Lab: Ontology Engineering (2) – Protégé

Camilo Thorne

Room 00.012
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
+49 (0) 711 685-81369
camilo.thorne@ims.uni-stuttgart.de

Semantic Web, SS 2017
(based on slides by W. Kessler)

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

Protégé

- We really don't want to write RDF/RDFS/OWL by hand.
- Thankfully, there exist tools to help us!
- Protégé[1] is a well-known and powerful ontology editor.
- Protégé is free, open source and implemented in Java.
- Protégé is maintained by the Stanford Center for Biomedical Informatics Research and the community.

[1] http://protege.stanford.edu/

Institut für Maschinelle Sprachverarbeitung

Protégé

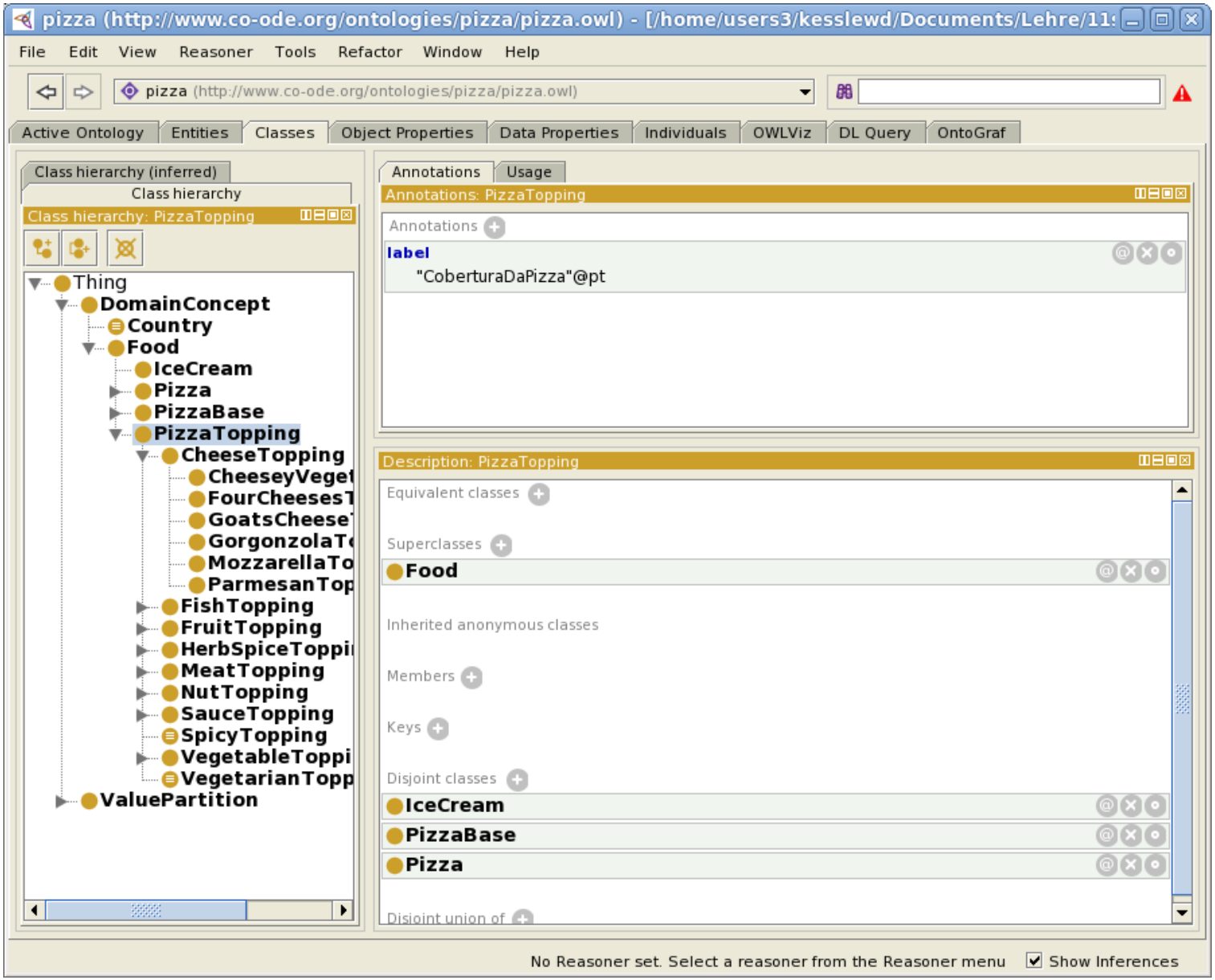
- The different parts of the ontology you are looking at are shown in different tabs (views).
- You choose which tabs to show/hide in “Window” – “Tabs”.
- Protégé allows plugins to add their own tabs.

