Pizza Ontology

“a review of core concepts for building a pizza ontology”

presented by:

Atif Khan

infoTrellis

http://www.infotrellis.com/

<u>presentation material based on:</u>

Horridge, Matthew.
"A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools – Edition1.3". The University of Manchester (2011).

Outline

- Disclaimer
  - I am **not** an ontology engineer
- Goal
  - duration ~ 30 mins
  - review some basic ontology components
    - concepts, object properties, data properties, individuals
    - classification
  - introduce Protégé ontology editor
  - share my experience building the Pizza Ontology

Core Terminology

- Ontology

> *"An ontology is a **formal**, **explicit** specification of a **shared** conceptualization"*
>
> *R. Studer (1998)*

- Components
  - *concepts* define **aggregation** of things
  - *individuals* are **instances** of concepts
  - *properties* ***link*** concepts/individuals

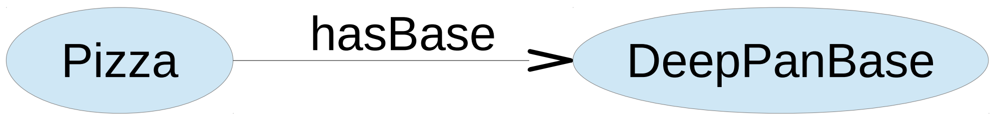
Core Terminology

- Triples
  - *a **representation** of ontological components*
    - using the following notation

Subject verb Object

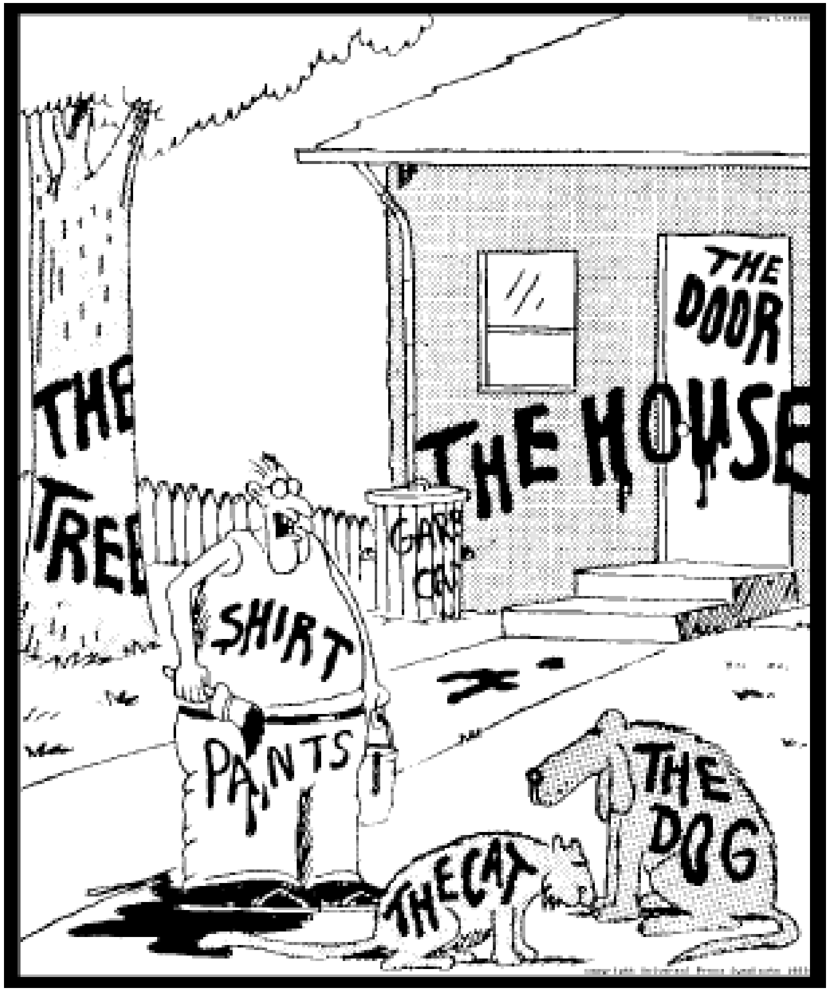
  - example: "*a pizza has a deep pan base*"

Pizza hasBase DeepPanBase



Why Use Ontologies

- Precision of:
  - representation/expression
  - information sharing
  - knowledge inference



“Now! *That* should clear up a few things around here!”

http://photos1.blogger.com/blogger2/1715/1669/1600/larson-oct-1987.gif

Creating a Pizza Ontology

Protégé

Version 4.2.0 (Build 284)

