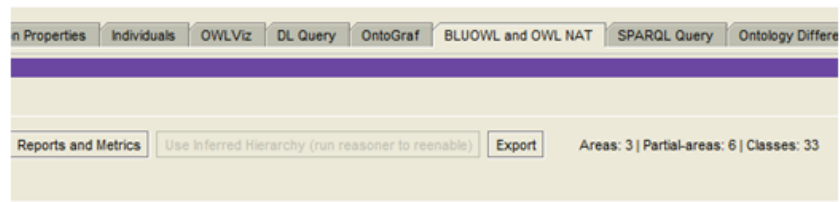


**Figure 2.19 (a)** The Area Details panel for the  $\{hasBase, hasTopping\}$  area. A summary of the area is shown at the top under **Details**, followed by the set of properties used in restrictions on the classes summarized by the area under **Properties**. Double clicking on a property takes the user to the definition/editing tab for the property in Protégé. Finally, a list of classes in the area is shown in alphabetical order under **Classes**. This list includes all classes from all partial-areas in the area. Double clicking on a class takes the user to its definition in Protégé. Additionally, this list indicates whether a given class is an *overlapping class*. Overlapping classes will be described in more detail in the next section. **(b)** The Area's "Partial-areas" tab, which enables a user to browse the classes summarized in the area by partial-area.

In **Exercise 22** additional types of pizzas, *AmericanHotPizza* and *SohoPizza*, are added as subclasses of *NamedPizza*. This exercise is illustrated in Figure 2.20. In **Exercise 23** the reader of the POT makes the different kinds of pizzas (*MargheritaPizza*, *AmericanPizza*, *AmericanHotPizza*, and *SohoPizza*) disjoint from one another. This has no effect on the Partial-area Taxonomy.



(a)



(b)

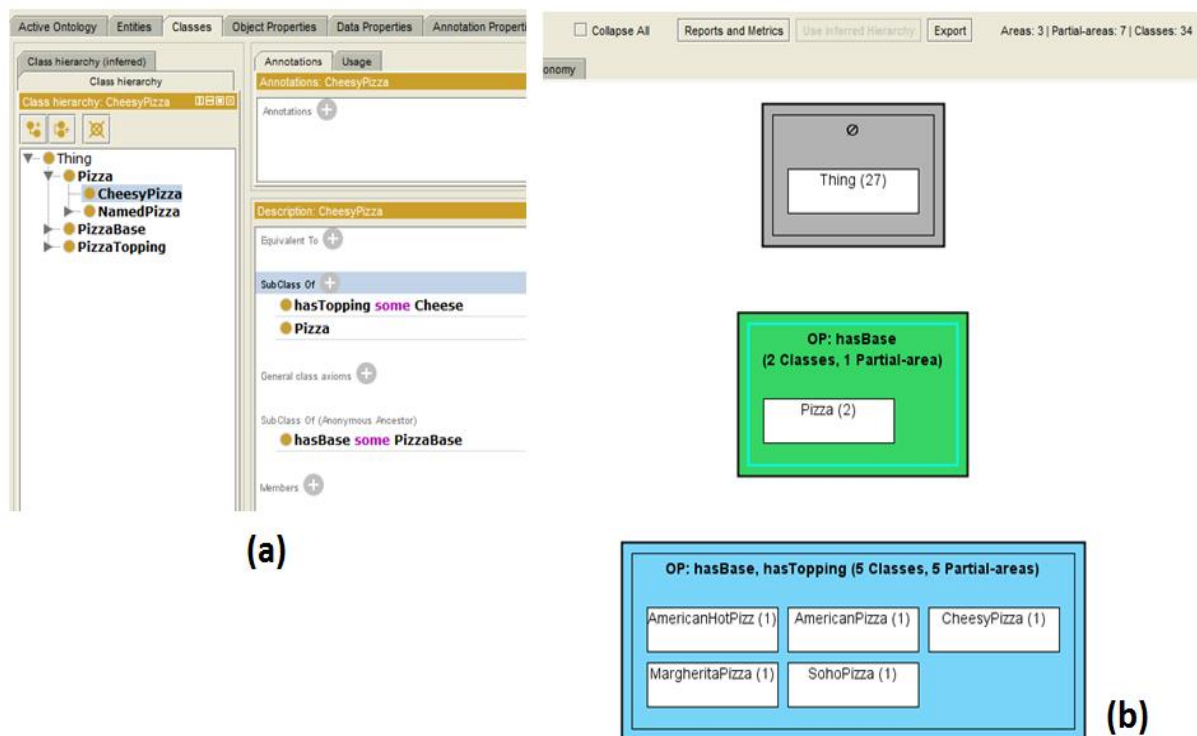
**Figure 2.20 (a)** The addition of two additional types of pizza, which both have restrictions that use the *hasTopping* object property. **(b)** Since these new types of pizza are different introduction points within the hierarchy for restrictions that use *hasTopping*, as was the case for *AmericanPizza* in the previous step, they are summarized by different partial-areas in the  $\{hasBase, hasTopping\}$  area.

The apparent levels inside of an area are NOT meaningful. They are simply the result of limited screen real estate. Ideally, all partial-areas should appear in one single row within their area. In this way, partial-area layout is different from area layout where levels ARE meaningful.

## 2.6 Object Properties in Class Equivalence Axioms

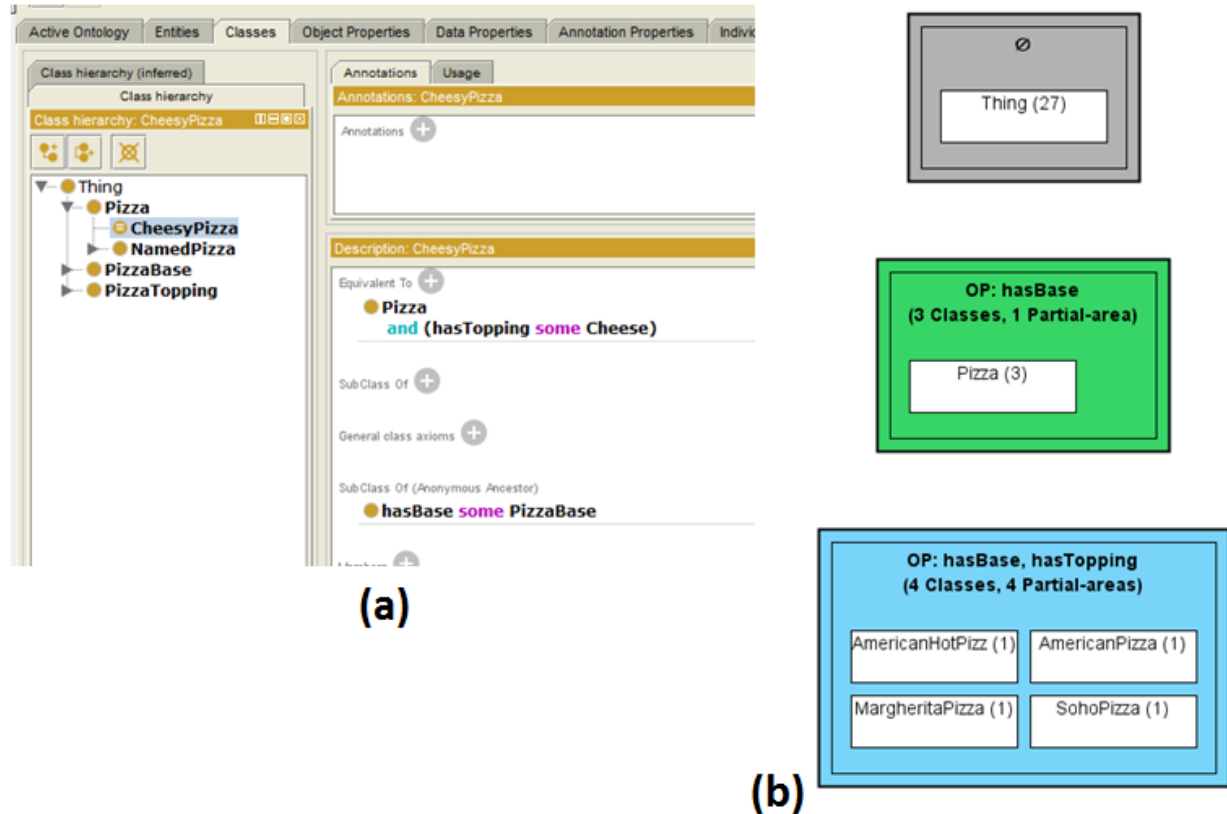
We now move to Section 4.9 of the POT, which covers the use of reasoners. Section 4.9.1 of the POT describes how reasoners can be invoked on an ontology and Section 4.9.2 explains how a reasoner can be used to detect inconsistent classes. These steps do not affect partial-area taxonomies, thus, we skip to Section 4.10, which describes necessary and sufficient conditions within an ontology. The most important part of this section of the Pizza Ontology Tutorial is the definition of equivalence axioms on classes.

We continue at **Exercise 28**, where a subclass of *Pizza* named *CheesyPizza* is added to the ontology. This class has one restriction, “*hasTopping* some *Cheese*.” This exercise is illustrated in Figure 2.21.