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

The Classes View (1)



Institut für Maschinelle Sprachverarbeitung

The Classes View (2)

- The "Classes" view contains all information about the classes in the ontology.
- The "Class Hierarchy" windows shows the taxonomy of the ontology. You can add new sibling classes or subclasses with the buttons at the left.
- "Usage" displays where the selected class is being referred to.
- In the "Description" window you can add definitions by clicking on the "+" symbol.
  - You can chose a class from the "class hierarchy" tab.
  - You can use the GUI-creators for simple restrictions on object properties or data properties.
  - You can use the class expression editor with a special syntax
- You can edit definitions by clicking on the "@" symbol.

Institut für Maschinelle Sprachverarbeitung

The Classes View (3) – Class Definitions

Equivalent classes Equivalent classes defined by name or restrictions ("necessary and sufficient criteria").

Superclasses A list of classes defined by name or restrictions, that this class is a subclass of ("necessary criteria").

Inherited anonymous classes A list of conditions that have been inherited from superclasses.

Members All instances of this class.

Keys A value of a "key" unambiguously identifies a particular instance, no two instances of the class can have the same value in the "key" property.

Disjoint classes Disjoint classes defined by name or restrictions.

Disjoint union of If A is a disjoint union of B1 and B2, then everything that is a member of A and not a member of B1 is a member of B2.

Institut für Maschinelle Sprachverarbeitung

Protégé Syntax for Class Expressions

```
Pizza and (not (hasTopping some FishTopping))
and (not (hasTopping some MeatTopping))
and hasTopping min 3
```

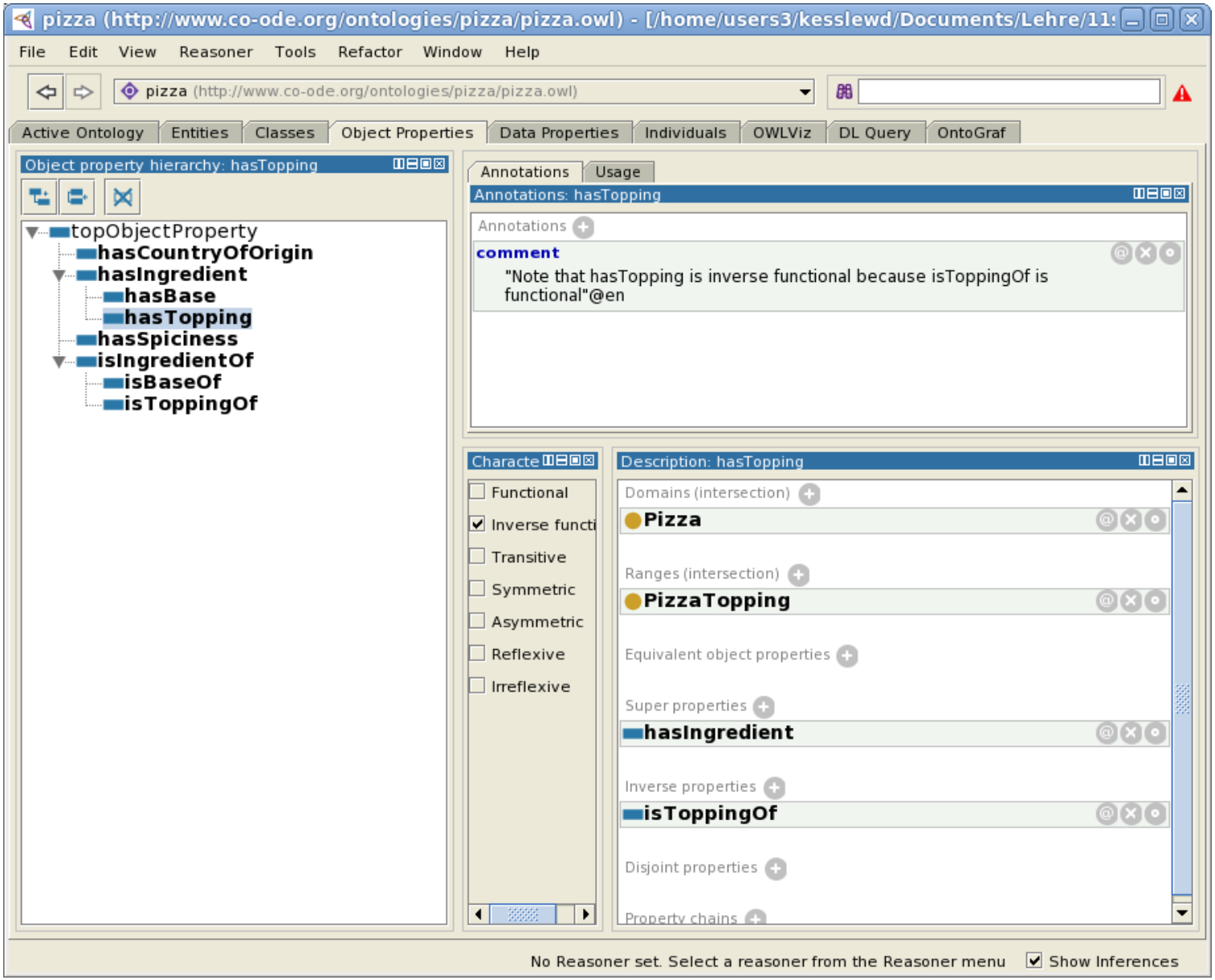
Name	Protégé Syntax	OWL Syntax
Existential restrictions	`some`	`owl:someValuesFrom`
Universal restrictions	`only`	`owl:allValuesFrom`
hasValue restriction	`value`	`owl:hasValue`
Intersection	`and`	`owl:intersectionOf`
Union	`or`	`owl:unionOf`
Complement	`not`	`owl:complementOf`
Minimum cardinality	`min`	`owl:minCardinality`
Maximum cardinality	`max`	`owl:maxCardinality`
Cardinality	`exactly`	`owl:cardinality`