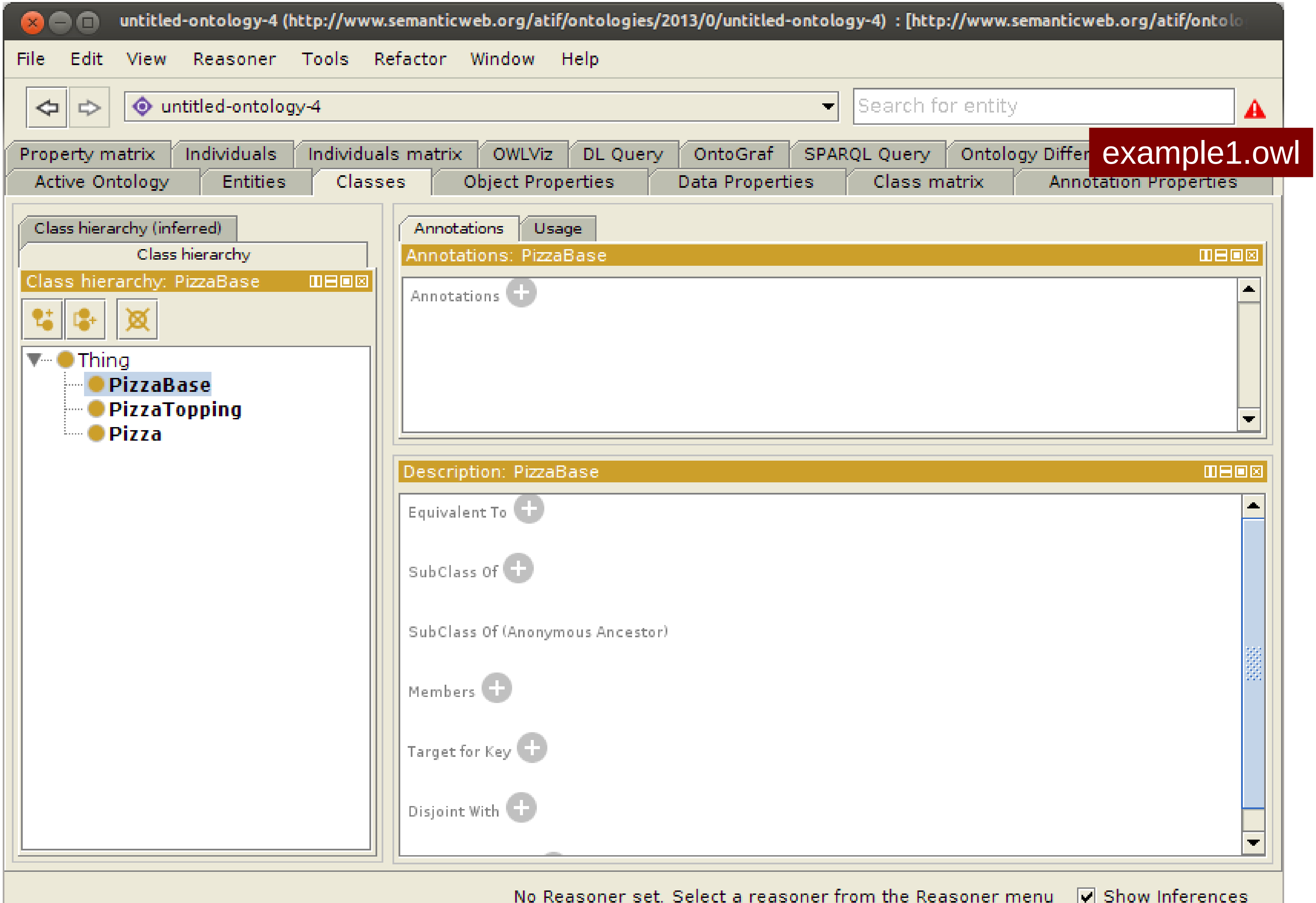
OWL

Web Ontology Language

http://www.w3.org/TR/owl-features/

Define Core Concepts

- Identify core concepts
  - *Pizza*
  - *Pizza Base*
  - *Pizza Toppings*

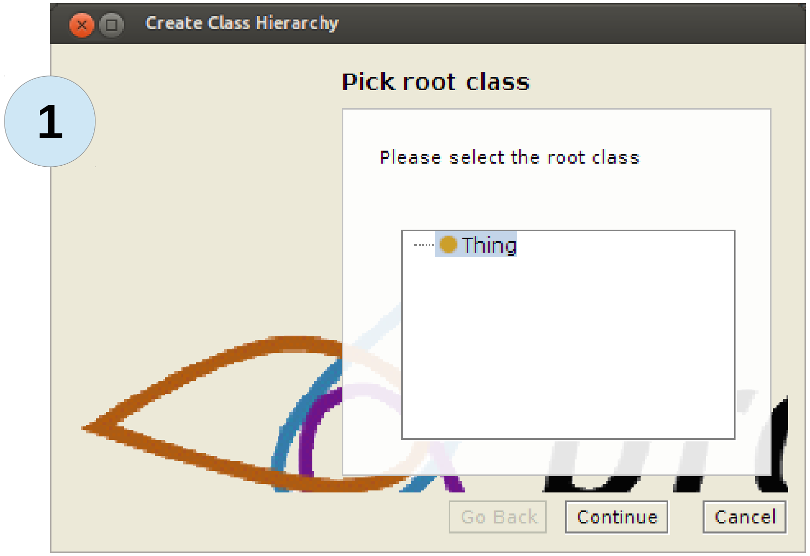


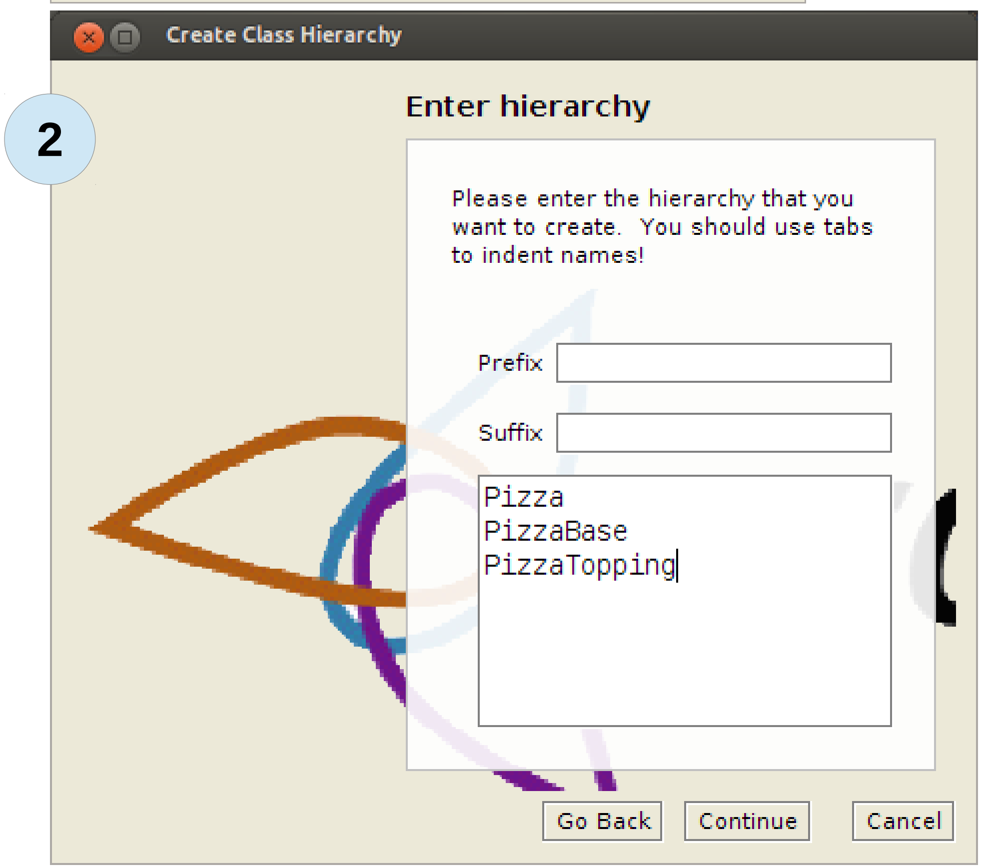
untitled-ontology-4 (http://www.semanticweb.org/atif/ontologies/2013/0/untitled-ontology-4) : [http://www.semanticweb.org/atif/ontolo
File Edit View Reasoner Tools Refactor Window Help
untitled-ontology-4
Search for entity
Property matrix
Individuals
Individuals matrix
OWLViz
DL Query
OntoGraf
SPARQL Query
Ontology Differ
example1.owl
Active Ontology
Entities
Classes
Object Properties
Data Properties
Class matrix
Annotation Properties
Class hierarchy (inferred)
Class hierarchy
Class hierarchy: PizzaBase
Thing
PizzaBase
PizzaTopping
Pizza
Annotations
Usage
Annotations: PizzaBase
Annotations
Description: PizzaBase
Equivalent To
SubClass Of
SubClass Of (Anonymous Ancestor)
Members
Target for Key
Disjoint With
No Reasoner set. Select a reasoner from the Reasoner menu
Show Inferences