**Figure 2.20 (a)** The *CheesyPizza* class and its restriction, as displayed in Protégé. **(b)** The *restriction-defined Partial-area Taxonomy*, after the *CheesyPizza* class was added. Like the various types of pizza introduced in Section 2.5 of this tutorial, this class is summarized by {*hasBase*, *hasTopping*}. Note, however, that *CheesyPizza* is not a subclass of *NamedPizza*. *NamedPizza* is summarized in *Pizza* (2).

In **Exercise 29** the *CheesyPizza* class is turned into a defined class by giving it a necessary and sufficient condition (i.e., an equivalence axiom). This process is illustrated in Figure 2.21(a).



**Figure 2.21 (a)** *CheesyPizza* has been turned into a defined class by assigning it an equivalent class (i.e., a necessary and sufficient condition) “*Pizza* and (*hasTopping* some *Cheese*)”. **(b)** *The restriction-defined Partial-area Taxonomy*. Note that *CheesyPizza* is no longer a partial-area in {*hasBase*, *hasTopping*} and the *Pizza* (3) partial-area in {*hasBase*} now summarizes three classes.

## Theory

We make a distinction between object properties used in restrictions on classes and object properties used in restrictions in equivalence axioms. That means we distinguish between *necessary conditions* and *necessary and sufficient conditions*.

- *CheesyPizza* does not have a restriction that uses the *hasTopping* object property in a sufficient condition, as it did in the previous step. Thus, it is summarized by the *Pizza* (3) partial-area in the current *restriction-defined Partial-area Taxonomy*.
- To capture the restriction used in the equivalent class axiom on *CheesyPizza* one must explicitly include restrictions used in equivalent class axioms.
- In Figure 2.22 we illustrate two additional options for creating Partial-area Taxonomies. First, we explicitly include object properties used in restrictions in equivalent class axioms. Second, we show a Partial-area Taxonomy derived using only object properties used in restrictions in equivalent class axioms.

Create New Taxonomy

Select Root Class

☒ Derive for Entire Ontology (use OWL:Thing)

or

Select a Specific Root

Derivation Options

☒ Use Object Properties ☐ Use Data Properties

Property Usage

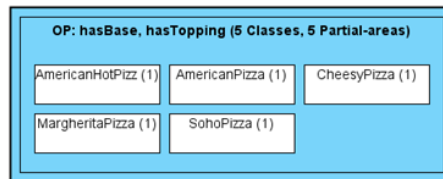
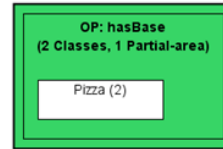
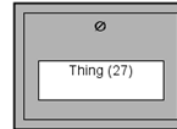
☐ Use Explicit Domains ☐ Use Explicit Domains

☒ Use Restrictions ☐ Use Restrictions

☒ Use Equiv. Restrictions ☐ Use Equiv. Restrictions

Derive Taxonomy

(a)



(b)

Create New Taxonomy

Select Root Class

☒ Derive for Entire Ontology (use OWL:Thing)

or

Select a Specific Root

Derivation Options

☒ Use Object Properties ☐ Use Data Properties

Property Usage

☐ Use Explicit Domains ☐ Use Explicit Domains

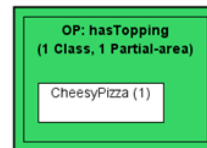
☐ Use Restrictions ☐ Use Restrictions

☒ Use Equiv. Restrictions ☐ Use Equiv. Restrictions

Derive Taxonomy

(c)

Use Inferred Hierarchy (run reasoner to reenable) Export Areas: 2 | Partial-areas: 2 | Classes: 34



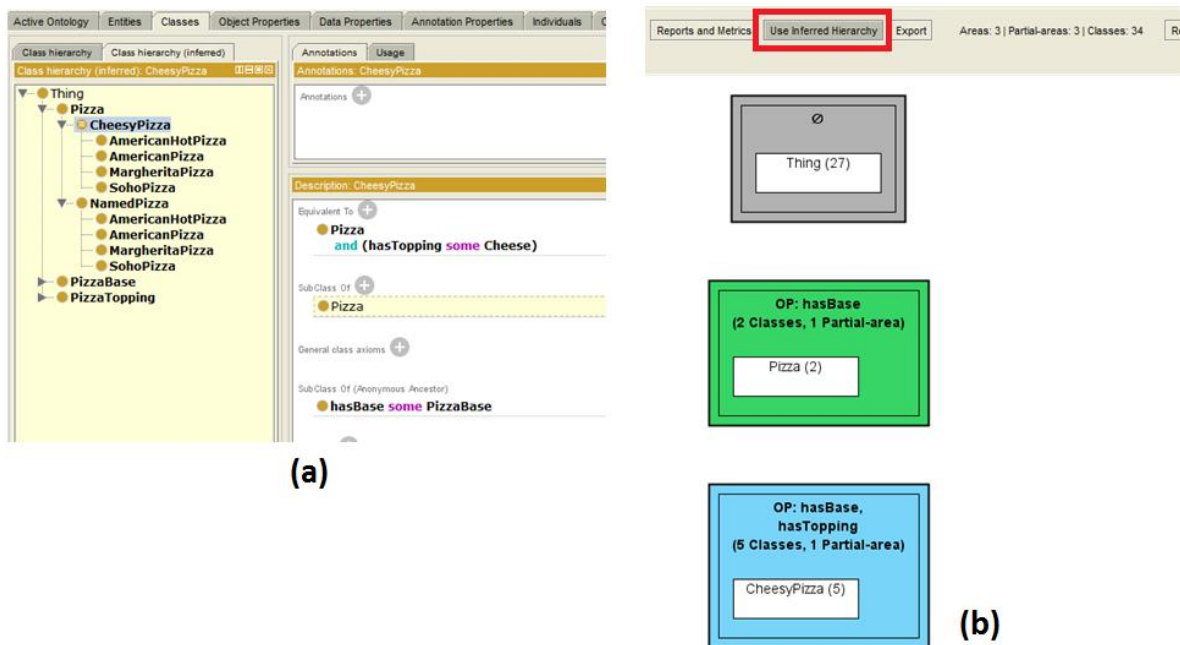
(d)

**Figure 2.22** (a) The Derivation Options panel with boxes checked for both object properties regular restrictions (described in Section 2.5 of this tutorial) and object properties used in restrictions in equivalent class axioms. (b) The *restriction-defined Partial-area Taxonomy* created using both of these options. Note that, since the *hasTopping* restriction used in *CheesyPizza*'s equivalent class axiom is now included, the *CheesyPizza (1)* partial-area once again appears in the  $\{hasBase, hasTopping\}$  area. (c) The Derivation Options panel with *only* object properties used in restrictions in equivalent class axioms selected. (d) The *restriction-defined Partial-area Taxonomy* created from **only** object properties used in restrictions in equivalent class axioms. Since the only class in the Pizza Ontology that has an equivalent class axiom up to this point is *CheesyPizza*, and the object property used in the equivalent class axiom's restriction is *hasTopping*, *CheesyPizza* is summarized by a partial-area named *CheesyPizza (1)* in the  $\{hasTopping\}$  area, capturing the fact that the *hasTopping* object property is used in an equivalent class axiom on this class. Since no other classes have an equivalent class axiom they are summarized by the *Thing (33)* partial-area in the  $\emptyset$  area. This demonstrates again that the user of OAF can view different kinds of Partial-area Taxonomies and choose the one that best meets his needs.

## 2.7 Inferred Partial-area Taxonomies

Up to this point we have explained the effects of editing operations on partial-area taxonomies derived for the **stated version** of the Pizza Ontology. However, to uncover all inferable axioms in an ontology it is necessary to apply a reasoner to the ontology. We thus move to Section 4.11 of the POT, which covers automated classification in an ontology.

In **Exercise 30** the ontology is processed by a reasoner. After an ontology has been processed this way it is possible to derive Partial-area Taxonomies for the inferred version of the ontology. In this section, we focus on a *restriction-defined Partial-area Taxonomy* that utilizes restrictions and restrictions in equivalent class axioms (i.e., the options shown in Figure 2.22(a)).



**Figure 2.23 (a)** The class hierarchy of the Pizza Ontology after a reasoner has been applied. Note that the named pizzas were inferred as subclasses of *CheesyPizza*, since they all had a restriction that met the necessary and sufficient conditions of *CheesyPizza*. **(b)** After a reasoner is applied to an ontology the “Use Inferred Hierarchy” option becomes available. This option has been selected and the *restriction-defined Partial-area Taxonomy* for the inferred version of the Pizza Ontology is shown.