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

The Properties Views (1)



Institut für Maschinelle Sprachverarbeitung

The Properties Views (2)

- Object properties and data properties are separated into the views "Object Properties" and "Data properties".
- The "Data/Object Property Hierarchy" windows shows the taxonomy of the properties. You can add new sibling properties or subproperties with the buttons at the left.
- "Usage" displays where this property is being referred to.
- In the "Description" window you can add domain, range, superproperties and inverse properties by clicking on the "+" symbol.
- In the "Characteristics" window a property can be defined as transitive, (a)symmetric, (ir)reflexive or (inverse) functional[2].

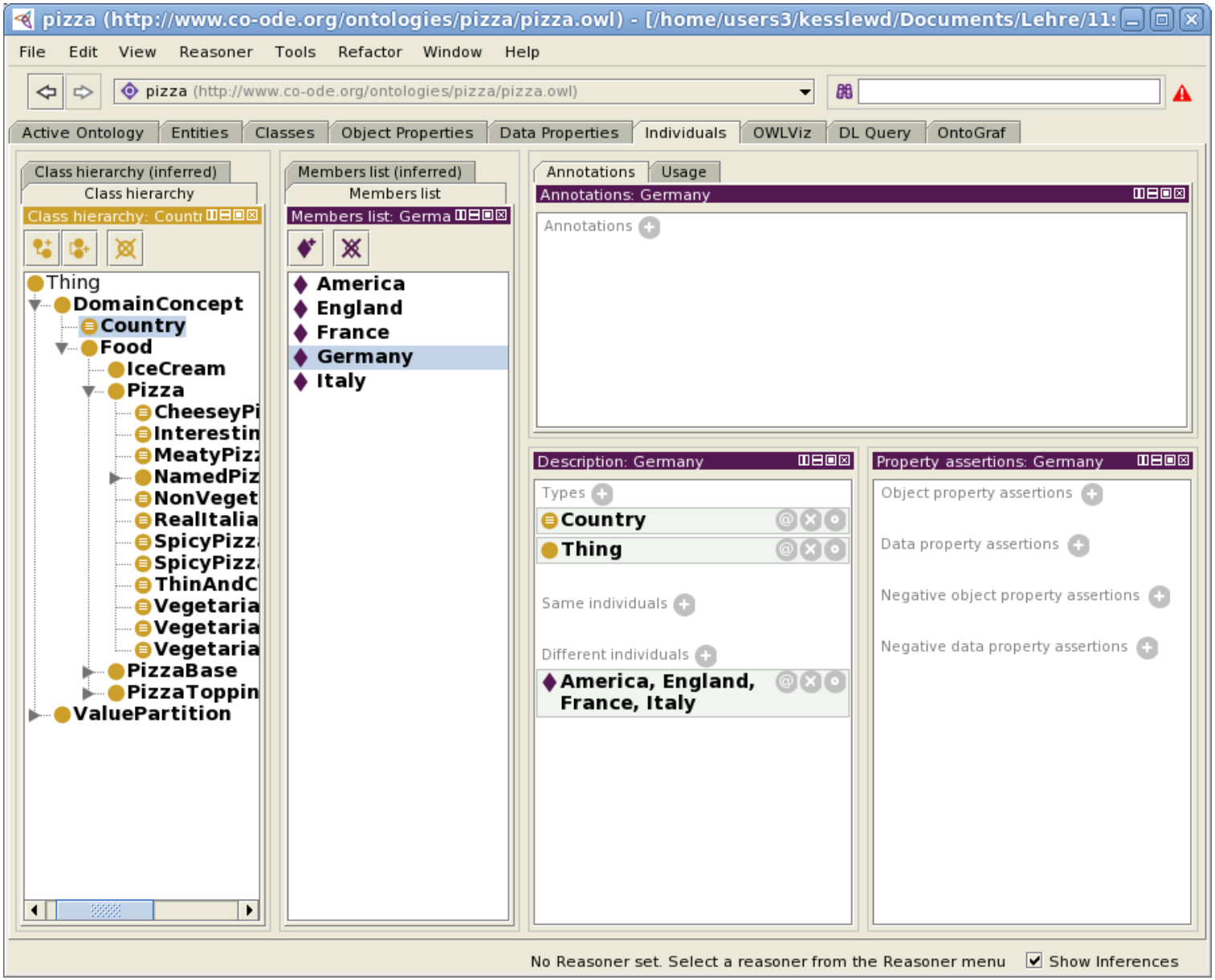
[2] If a property is functional, for a given individual, there can be at most one other individual that is related to this individual via the functional property.

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

The Instances View (1)



Institut für Maschinelle Sprachverarbeitung

The Instances View (2)

- Instances are managed in the view "Individuals".
- The "Class Hierarchy" windows shows the taxonomy of the classes.
- When you select a class, the "Members list" window will show all individuals that are instances of this class.
- "Usage" displays where the selected individual is being referred to.
- In the "Description" window you can add types (which classes the individual is an instance of), same and different individuals.
- In the "Property assertions" window you can add object and data properties for the individual (e.g., hasPrice 5.0, ...).

Institut für Maschinelle Sprachverarbeitung

Outline

Institut für Maschinelle Sprachverarbeitung

Protégé's other Views (1)

Active Ontology Shows an overview of the active ontology, including metrics on its contents, annotations about the ontology as a whole, and other imported ontologies (if any imports exist).

Entities An overview of the whole ontology containing classes, properties and instances.

Annotation Properties Properties that are only "comments" for humans to describe the ontology and its parts and not used in reasoning.

Institut für Maschinelle Sprachverarbeitung

Protégé's other Views (2)

OWLViz Shows a graphical representation of part of the class hierarchy when you click on a class. Requires installation of Graphviz before anything will be visible.

OntoGraf Visualization of the classes of the ontology.

DLQuery An arbitrary class description in Description Logics can be entered and the reasoner is queried for the sub/super classes, inferred members, etc. Requires the reasoner to run beforehand.

SPARQL Query Allows the execution of arbitrary SPARQL queries on the ontology.