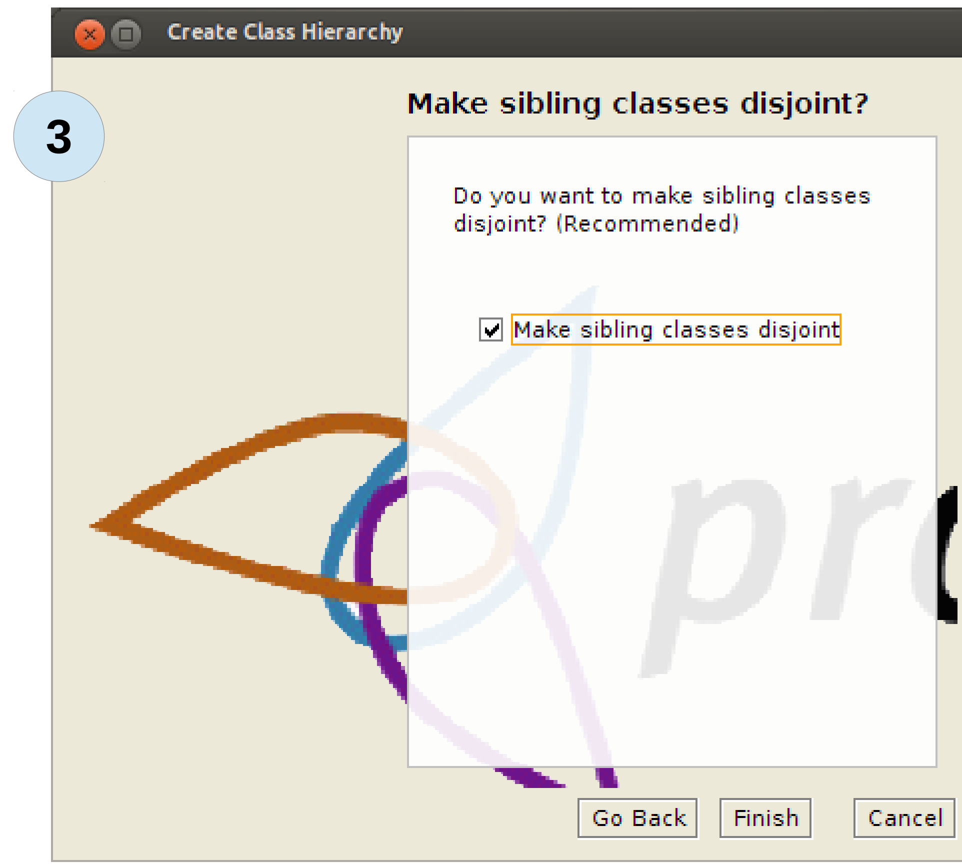
Define Core Concepts

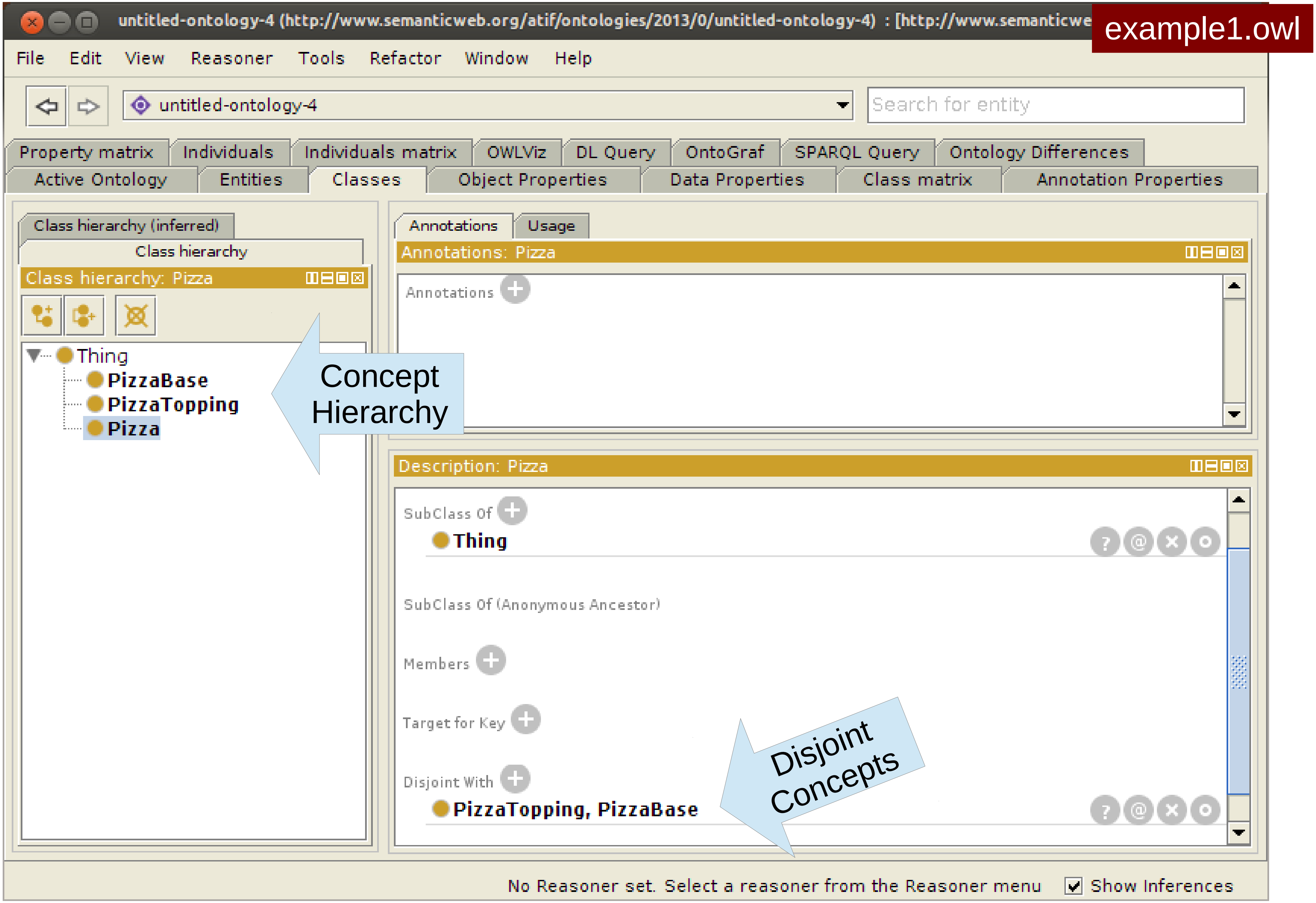
- Unique name assumption
  - need to explicitly define **same***ness* & **unique***ness* using
    - Equivalent to
    - Disjoint with

example1.owl








example1.owl
untitled-ontology-4 (http://www.semanticweb.org/atif/ontologies/2013/0/untitled-ontology-4) : [http://www.semanticwe
File Edit View Reasoner Tools Refactor Window Help
untitled-ontology-4
Search for entity
Property matrix
Individuals
Individuals matrix
OWLViz
DL Query
OntoGraf
SPARQL Query
Ontology Differences
Active Ontology
Entities
Classes
Object Properties
Data Properties
Class matrix
Annotation Properties
Class hierarchy (inferred)
Class hierarchy
Class hierarchy: Pizza
Thing
PizzaBase
PizzaTopping
Pizza
Concept Hierarchy
Annotations
Usage
Annotations: Pizza
Annotations
Description: Pizza
SubClass Of
Thing
SubClass Of (Anonymous Ancestor)
Members
Target for Key
Disjoint With
PizzaTopping, PizzaBase
Disjoint Concepts
No Reasoner set. Select a reasoner from the Reasoner menu
Show Inferences

Define Properties

- Link concepts using properties
  - a pizza has a deep pan base (*hasBase*)
  - a pizza has a mozzarella cheese topping (*hasCheeseTopping*)
  - a pizza has a tomato and cheese topping (*hasTomatoTopping*) and (*hasCheeseTopping*)
- Property Hierarchy

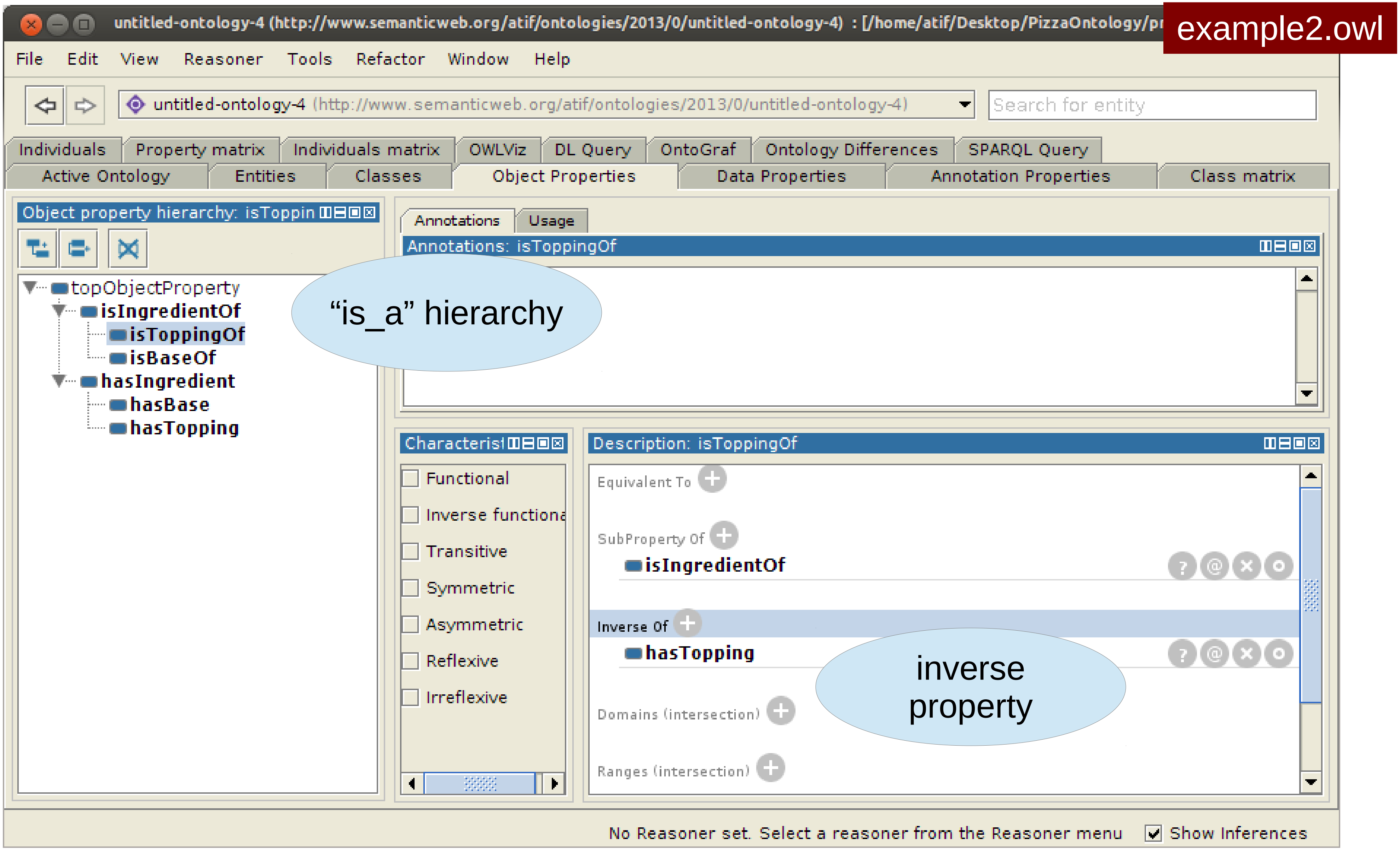
  hasBase ► hasIngrediant ◄hasTopping

Define Inverse Properties

- Inverse property
  - each object property may have a corresponding inverse property
  - "a pizza has a deep pan base"