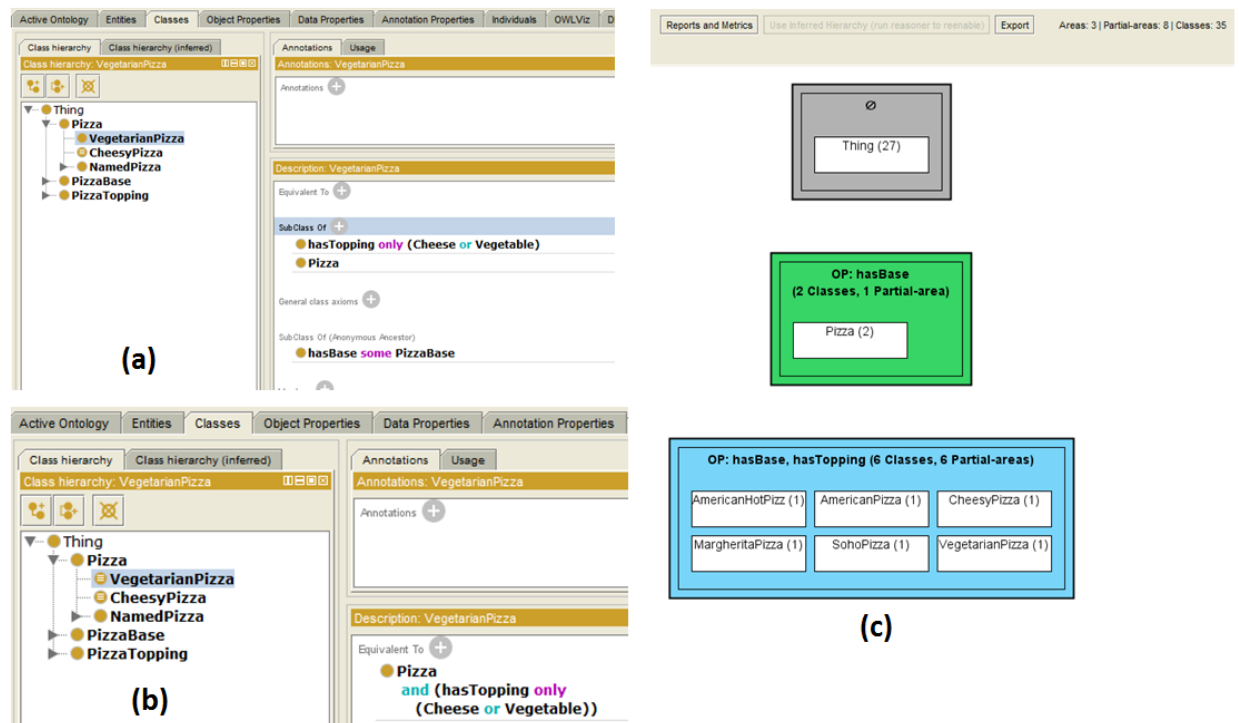
### Theory

An important point is illustrated by the inferred version of the Partial-area Taxonomy. Since *MargheritaPizza*, *AmericanPizza*, *AmericanHotPizza*, and *SohoPizza* have been inferred to be subclasses of *CheesyPizza*, which has a restriction that includes *hasTopping*, *CheesyPizza* is now the introduction point for *hasTopping* in a restriction.

Thus, the four named pizza types are now summarized by the CheesyPizza (5) partial-area in the Partial-area Taxonomy for the inferred version of the Pizza Ontology.

Section 4.12 of the POT described universal restrictions. The current definition of *restriction-defined Partial-area Taxonomy* does not distinguish between existential restrictions (which we have illustrated up to this point) and universal restrictions.

In **Exercise 31** the reader creates a class *VegetarianPizza*, a subclass of *Pizza*, with a universal restriction. In **Exercise 32** this universal restriction is used to turn *VegetarianPizza* into a defined class. Figure 2.24 illustrates the effects of adding this class on the *restriction-defined Partial-area Taxonomy* for the stated version of the Pizza Ontology.



**Figure 2.24** (a) The *VegetarianPizza* class and its universal restriction. (b) The defined version of the *VegetarianPizza* class. (c) The *restriction-defined Partial-area Taxonomy* for the stated version of the Pizza Ontology.

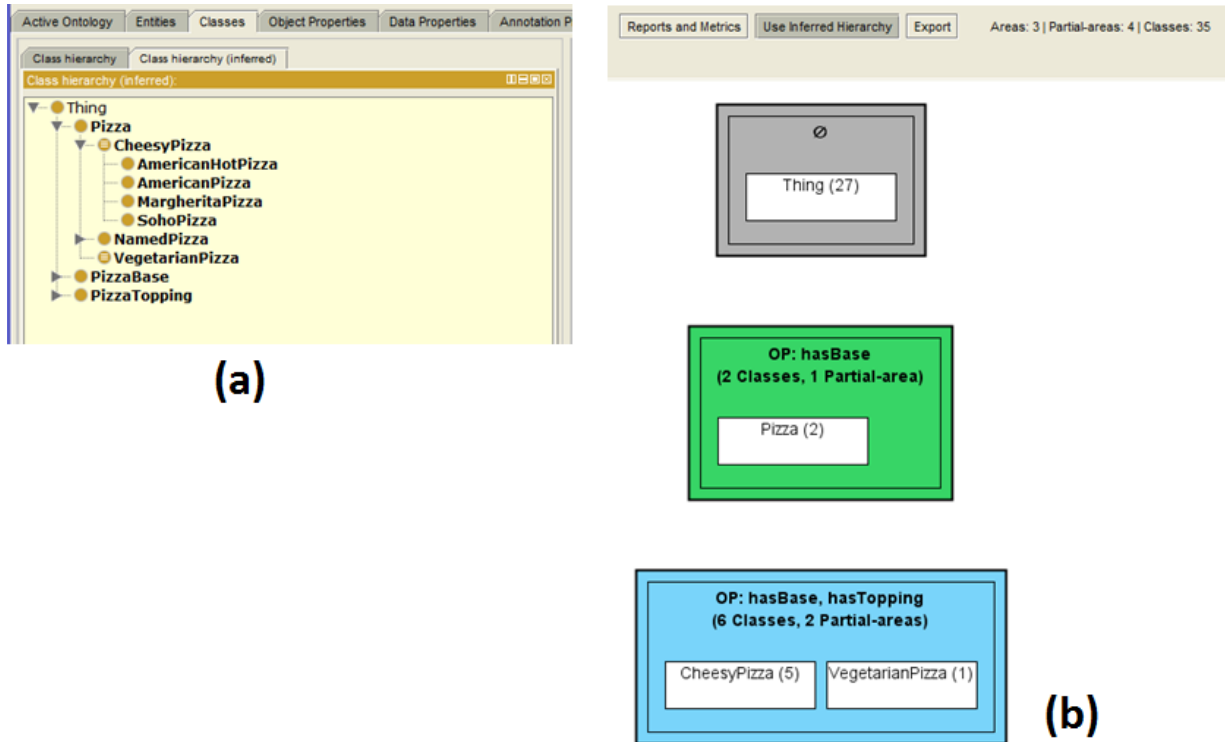
## Theory

We note two significant ideas:

- The Partial-area Taxonomy derivation does not distinguish between universal restrictions (“only” in a restriction) and existential restrictions (“some” in a restriction). The *VegetarianPizza* class, which has a universal restriction, is summarized with the classes that have an existential restriction. In a future version of our derivation software there will be an option to distinguish between the two restrictions.

- As before, the range is not considered when creating the Partial-area Taxonomy. Even though the *hasTopping* restriction has a range with a disjunction of classes (“*Cheese* or *Vegetable*”) this does not affect the derivation of the restriction-defined Partial-area Taxonomy.

After applying a reasoner (**Exercise 33**) the class hierarchy appears as shown in Figure 2.25(a).



**Figure 2.25 (a)** The inferred hierarchy of classes in the Pizza Ontology. **(b)** The restriction-defined Partial-area Taxonomy after applying the reasoner.

The next section of the POT, Section 4.13, covers Automated Classification and Open World Reasoning. **Exercises 34 – 37** create *closure axioms* for the various types of pizza defined in the ontology (see Figure 2.26). This enables the correct classification of the pizzas. This has no effect on the *restriction-defined Partial-area Taxonomy* derived for the stated version of the Pizza Ontology but will have a significant effect in the Partial-area Taxonomy for the inferred version of the ontology.

