



Protégé

Ontology Building Environment

Prof Christophe Roche

ERA Chair Holder - University of Crete

<https://christophe-roche.fr/>

May 2025

<https://talos-ai4ssh.uoc.gr/>



Horizon ERA Chair TALOS AI4SSH Project funded by the European Commission
Grant Agreement n° 101087269, <https://cordis.europa.eu/project/id/101087269>



Contents

0. Introduction

1. Definitions

- Ontology

2. Principles

- Individual
- Property
- Class

3. Example

- Greek Vases

4. Ontology Building

- Class Hierarchy
- Property Restriction
- Populating
- Annotating

5. Open Questions

- Definition
- Linguistic Dimension

6. Conclusion

The screenshot displays the Protégé ontology editor interface. The top navigation bar includes tabs for 'Active ontology', 'Individuals by class', 'Classes', 'OntoGraf', and 'Data properties'. The main workspace is divided into several panels:

- Class hierarchy:** Shows a tree structure starting with 'owl:Thing', followed by 'Krater', and then subclasses: 'Bell_krater', 'Calyx_krater', 'Column_krater', and 'Volute_krater'. 'Bell_krater' is currently selected.
- Diagram:** A central diagram shows a vase with two smaller vase parts. Arrows labeled 'hasPart' connect the main vase to each of the smaller parts.
- Annotations:** A panel titled 'Annotations: 215332' shows a list of properties for the selected instance. A red circle highlights the 'Date' property, which is '-450 to -400'. Other properties include 'Vase Number: 215332', 'Fabric: ATHENIAN', 'Technique: RED-FIGURE', 'Shape Name: KRATER, BELL', 'Attributed To: Compare DINOS P by BEAZLEY', 'Registration: A: SACRIFICE, DRAPED MEN, ONE WITH STAFF, DRAPED YOUTH WITH SACRIFICIAL BASKET, H B: DRAPED YOUTHS', 'Current Collection: Athens, National Museum: CC1', 'Previous Collections: Athens, National Museum: 1466', and 'Publication Record: Beazley, J.D., Attic Red-Figure 1963): 1158'.
- Description:** A panel titled 'Description: 215332' shows the class 'Bell_krater' and various assertion options like 'Same Individual As' and 'Different Individuals'.
- Property assertions:** A panel titled 'Property assertions: 215332' shows 'vase_number 215332' and other assertion options.
- Direct instances:** A panel titled 'Direct instances: 215332' shows the instance '215332' for the class 'Bell_krater'.
- Diagram annotations:** The diagram includes labels like 'rdfs:label' pointing to a text box containing '"bell krater"@en', 'rdf:type' pointing to the 'Bell_krater' class, 'attributedTo' pointing to a painter's portrait, and 'foaf:name' pointing to a text box containing '"Gorgon Painter"@en'.



0. Introduction



<https://protege.stanford.edu/>

- ✓ Protégé is a free, open-source ontology editor written in Java developed at Stanford University
- ✓ More than 300,000 users are registered.

- ✓ *W3C standards compliant*
- ✓ *Customizable user interface*
- ✓ *Visualization support*
- ✓ *Ontology refactoring support*
- ✓ *Direct interface to reasoners*
- ✓ *Highly pluggable architecture*
- ✓ *Cross compatible with WebProtégé*

WHY PROTÉGÉ

Protégé's plug-in architecture can be adapted to build both simple and complex ontology-based applications. Developers can integrate the output of Protégé with rule systems or other problem solvers to construct a wide range of intelligent systems. Most important, the Stanford team and the vast Protégé community are here to help.



ACTIVE COMMUNITY

Protégé is actively supported by a strong community of users and developers that field questions, write documentation, and contribute plug-ins.



W3C STANDARDS SUPPORT

Protégé fully supports the latest OWL 2 Web Ontology Language and RDF specifications from the World Wide Web Consortium.



EXTENSIBLE OPEN SOURCE ENVIRONMENT

Protégé is based on Java, is extensible, and provides a plug-and-play environment that makes it a flexible base for rapid prototyping and application development.



1. Definitions

Ontology

An ontology is a formally-defined vocabulary for a particular domain of interest. Ontologies are typically based on a class hierarchy (asserted and/or inferred), supplemented by assorted properties.

https://protegewiki.stanford.edu/wiki/Pr4_UG_mi_Glossary#Ontology



Ontologies are used to capture knowledge about some domain of interest. An ontology describes the concepts in the domain and also the relationships that hold between those concepts.