    ≡ a deep pan is a base of a pizza

    (*isBaseOf*) is inverse of (*hasBase*)
    (*hasBase*) is inverse of (*isBaseOf*)



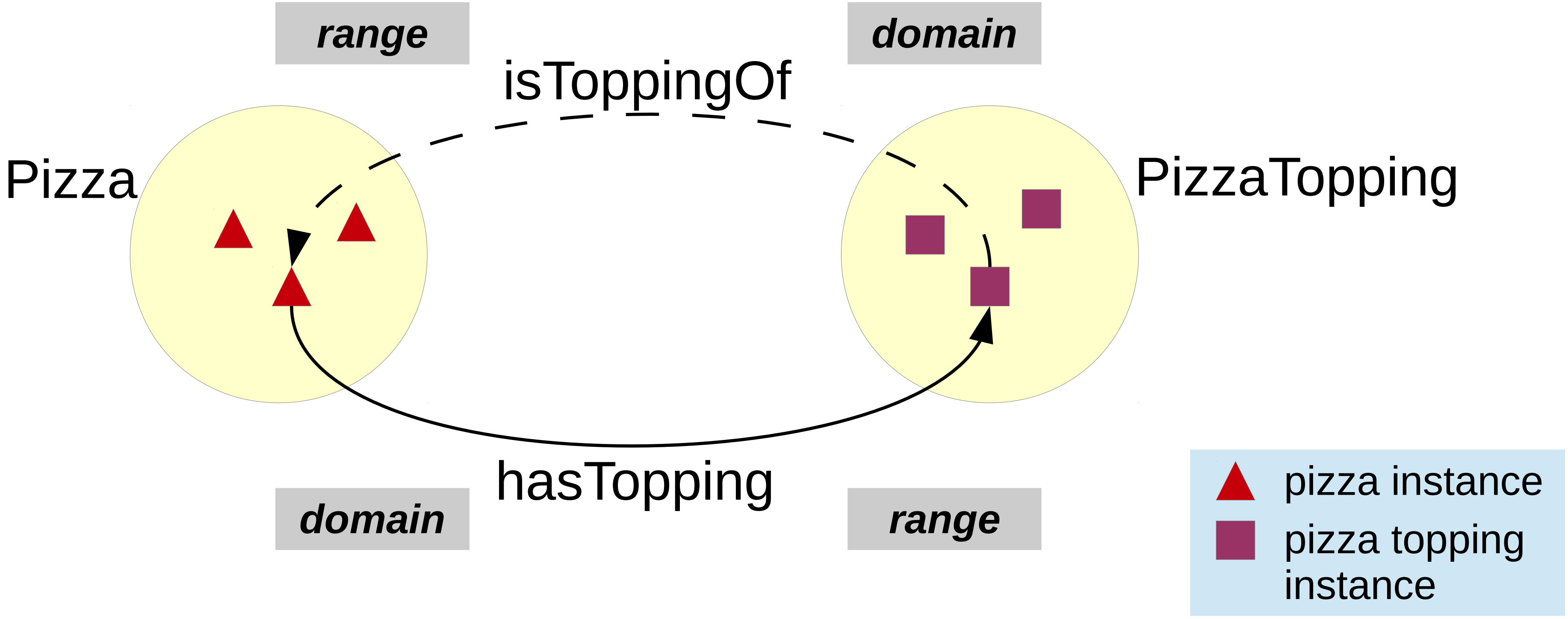
example2.owl
untitled-ontology-4 (http://www.semanticweb.org/atif/ontologies/2013/0/untitled-ontology-4) : [/home/atif/Desktop/PizzaOntology/pr
File Edit View Reasoner Tools Refactor Window Help
untitled-ontology-4 (http://www.semanticweb.org/atif/ontologies/2013/0/untitled-ontology-4)
Search for entity
Individuals Property matrix Individuals matrix OWLViz DL Query OntoGraf Ontology Differences SPARQL Query
Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Class matrix
Object property hierarchy: isToppin
topObjectProperty
isIngredientOf
isToppingOf
isBaseOf
hasIngredient
hasBase
hasTopping
“is_a” hierarchy
Annotations Usage
Annotations: isToppingOf
Characterist
Functional
Inverse functiona
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive
Description: isToppingOf
Equivalent To
SubProperty Of
isIngredientOf
Inverse Of
hasTopping
inverse property
Domains (intersection)
Ranges (intersection)
No Reasoner set. Select a reasoner from the Reasoner menu
Show Inferences

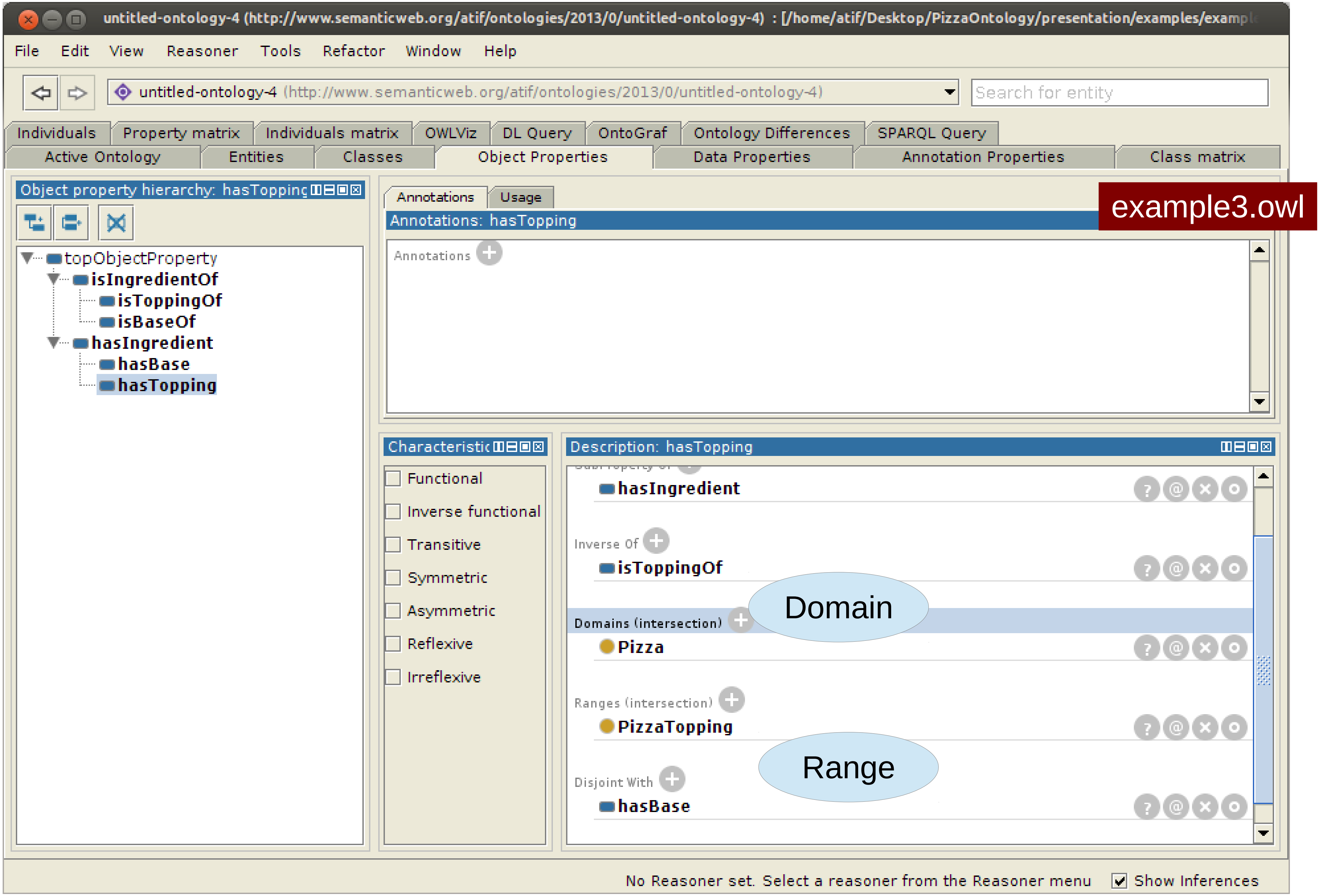
Characteristics of Properties

- OWL *primitives* to enrich relationship definitions (see §4.6)
  - functional & inverse functional
  - transitive
  - symmetric & anti-symmetric
  - reflexive & irreflexive