Description: MargheritaPizza

Equivalent To +

SubClass Of +

- hasTopping **only** (Mozzarella or Tomato)
- hasTopping **some** Mozzarella
- hasTopping **some** Tomato
- NamedPizza

(a)

Description: SohoPizza

Equivalent To +

SubClass Of +

- hasTopping **only** (Mozzarella or Tomato or Parmesan or Olive)
- hasTopping **some** Mozzarella
- hasTopping **some** Olive
- hasTopping **some** Parmesan
- hasTopping **some** Tomato
- NamedPizza

(b)

Description: AmericanPizza

Equivalent To +

SubClass Of +

- hasTopping **only** (Mozzarella or Pepperoni or Tomato)
- hasTopping **some** Mozzarella
- hasTopping **some** Pepperoni
- hasTopping **some** Tomato
- NamedPizza

(c)

Description: AmericanHotPizza

Equivalent To +

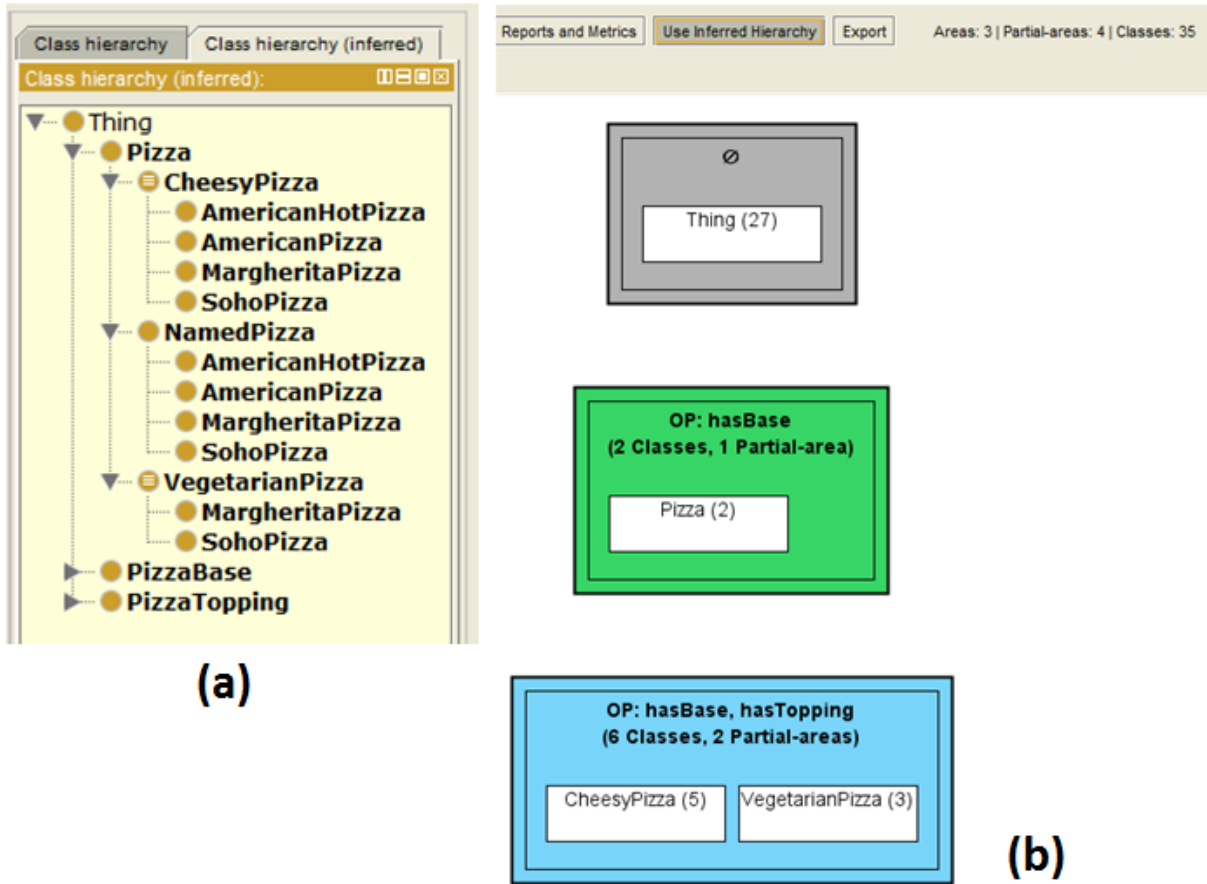
SubClass Of +

- hasTopping **only** (JalapenoPepper or Mozzarella or Pepperoni or Tomato)
- hasTopping **some** JalapenoPepper
- hasTopping **some** Mozzarella
- hasTopping **some** Pepperoni
- hasTopping **some** Tomato
- NamedPizza

(d)

**Figure 2.26** (a) The closure axiom for *MargheritaPizza*. (b) The closure axiom for *SohoPizza*. (c) The closure axiom for *AmericanPizza*. (d) The closure axiom for *AmericanHotPizza*.

In **Exercise 38** the reasoner is applied to the ontology after the closure axioms were created. Figure 2.27 (a) shows the inferred hierarchy and Figure 2.27(b) shows the restriction-defined Partial-area Taxonomy.



**Figure 2.27 (a)** The inferred class hierarchy, as shown in Protégé. Note that *MargheritaPizza* and *SohoPizza* were inferred as subclasses of *VegetarianPizza*. **(b)** The *restriction-defined Partial-area Taxonomy* for the inferred version of the Pizza Ontology. Note that the *VegetarianPizza* (3) partial-area now summarizes three classes.

### Theory

Note that partial-areas may be **non-disjoint**. In other words, one class may belong to two<sup>2</sup> partial-areas. This happens exactly when there exist two<sup>2</sup> paths from the class to two<sup>2</sup> different roots of the same area. Then this class will belong to the two<sup>2</sup> partial-areas defined by the two<sup>2</sup> roots.

This possibility of non-disjoint partial-areas differs markedly from areas. All areas are always disjoint from each other.

A class that belongs to two partial-areas is called an “overlapping class.” Two partial-areas that share one (or more) classe(s) are called overlapping partial-areas. A partial-area without any overlapping classes is therefore, naturally, called a non-overlapping partial-area and also called a disjoint partial-area.<sup>3</sup>

<sup>2</sup> “Two” in this sentence should be read as “two or more.”

<sup>3</sup> A disjoint partial-area is “disjoint with every other partial-area.”

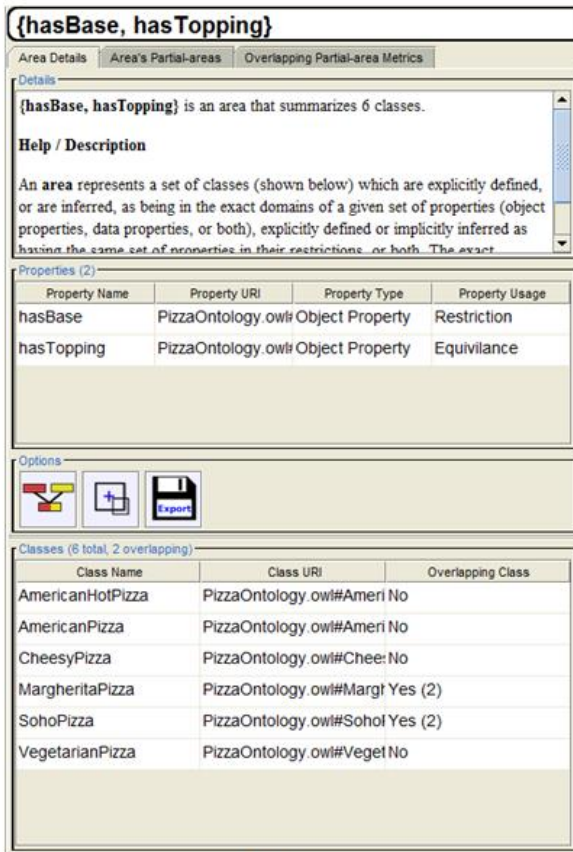
Due to multiple inheritance (i.e., a class may have multiple superclasses) within an ontology, a given class may be summarized by multiple partial-areas. For example, *MargheritaPizza* and *SohoPizza* are subclasses of *CheesyPizza* and of *VegetarianPizza*. Thus, *MargheritaPizza* and *SohoPizza* are summarized by their partial-areas. In Figure 2.27(b) one can see that the  $\{hasBase, hasTopping\}$  area summarizes **6 Classes**. However, the *CheesyPizza* (5) partial-area summarizes five classes and the *VegetarianPizza* (3) partial-area summarizes three classes. As  $5+3 = 8 > 6$ , two classes must be summarized by both partial-areas.

- An *overlapping partial-area* is a partial-area that summarizes a class that is summarized by another partial-area. *CheesyPizza* (5) and *VegetarianPizza* (3) are overlapping partial-areas.
- A class that is summarized by multiple partial-areas is called an *overlapping class*. *MargheritaPizza* and *SohoPizza* are overlapping classes in the *restriction-defined Partial-area Taxonomy* derived for the stated version of the Pizza Ontology.
- Specifically, *MargheritaPizza* and *SohoPizza* are overlapping classes because they are descendants of two different introduction points in the set of classes that has the object properties *hasBase* and *hasTopping* used in restrictions.
- The number of partial-areas that summarize a given class *C* is called the *degree of overlap of C*. *MargheritaPizza* and *SohoPizza* have a degree of overlap of two.

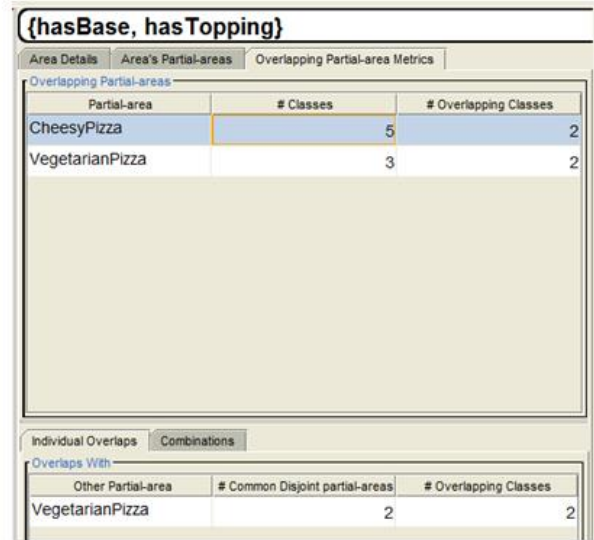
Non disjoint partial-areas may be considered undesirable and can be eliminated at the expense of introducing additional partial-areas. For this purpose we define the *disjoint Partial-area Taxonomy*. In a *disjoint Partial-area Taxonomy* each partial-area is disjoint with every other partial-area. This will be illustrated below.