A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools - Edition 1.3

Language

OWL provides the theoretical basis for Protégé ontologies.

https://protegewiki.stanford.edu/wiki/Pr4_UG_mi_Glossary#Ontology

Different ontology languages provide different facilities. The most recent development in standard ontology languages is OWL from the World Wide Web Consortium (W3C).

A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools - Edition 1.3

Reasoner

The logical model allows the use of a reasoner which can check whether or not all of the statements and definitions in the ontology are mutually consistent and can also recognise which concepts fit under which definitions. The reasoner can therefore help to maintain the hierarchy correctly.

A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools - Edition 1.3



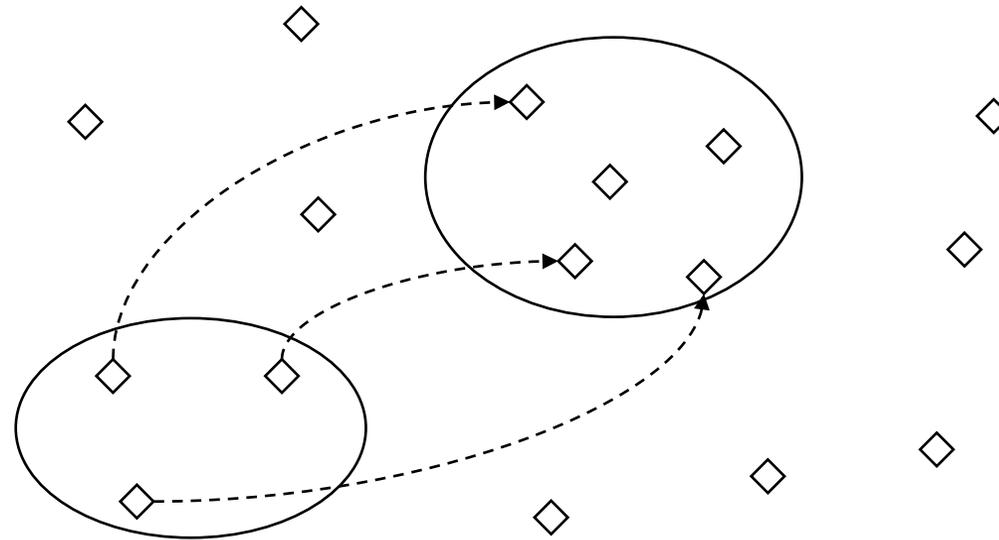
an inconsistent class is a class which cannot contain any individual because of its definition



2. Principles

Extensional Logic

Goal: **Organising** the objects which populate the world into **classes** according to the **relationships** that linked objects together



⇒ An object is not defined by its “nature“, but by its relations with other objects



2. Principles

Components of OWL Ontologies

1) Individuals

Individuals, represent objects in the domain in which we are interested



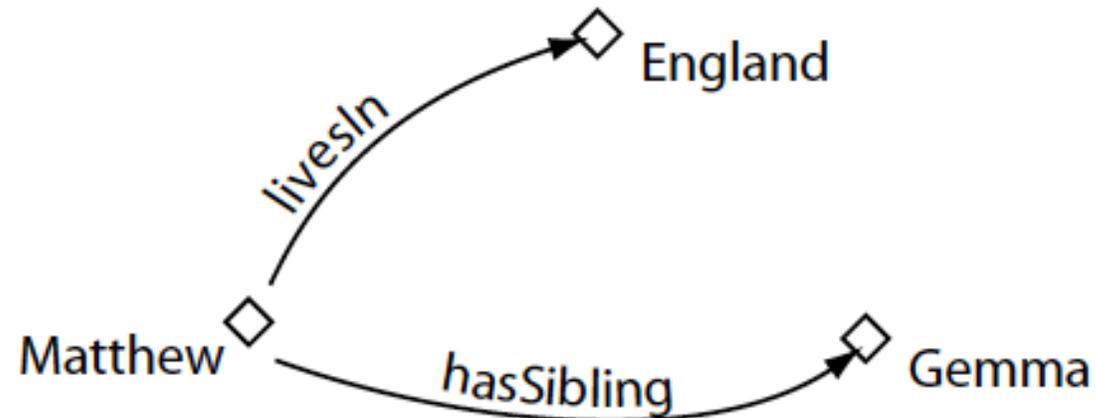
Terminology: « individual », « instance », « object »

2. Principles

Components of OWL Ontologies

2) Object Properties

Properties are binary relations on individuals, i.e. properties link two individuals together.