Property Domains & Ranges

- Definition
  - properties link individuals from the ***domain*** to individuals from the ***range***




untitled-ontology-4 (http://www.semanticweb.org/atif/ontologies/2013/0/untitled-ontology-4) : [/home/atif/Desktop/PizzaOntology/presentation/examples/exampl
File Edit View Reasoner Tools Refactor Window Help
untitled-ontology-4 (http://www.semanticweb.org/atif/ontologies/2013/0/untitled-ontology-4)
Search for entity
Individuals
Property matrix
Individuals matrix
OWLViz
DL Query
OntoGraf
Ontology Differences
SPARQL Query
Active Ontology
Entities
Classes
Object Properties
Data Properties
Annotation Properties
Class matrix
Object property hierarchy: hasTopping
topObjectProperty
isIngredientOf
isToppingOf
isBaseOf
hasIngredient
hasBase
hasTopping
Annotations
Usage
Annotations: hasTopping
example3.owl
Annotations
Characteristic
Functional
Inverse functional
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive
Description: hasTopping
hasIngredient
Inverse Of
isToppingOf
Domain
Domains (intersection)
Pizza
Ranges (intersection)
PizzaTopping
Range
Disjoint With
hasBase
No Reasoner set. Select a reasoner from the Reasoner menu
Show Inferences

Property Restrictions

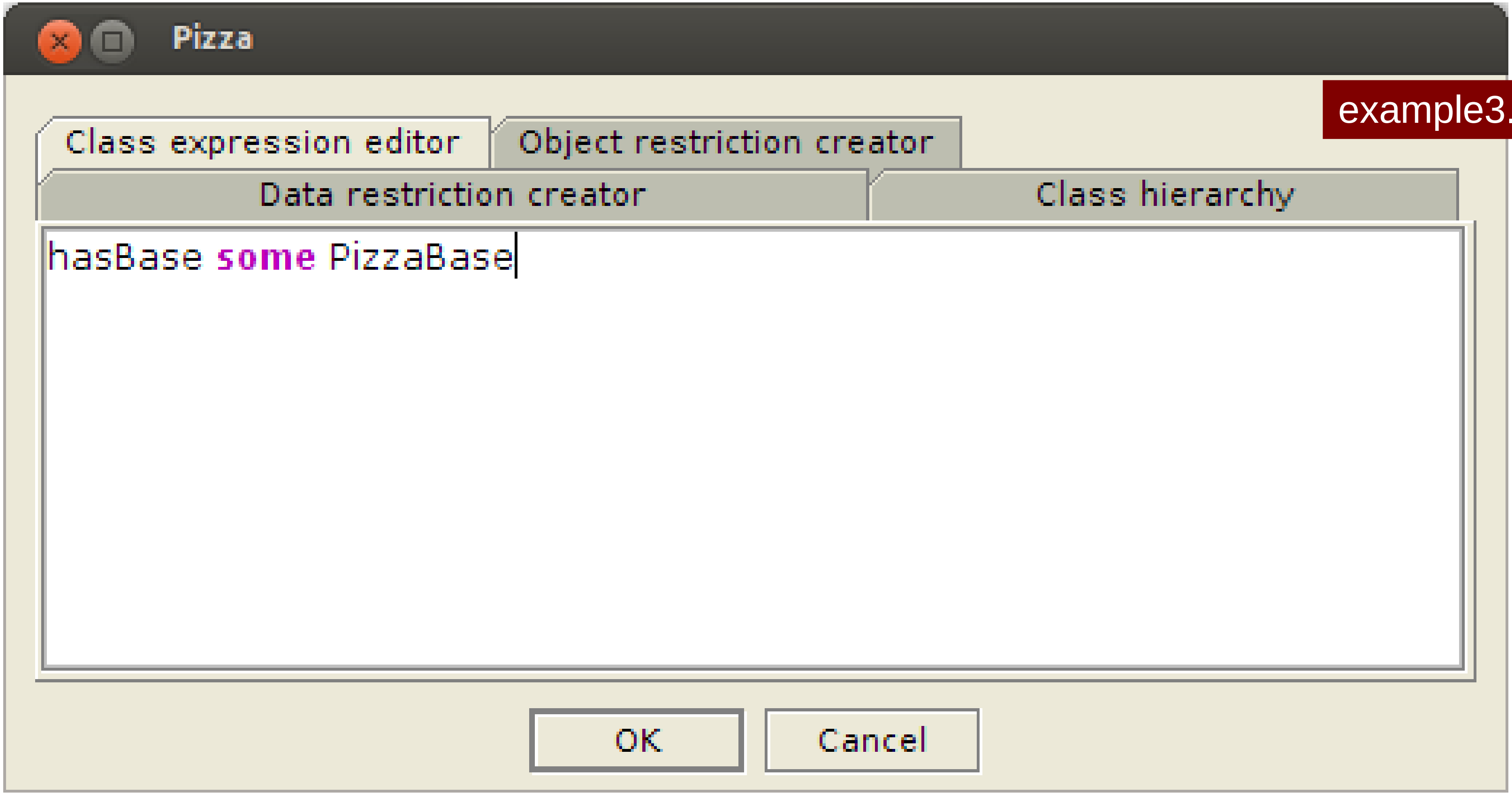
- Restriction ≈ Anonymous Class
  - a restriction is a class definition that groups individuals together based on one or more object properties
- Example
  - class of **individuals** that have **at least one** "***hasTopping***" relationship to individuals that are members of **MozzarellaTopping**

Existential Restrictions

- Intention
  - describe *"**<u>some</u>** values from"* restrictions

- Example
  - a pizza must have a pizza base

Existential Restrictions



example3.owl

Existential Restrictions

- Implication of "*hasBase some PizzaBase*"
  - if something is a Pizza then it is ***necessary*** for it to have a kind of PizzaBase