## Software

Within the OAF tool, overlapping classes can be identified by clicking on a given area. For example, Figure 2.28(a) shows the area details dialog for the  $\{hasBase, hasTopping\}$  area in the *restriction-defined Partial-area Taxonomy* derived for the inferred version of the Pizza Ontology.



(a)

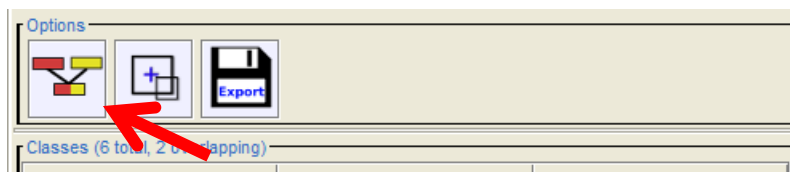


(b)

**Figure 2.28 (a)** The area details dialog for the  $\{hasBase, hasTopping\}$  area. In the class list at the bottom *MargheritaPizza* and *SohoPizza* are identified as overlapping classes. The number in parenthesis is the degree of overlap. **(b)** The “Overlapping Partial-area Metrics” tab is enabled when an area has overlapping classes. In this tab the user can see which partial-areas are overlapping partial-areas and where the overlapping classes are. For example, the two overlapping classes in the *CheesyPizza* partial-area are also summarized by the *VegetarianPizza* partial-area.

### Deriving Disjoint Partial-area Taxonomies

If an area contains an overlapping class, the OAF tool can derive a *disjoint Partial-area Taxonomy* for the area. A *disjoint Partial-area Taxonomy* will summarize each of the area’s classes by exactly one *disjoint partial-area*. Figure 2.29 illustrates how a disjoint partial-area can be derived.



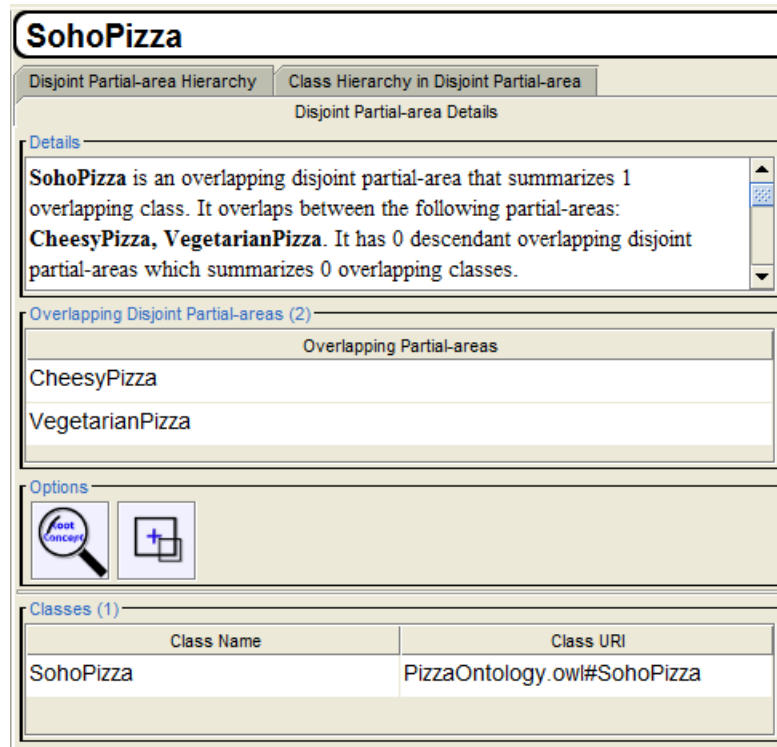
**Figure 2.29** The area Details Panel options display. The left-most button, enabled when the selected area has overlapping classes, derives a *disjoint Partial-area Taxonomy* for the selected area.

Clicking the left-most button in Figure 2.29 will derive a disjoint Partial-area Taxonomy for the selected area. The full description of the properties of a disjoint Partial-area Taxonomy is outside of the scope of this tutorial, but the idea is illustrated with an example in Figure 2.30.



**Figure 2.30** The disjoint Partial-area Taxonomy for the  $\{hasBase, hasTopping\}$  area. The non-overlapping partial-areas are shown using a single color border. For example, the CheesyPizza (3) non-overlapping partial-area summarizes three non-overlapping classes and has a blue border. There is only one non-overlapping class summarized by the VegetarianPizza (1) non-overlapping partial-area, which has a red border. The overlapping partial-areas are color-coded according to the partial-areas that created the overlap. There are two points of overlap between the CheesyPizza (3) and VegetarianPizza (1) partial-areas: *SohoPizza* and *MargheritaPizza*. Thus, these two classes appear as roots of two overlapping partial-areas and are marked by red/blue borders.

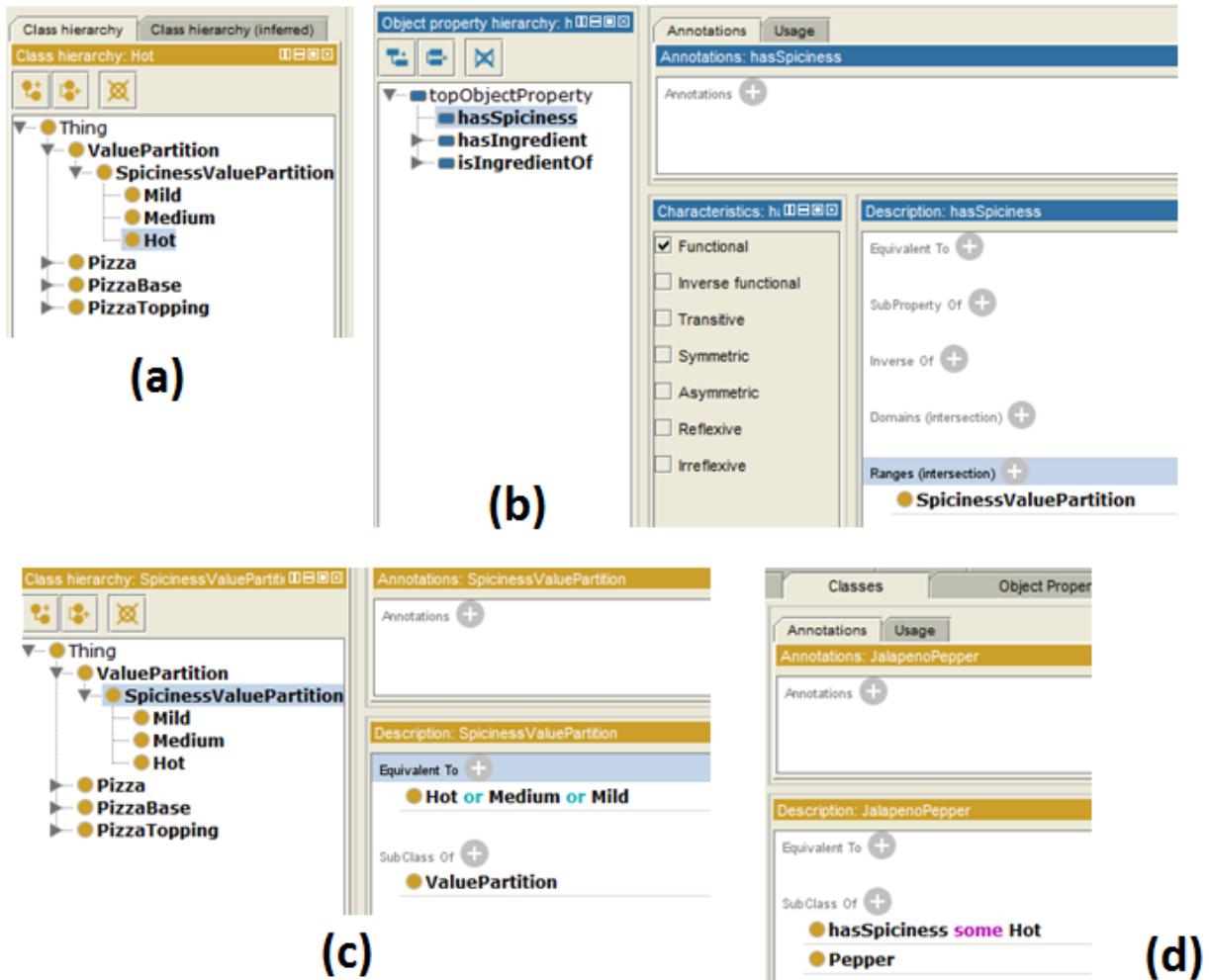
Clicking on a disjoint partial-area provides details about the class(es) summarized by the disjoint partial-area, along with what partial-areas caused the overlap. Figure 2.31 illustrates the information displayed within the Ontology Abstraction Framework tool when the SohoPizza (1) disjoint partial-area is selected by a mouse click.