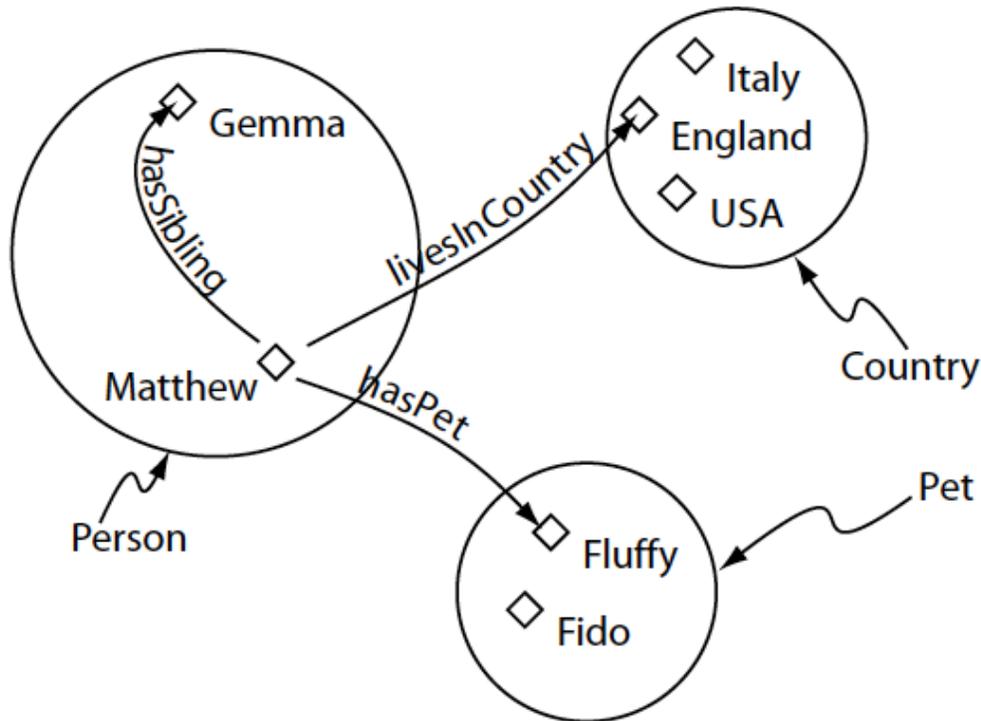
Terminology: « properties », « slots » (Protégé), « roles » (DL), « relations », « attributes »

2. Principles

Components of OWL Ontologies

3) Classes

OWL classes are interpreted as sets that contain individuals.



Classes are a concrete representation of concepts.

Classes are defined using formal descriptions that state precisely the requirements for membership of the class.

Person = ?

Person = { $x / \exists y \text{ Country}(y) \wedge \text{livesInCountry}(x,y)$ }