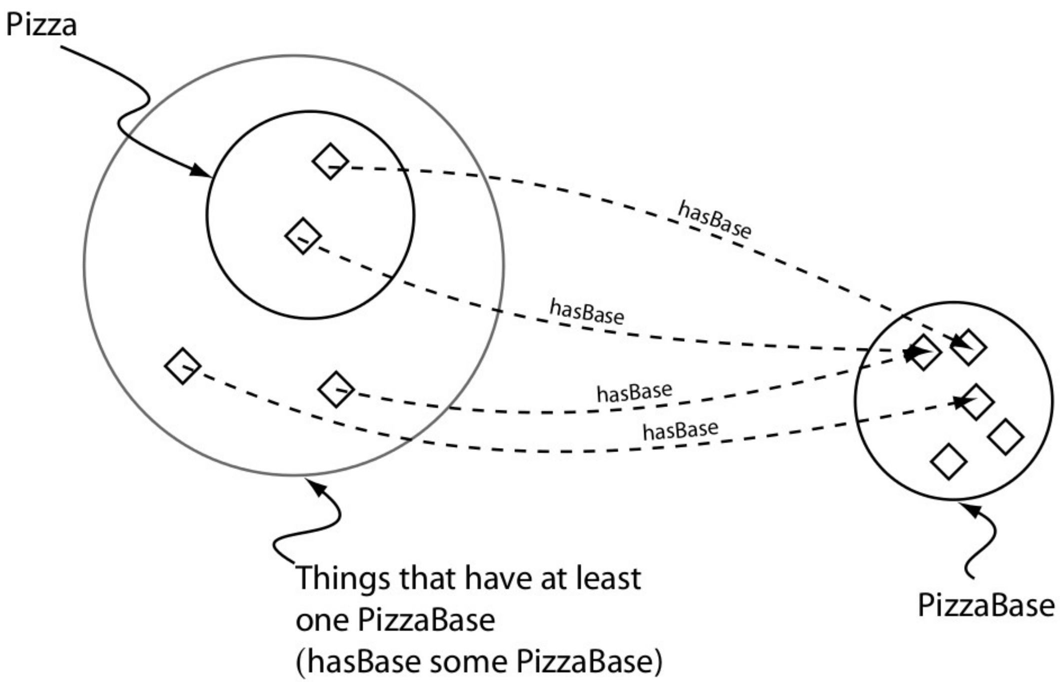


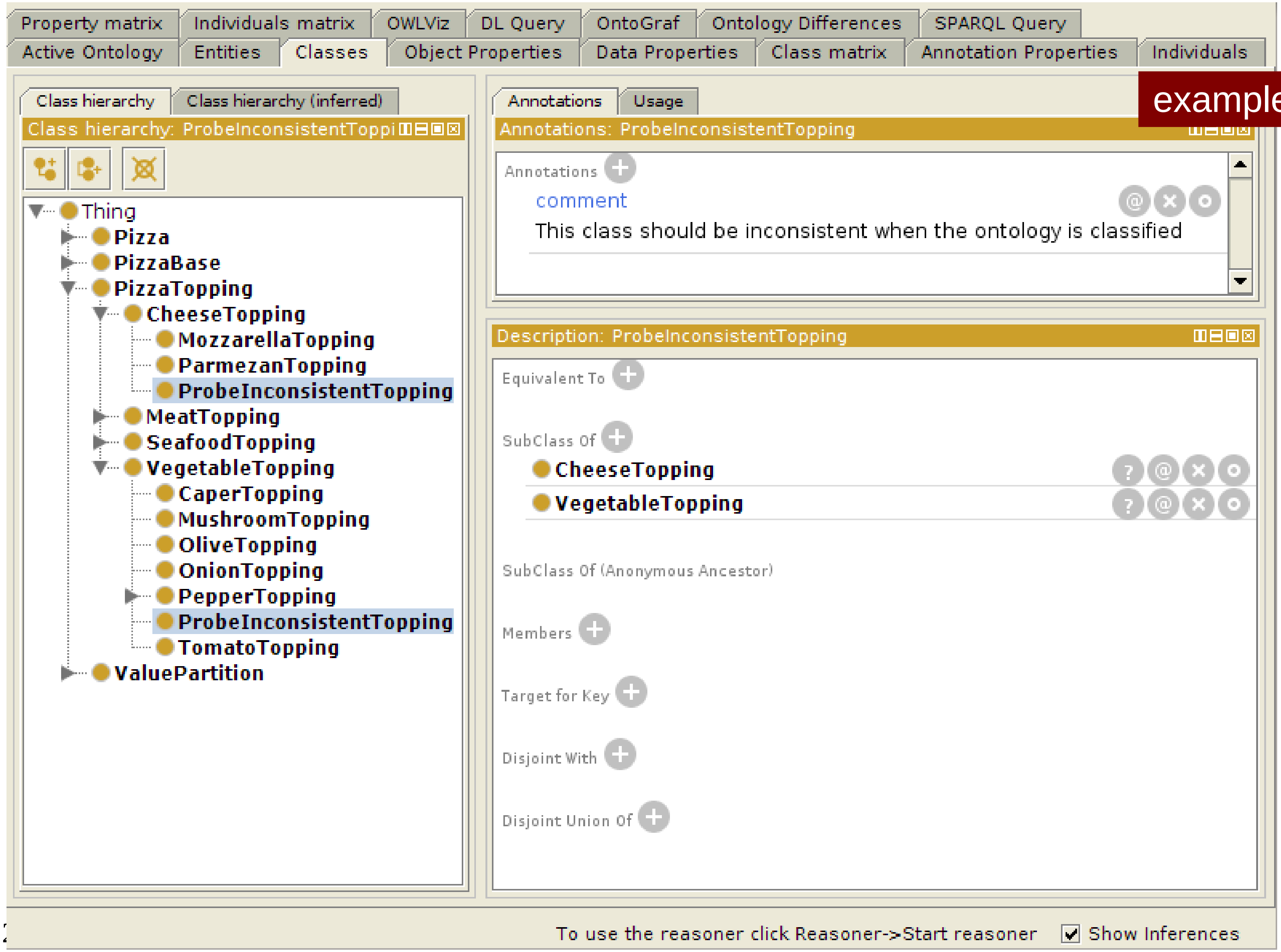
Figure 4.34: A Schematic Description of a Pizza — In order for something to be a Pizza it is necessary for it to have a (at least one) PizzaBase — A Pizza is a subclass of the things that have at least one PizzaBase

(M . Horridge, 2011)

Using the Reasoner (Classifier)

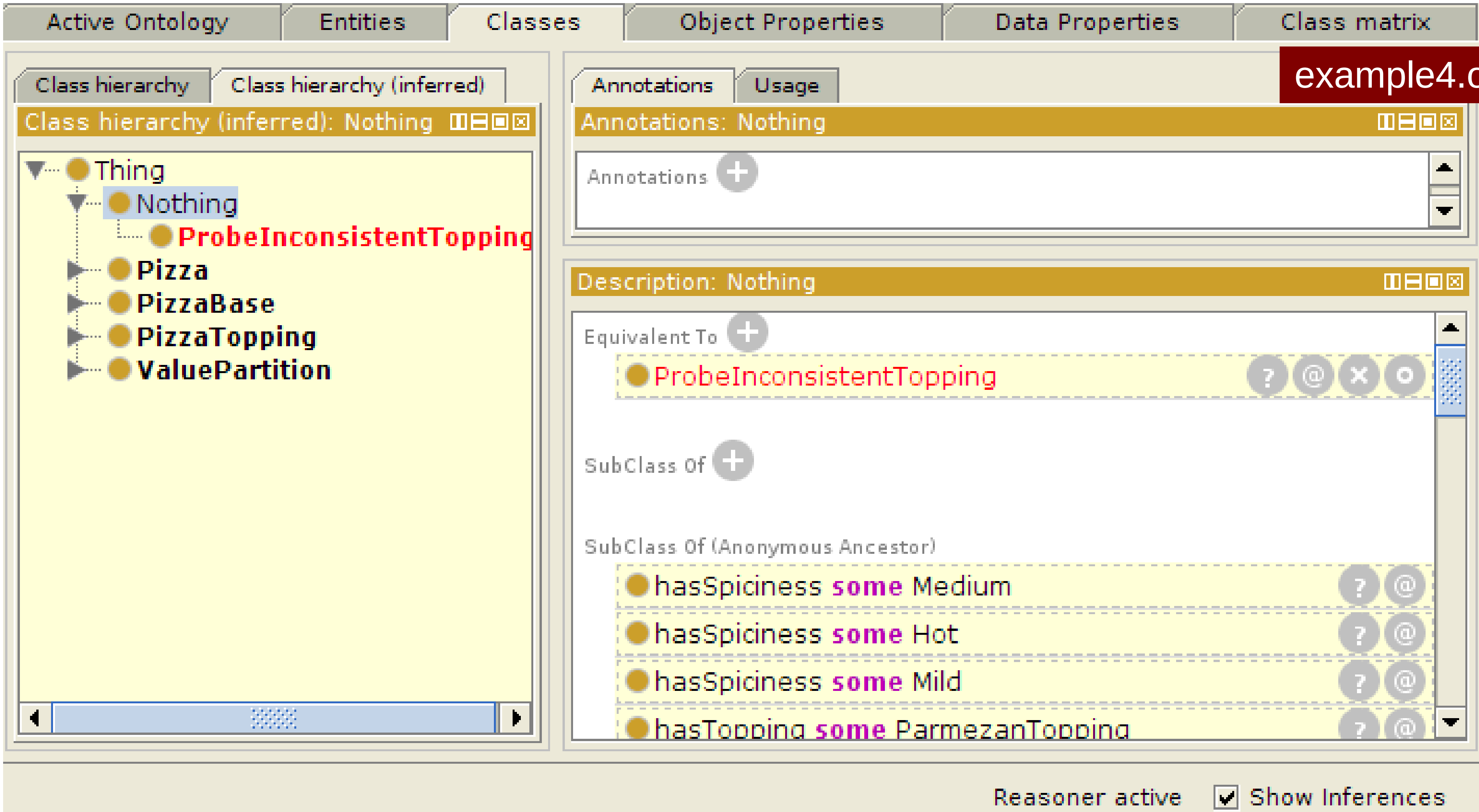
- Using a reasoner we can
  - determine class inconsistencies
    - e.g. inconsistent pizza
  - discovering implicit information
    - using necessary and sufficient conditions
    - e.g. cheesy pizza