**Figure 2.31** The disjoint partial-area Details Panel with the SohoPizza (1) disjoint partial-area selected.

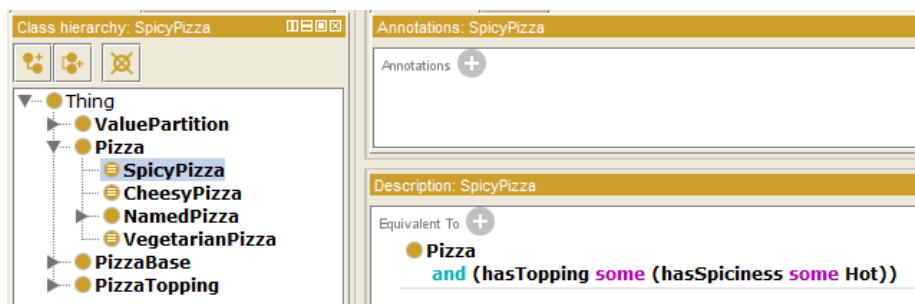
## 2.7 Value Partition Design Pattern

Section 4.14 of the POT illustrates the application of the Value Partition design pattern within the Pizza Ontology. In **Exercise 39** the reader creates a new subhierarchy of spiciness-related classes and a new object property *hasSpiciness*, as illustrated in Figure 2.32 (a) and Figure 2.32 (b). In **Exercise 40** restrictions representing spiciness are added to the leaf classes in the *PizzaTopping* subhierarchy. For example, *JalapenoPepper* has a spiciness of *Hot* (Figure 2.32(d)).

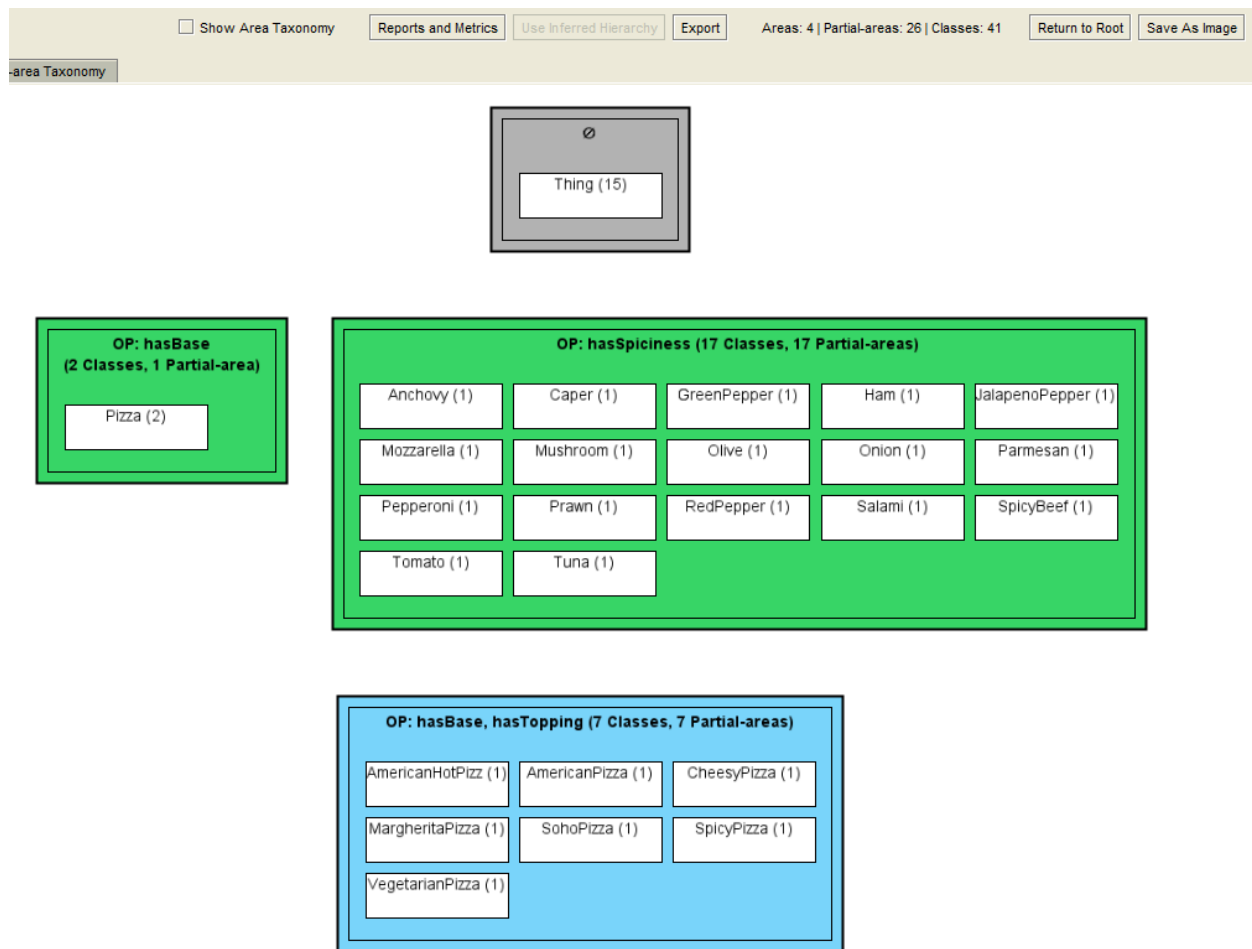


**Figure 2.32** (a) The *SpicinessValuePartition* subhierarchy in the Pizza Ontology. (b) The *hasSpiciness* object property. Note that no domain is assigned to this object property. (c) An equivalence added to the *SpicinessValuePartition* class. (d) An example of a spiciness restriction on the *JalapenoPepper* pizza topping class.

In **Exercise 41** a *SpicyPizza* class is introduced to the ontology, as illustrated in Figure 3.33. Figure 3.34 shows the restriction-defined Partial-area Taxonomy for the stated version of the ontology.



**Figure 3.33** The *SpicyPizza* class and its definition.



**Figure 3.34** The restriction-defined Partial-area Taxonomy derived for the stated version of the Pizza Ontology after value partition hierarchy was added and pizza toppings were assigned a spiciness level.

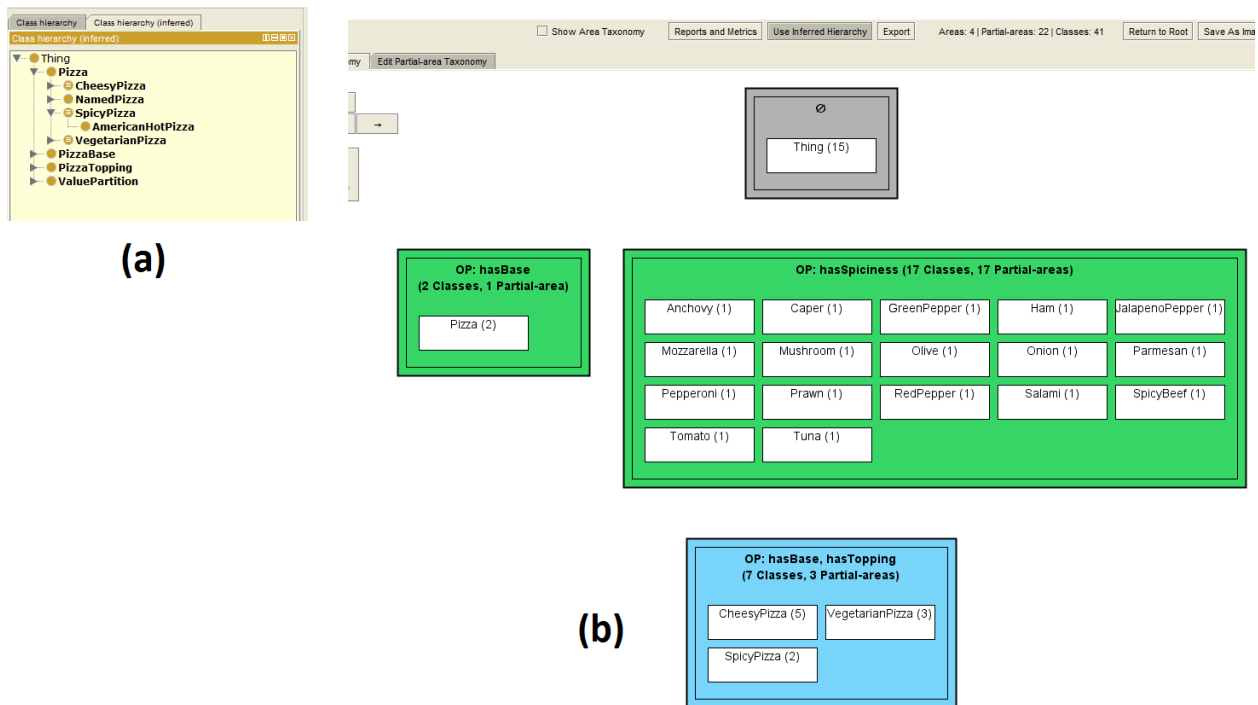
### Theory

Restrictions that use the *hasSpiciness* object property were added to the seventeen leaf classes in the *PizzaTopping* subhierarchy. Thus, a  $\{hasSpiciness\}$  area with seventeen partial-areas was introduced into the Partial-area Taxonomy. This area is on level 1, the level marked by the use of a green background.