Person = { $x / \exists y \text{ Pet}(y) \wedge \text{hasPet}(x,y)$ }

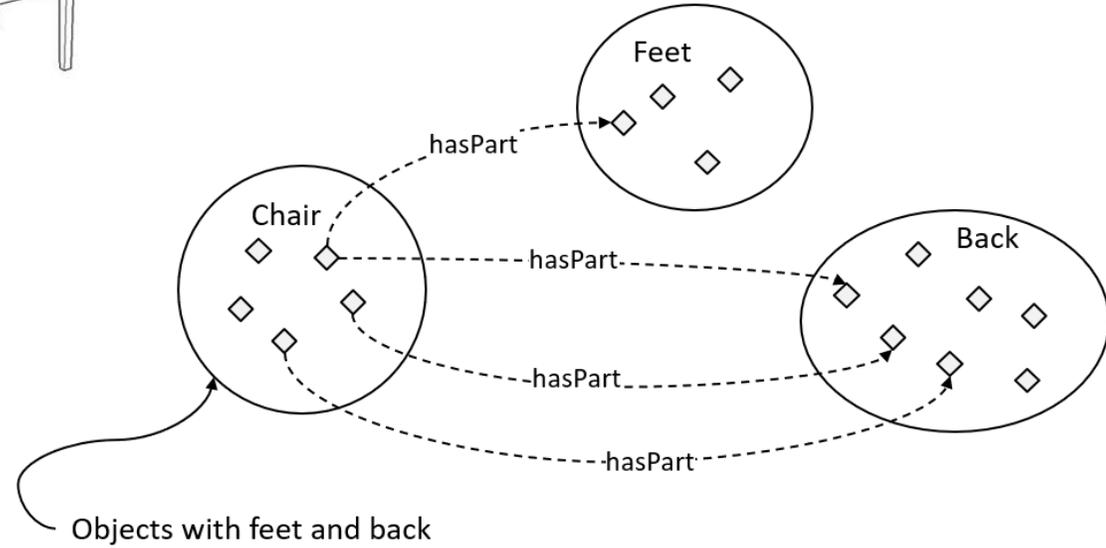
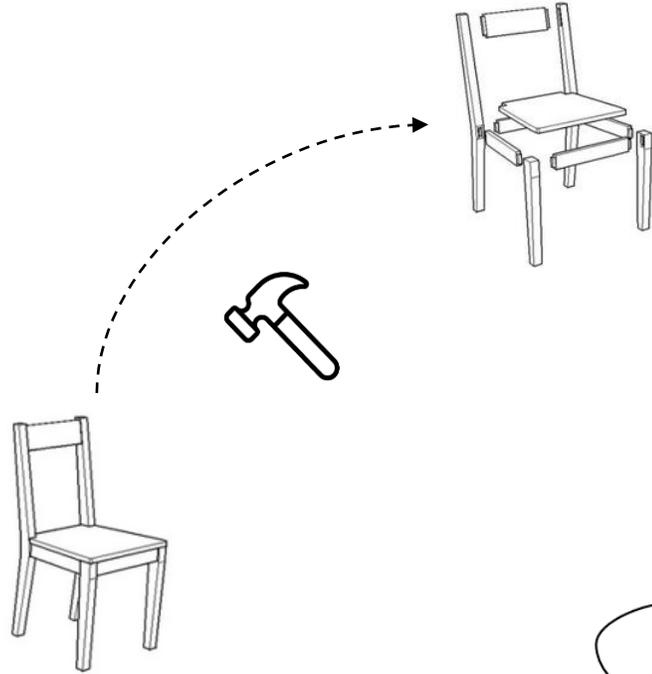
Person = { $x / \exists y \text{ hasSibling}(x,y) \vee \text{hasSibling}(y,x)$ }



2. Principles



Change your way of thinking



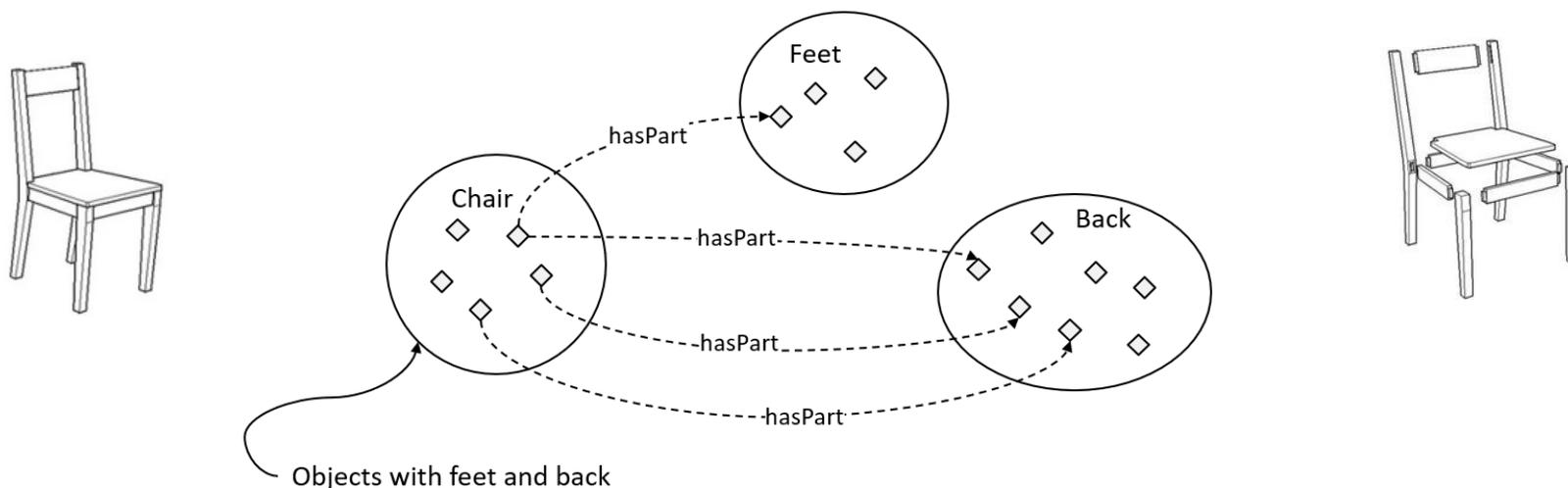
2. Principles



4) Property restriction

A means to define classes of individuals

- a) **Existential Restrictions:** describes (anonymous) classes of individuals that participate in *at least one* (some) relationship along a specified property to individuals that are members of a specified class.