Inconsistent Pizza Topping



example4.owl

Inconsistent Pizza Topping



example4.owl

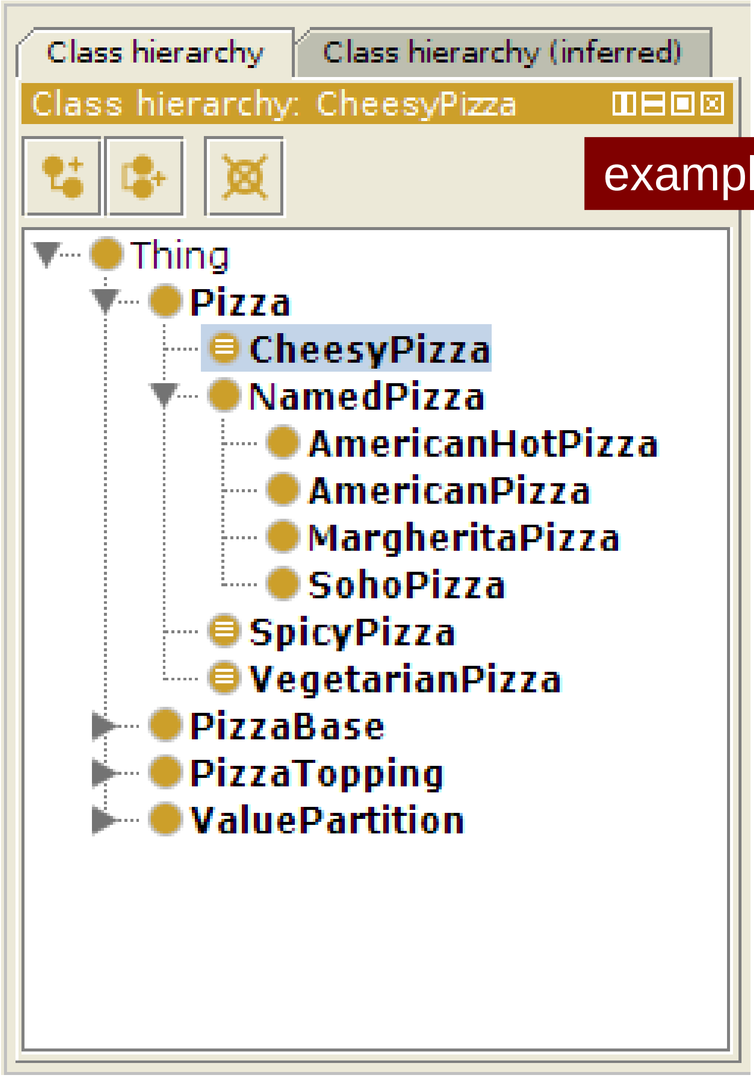
Using the Reasoner (Classifier)

- Using a reasoner we can
  - determine class inconsistencies
    - e.g. inconsistent pizza
  - discovering implicit information
    - using necessary and sufficient conditions
    - e.g. cheesy pizza

Cheesy Pizza

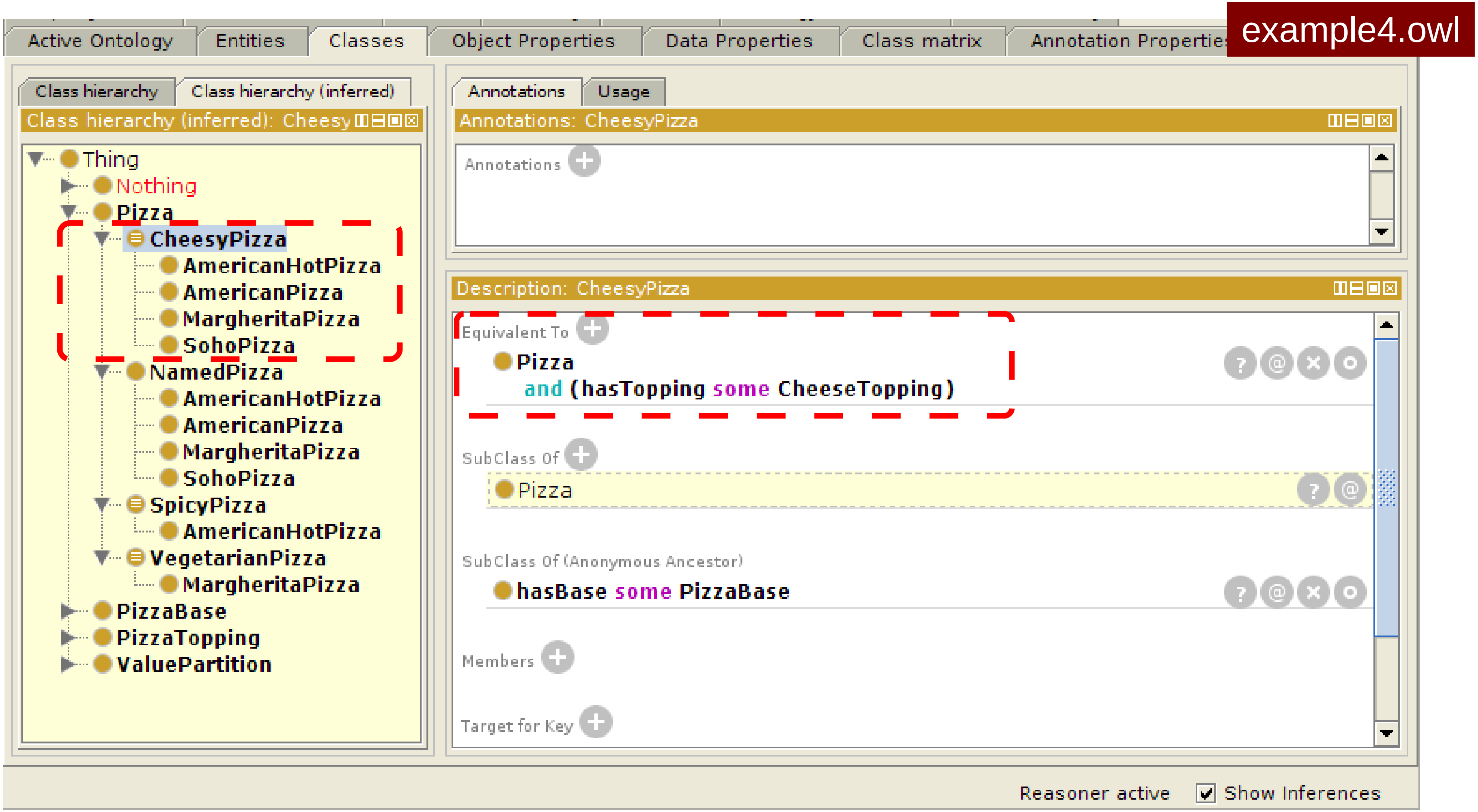
Explicit & Implicit definitions

- NamedPizza and its sub-classes are explicitly defined
- Discover sub-classes of CheesyPizza