The *SpicyPizza* class was defined as having an equivalence with a restriction “*hasTopping* some (*hasSpiciness* some *Hot*).” This indicates that any subclass of *SpicyPizza* must have a topping that has a spiciness of *Hot*. In a previous step we mentioned that the range of a restriction is not considered when deriving a *restriction-defined Partial-area Taxonomy*. The *hasSpiciness* property is used in a restriction that is the **range** of a restriction on the *SpicyPizza* class. Thus, it is not considered when deriving the partial-area taxonomy. A single partial-area *SpicyPizza* (1) is introduced to the  $\{hasBase, hasTopping\}$  area since the *hasTopping* property is used in the restriction on the class.

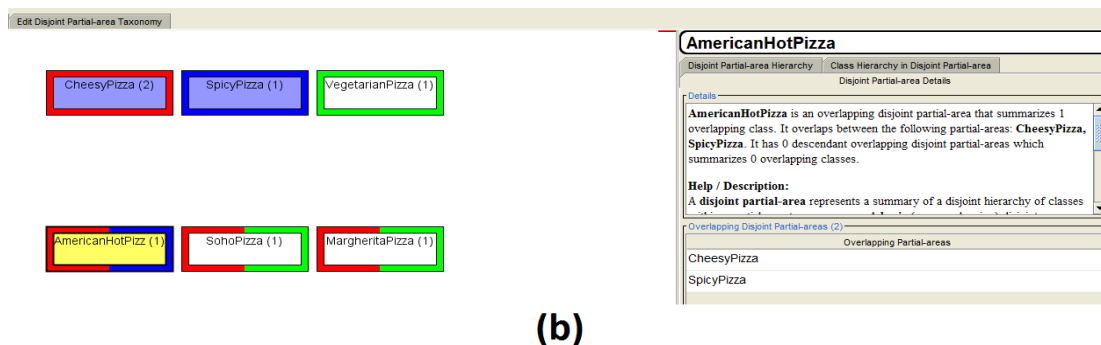
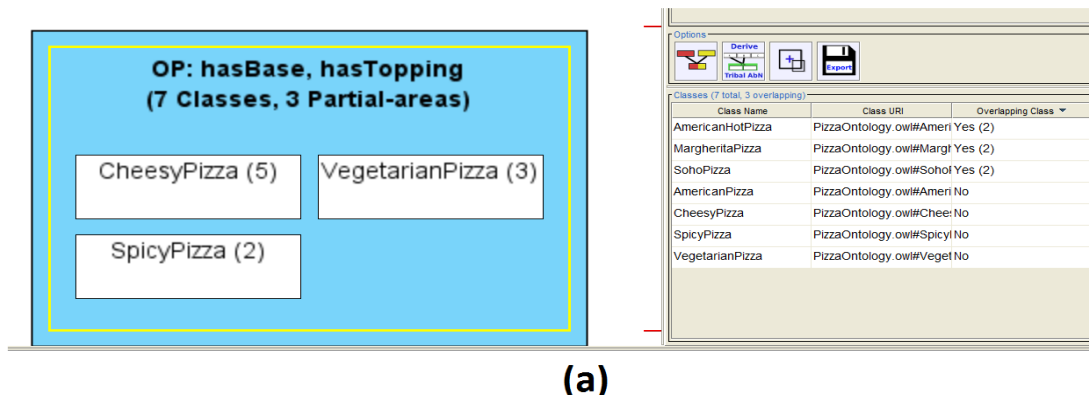
In a future version of the OAF there will be an option to include object properties used in the range of a restriction. This will provide a view of the ontology that identifies the various uses of object properties within the ontology.

After applying a reasoner to the ontology (**Exercise 42**), we see the inferred class hierarchy in Figure 2.35(a). *AmericanHotPizza* has been inferred as a subclass of *SpicyPizza* because an *AmericanHotPizza* has a *JalapenoPepper* topping and a *JalapenoPepper* topping has a spiciness of *Hot*. The restriction-defined Partial-area Taxonomy for the inferred version of the ontology is shown in Figure 3.35(b). Since *AmericanHotPizza* was inferred as a child of *SpicyPizza*, *AmericanHotPizza* is now summarized by the *SpicyPizza* (2) partial-area in the area  $\{hasBase, hasTopping\}$ .



**Figure 2.35 (a)** The inferred class hierarchy in the Pizza Ontology, as shown in Protégé, after adding the *hasSpiciness* value partition classes and restrictions. **(b)** The *restriction-defined Partial-area Taxonomy* derived from the inferred version of the Pizza Ontology.

By selecting the  $\{hasBase, hasTopping\}$  area, we observe that *AmericanHotPizza* is now an overlapping class (see Figure 2.36(a)). By deriving the disjoint partial-area taxonomy for  $\{hasBase, hasToppin\}$  we observe that it overlaps between the *CheesyPizza* and *SpicyPizza* partial-areas (Figure 2.36(b)).



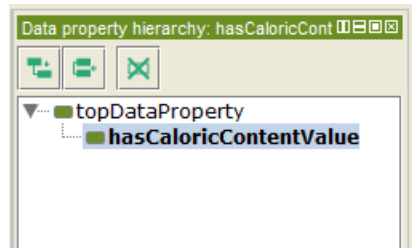
**Figure 2.36. (a)** The  $\{hasBase, hasTopping\}$  area in the inferred version of the ontology. Note that, on the right hand side, *AmericanHotPizza* is listed as an overlapping concept. **(b)** The disjoint partial-area taxonomy for the  $\{hasBase, hasTopping\}$  area in the inferred version of the ontology. Note the introduction of the *AmericanHotPizza* disjoint partial-area (selected in yellow).

Sections 4.16 and 4.17 of the POT describe cardinality restrictions on classes. These have no significant effect on the partial-area taxonomies. Thus we skip to Chapter 5 of the tutorial, “Datatype Properties.”

## 2.8 Data Properties

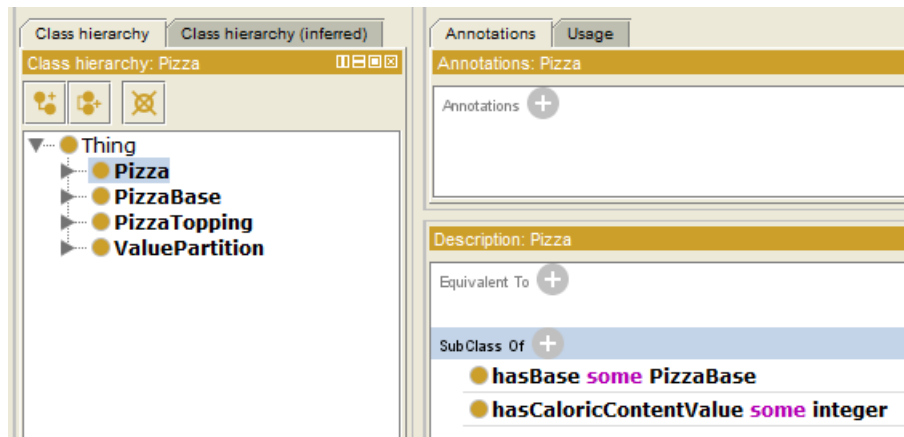
Up to this point, the derivation of partial-area taxonomies was based on the domains of object properties or the use of object properties in restrictions. However, Datatype Properties, or Data Properties, for short, function the same way as object properties except their ranges are literal values, like strings or integers, instead of classes in the ontology. Thus, it is possible to create partial-area taxonomies using data properties.

In **Exercise 46** the reader of the POT creates a data property *hasCaloricContentValue*, for representing the number of calories in a type of pizza. This is illustrated in Figure 2.37.