Institut für Maschinelle Sprachverarbeitung

Outline

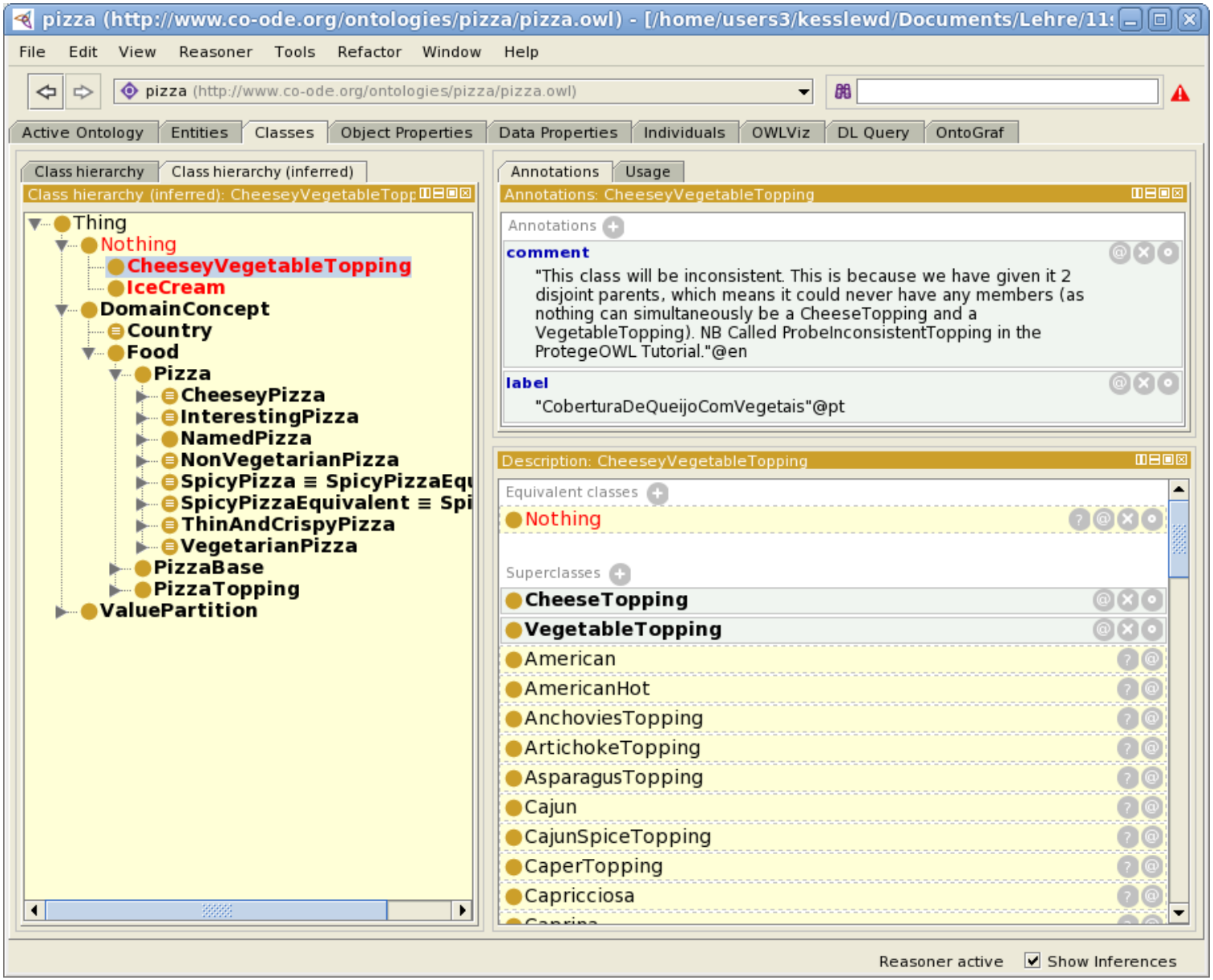
Institut für Maschinelle Sprachverarbeitung

Protégé's Reasoner

- A reasoner can test whether or not one class is a subclass of another class.
  → This results in an inferred class hierarchy.
- A reasoner can also test for consistency, based on the description (restrictions) of a class the reasoner can check whether or not it is possible for the class to have any instances.
  → Inconsistent classes are subclasses of `Nothing`.
- Protégé has different reasoners, select one in the menu "Reasoner" and use "Start Reasoner" to start the test.

Institut für Maschinelle Sprachverarbeitung

The Pizza Ontology in Protégé's Reasoner



Institut für Maschinelle Sprachverarbeitung

The Inferred Class Hierarchy Tab

- The "inferred class hierarchy" tab has a yellow background, as well as all classes added to the description by the reasoner.
- Classes that have changed their superclasses in the hierarchy are marked in blue (e.g., `Caprina` is now a subclass of `VegetarianPizza`).
- Inconsistent classes are marked in red and are subclasses of `Nothing` (e.g., `IceCream`).
- Equivalent classes are marked with $\equiv$ (e.g., `SpicyPizza` and `SpicyPizzaEquivalent`).

Institut für Maschinelle Sprachverarbeitung