example4.owl

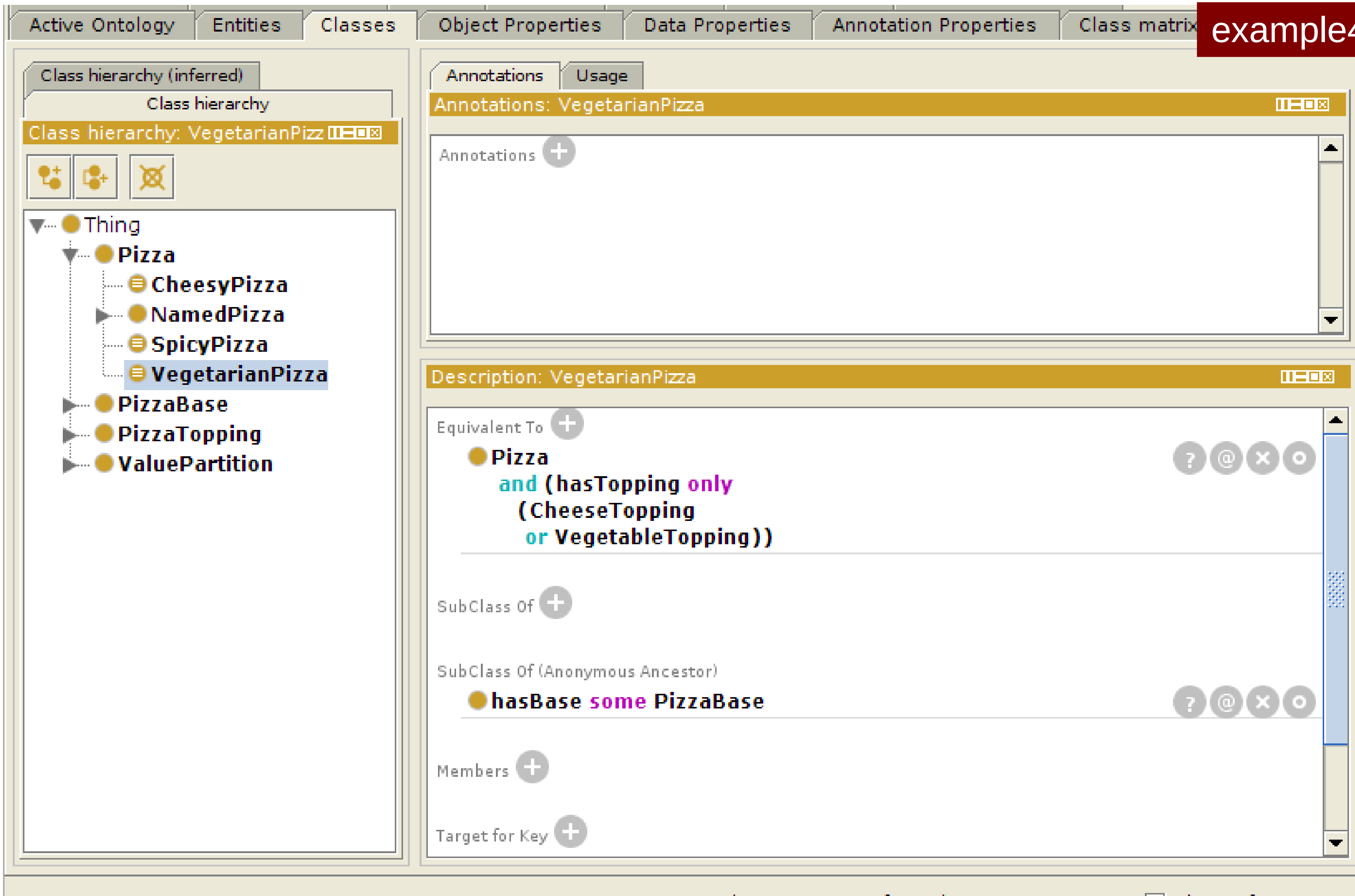
Cheesy Pizza



Universal Restrictions

- Intention
  - describe *“all and **only** values from”* restrictions
- Example
  - a “*vegetarian pizza*”
    can **only** have
    cheese or vegetable toppings

Universal Restrictions



example4.owl

Universal Restrictions

- Run the reasoner
  - expected behaviour:
    - Soho pizza and Margherita pizza should be classified as vegetarian pizzas
  - actual behaviour
    - reasoner does not find any vegetarian pizza subclasses

Open World Assumption

- OWA – What it means:
  - missing information is **<u>not</u>** confirmation of negation
  - in other words:
    - SohoPizza and MargheritaPizza toppings must be explicitly limited to their toppings

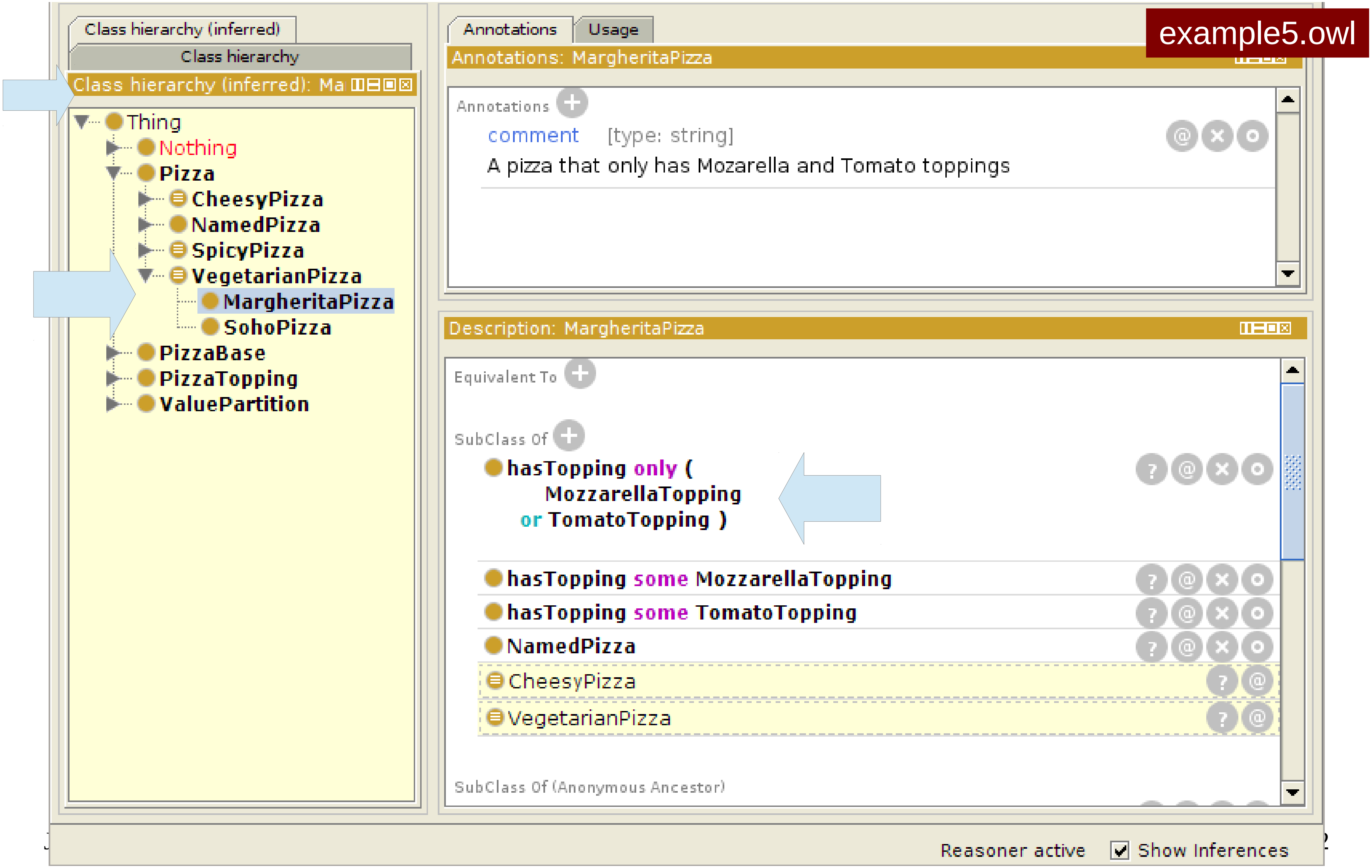
<u>SohoPizza:</u>

hasTopping **only** (
    MozzarellaTopping
  or TomatoTopping
  or OliveTopping
  or ParmezanTopping)

<u>MargheritaPizza:</u>

hasTopping **only** (
    MozzarellaTopping
  or TomatoTopping)

Universal Restrictions



Working with Protégé

- Protégé is simply an ontology IDE
  - editing
  - visualization
  - validation
- not required but extremely useful for
  - managing large ontologies
  - discovering existing ontologies