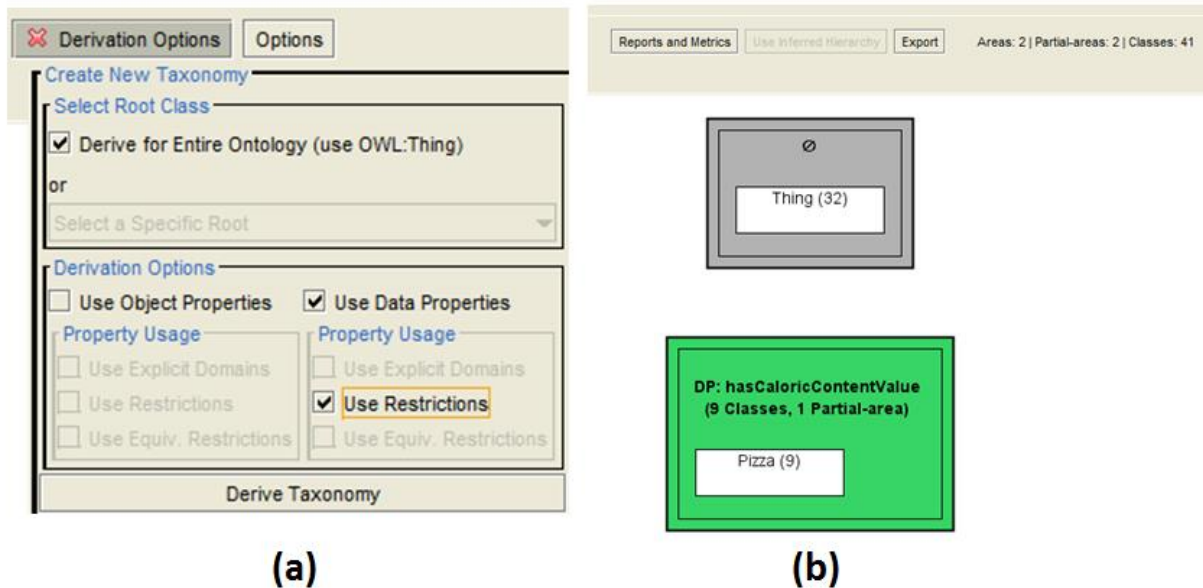
**Figure 2.37** The *hasCaloricContentValue* data property.

**Exercise 47** focuses on the creation of Pizza individuals/instances. This information is not reflected in partial-area taxonomies. Thus, we skip to **Exercise 48**, where the reader creates a restriction using the *hasCaloricContentValue* data property which states that all Pizzas have a caloric value, as illustrated in Figure 2.38.



**Figure 2.38** A restriction that states that *Pizza* must have a *hasCaloricContentValue* with some integer value as its range.

This change is not immediately reflected in the *restriction-defined Partial-area Taxonomy* since we have not yet selected data properties for use in derivation. In Figure 2.39 (a) the “Use Data Properties” option is selected. This option becomes available when a data property is added to the ontology and that data property is assigned a domain or is used in a restriction. Figure 2.39 (b) illustrates a Partial-area Taxonomy created using data properties used in restrictions.



**Figure 2.39 (a)** The Derivation Options panel. “Use Object Properties” has been unselected and “Use Data Properties” has been selected. Since there is a data property used in a restriction (*hasCaloricContentValue* is used in a restriction on *Pizza*) the “Use Restrictions” option is available and has been selected. **(b)** The restriction-defined Partial-area Taxonomy created using data properties. The Partial-area Taxonomy for the stated version of the Pizza Ontology is shown.

### Theory

The derivation of partial-area taxonomies using data properties follows that of partial-area taxonomies derived using object properties. In this example:

- There is only one data property defined in the Pizza ontology. It has no domain assigned, thus it is not possible to create a domain-defined Partial-area Taxonomy using data properties in the pizza ontology.
- There is one class that has the *hasCaloricContentValue* data property used in a restriction, *Pizza*. All of the descendants of *Pizza* (eight classes in total) inherit this restriction.
- Thus, in the *data-property-based Partial-area Taxonomy* shown in Figure 2.38(b) there are two areas. The area  $\emptyset$  summarizes the 32 classes that do not have any restriction using a data property. The area  $\{hasCaloricContentValue\}$  summarizes the nine classes that are explicitly or implicitly in the restriction defined on *Pizza*.

### Software

The functionality of the Ontology Abstraction Framework tool is the same when working with partial-area taxonomies defined using object properties or data properties. Figure 2.40 shows the partial-area Details Panel for the Pizza partial-area.

Pizza

Partial-area Details

Partial-area Hierarchy

Class Hierarchy in Partial-area

Details

Pizza is a partial-area that summarizes 9 classes. It has 1 parent partial-area and no child partial-areas.

**Help / Description**

A **partial-area** summarizes a subhierarchy of structurally and semantically similar classes in an ontology. All of the classes in a partial-area are defined, explicitly or implicitly, using the same set of properties (based on property...

Properties (1)

Property Name	Property URI	Property Type	Property Usage	Property Inherita...
hasCaloricCon	PizzaOntology.	Data Property	Restriction	Introduced

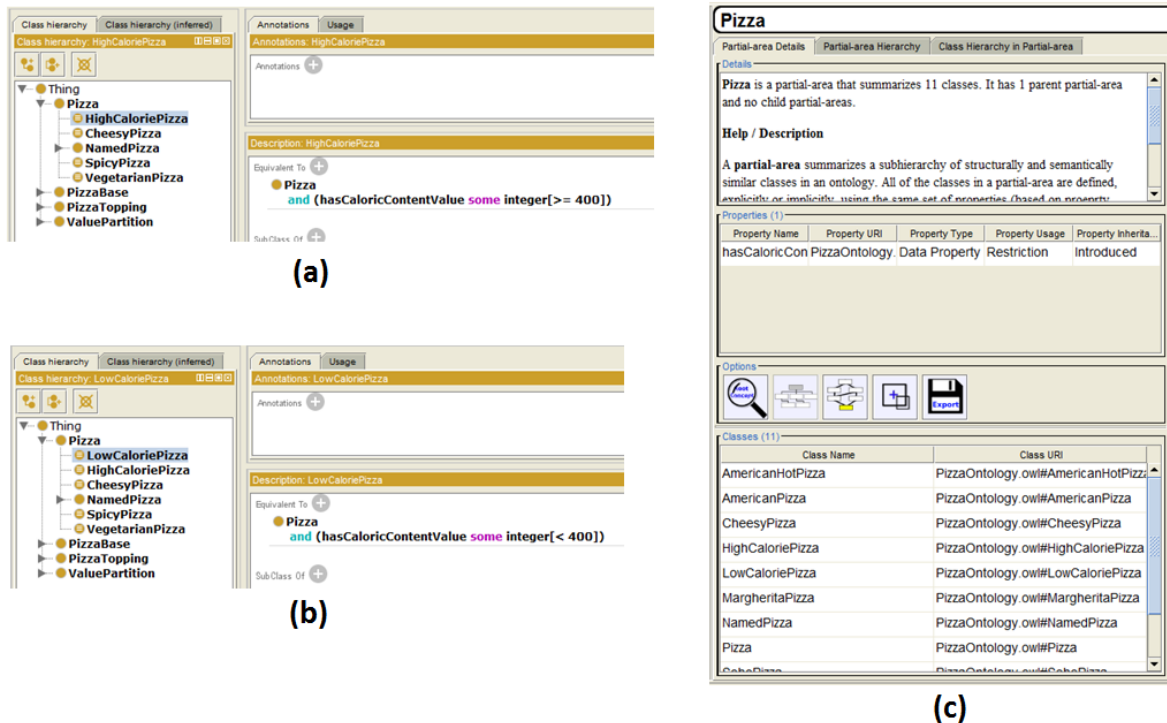
Options

Classes (9)

Class Name	Class URI
AmericanHotPizza	PizzaOntology.owl#AmericanHotPizza
AmericanPizza	PizzaOntology.owl#AmericanPizza
CheesyPizza	PizzaOntology.owl#CheesyPizza
MargheritaPizza	PizzaOntology.owl#MargheritaPizza
NamedPizza	PizzaOntology.owl#NamedPizza
Pizza	PizzaOntology.owl#Pizza
SohoPizza	PizzaOntology.owl#SohoPizza
SpicyPizza	PizzaOntology.owl#SpicyPizza
VegetarianPizza	PizzaOntology.owl#VegetarianPizza

**Figure 2.40** The partial-area Details Panel for the Pizza partial-area in the *data-property-based Partial-area Taxonomy*. Note that the “Property Type” under **Properties** has the value “Data Property.”

In **Exercise 49** the reader creates two classes, *HighCaloriePizza* and *LowCaloriePizza*, with more specific data property restrictions than the one on *Pizza* (see Figures 2.41 (a) and 2.41(b)).



**Figure 2.41** (a) The definition of *HighCaloriePizza*. (b) The definition of *LowCaloriePizza*. (c) The partial-area Details Panel for the *Pizza* partial-area after the *HighCaloriePizza* and *LowCaloriePizza* classes were added to the ontology. Note that they are summarized by the *Pizza* (11) partial-area.

## Theory

We note the following:

- One can also derive a restriction-defined Partial-area Taxonomy using data properties used in class equivalence axioms (as above for *HighCaloriePizza* and *LowCaloriePizza*). Again, this option is enabled in the “Derivation Options” panel.
- Creating a refined restriction does not result in a new area or partial-area being created. Since areas and partial-areas are defined according to the set of object properties or data properties used in an ontology, refinement is not considered.
- In the above example, the restrictions that use *hasCaloricContentValue* that are defined on *HighCaloriePizza* and *LowCaloriePizza* are more specific than the restriction defined on *Pizza* that uses the *hasCaloricContentValue* data property.
- These classes are still summarized by the *Pizza* partial-area in the {*hasCaloricContentValue*} area since they do not introduce the use of any additional types of data properties.

The last exercises of the POT, **Exercise 50** and **Exercise 51**, relate to classifying instances and making the *hasCaloricContentValue* data property functional, respectively. They do not affect the Partial-area Taxonomy and are therefore omitted.