- b) **Universal Restrictions:** describes (anonymous) classes of individuals that for a given property *only* (only) have relationships along this property to individuals that are members of a specified class (all values of the property must be of a certain type)

- c) **Has value:** at least one of the values of the property is a certain value

3. Example

CLASSICAL ART RESEARCH CENTRE

Beazley Archive: <https://www.carc.ox.ac.uk/carc/Home>



The Krater Ontology

“The term '**krater**' suggests a mixing-vessel (compare Greek *kerannumi* - to mix), and we know that the wine served at the symposium was mixed with water.



Column-krater: Named for its column-like handles, the column-krater is first known from Corinthian examples dated to the late seventh century. It is regularly produced by Athenian potters from the first half of the sixth-century until the third quarter of the fifth. It seems from graffiti on Athenian red-figure examples that the vessel was referred to as *Korinthios* or *Korinthiourges*.



Volute-krater: The volute-krater is named after its handles. The François Vase is a famous and early example, but the typical Athenian form occurs only later in the sixth century, with the handles tightly curled so that they look like the volutes on Ionic columns. The shape is also found in metal. Over the course of the fifth and fourth centuries, examples become slimmer, and Apulian volute-kraters from South Italy are particularly elaborate.



Calyx-krater: The handles of the calyx-krater are placed low down on the body, at what is termed the *cul*. Their upward curling form lends the shape an appearance reminiscent of the calyx of a flower, hence the name. The earliest known example was possibly made by Exekias in the third quarter of the sixth century. It continues to be produced, mainly in red-figure, becoming more elongated over the course of the fifth and fourth centuries.



Bell-krater: The latest of the four krater-types, it first occurs in the early fifth century, and is not found decorated in black-figure. It is named for its bell-like shape, perhaps originating in wood. It has small horizontal upturned handles just over halfway up the body. Some do not have a foot, and earlier examples may have lugs for handles.





4. Ontology Building



untitled-ontology-2 (http://www.semanticweb.org/roche/ontologies/2025/3/untitled-ontology-2) : [http://www.semanticweb.org/roche/on...

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-2 (http://www.semanticweb.org/roche/ontologies/2025/3/untitled-ontology-2)

Active ontology x Entities x Individuals by class x DL Query x

Ontology header: ? || ≡ × **Ontology metrics:** ? || ≡ ×

Ontology IRI http://www.semanticweb.org/roche/ontologies/2025/3/untitled-ontology-2

Ontology Version IRI e.g. http://www.semanticweb.org/roche/ontologies/2025/3/untitled-ontology-

Annotations +

Metrics

Axiom	0
Logical axiom count	0
Declaration axioms count	0
Class count	0
Object property count	0
Data property count	0
Individual count	0
Annotation Property count	0

Class axioms

SubClassOf	0
EquivalentClasses	0
DisjointClasses	0
GCI count	0
Hidden GCI Count	0

Ontology imports **Ontology Prefixes** General class axioms

Imported ontologies: ? || ≡ ×

Direct Imports +

Indirect Imports

No Reasoner set. Select a reasoner from the Reasoner menu Show Inferences ≡





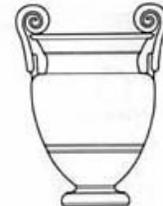
4. Ontology Building



The screenshot shows the Protégé ontology editor with the 'Window' menu open. The 'Classes' sub-menu is selected, showing the following options:

- Active ontology
- Entities
- Classes
- Object properties
- Data properties
- Annotation properties
- Individuals by class
- OWLviz
- DL Query
- AOWLN
- OntoGraf
- SWRLTab
- OWLax
- SQWRLTab
- SPARQL Query
- DisjointClasses
- GCI count
- Hidden GCI Count

The main interface shows the 'Ontology IRI' as `http://www.semanticweb.org/roche/ontologies/2025/3/untitled-ontology-2` and the 'Ontology Version IRI' as `e.g. http://www.semanticweb.org/roche/ontologies/2025/3/untitled-ontology-2`. The 'Imported ontologies' section is currently empty.





4. Ontology Building



untitled-ontology-5 (http://www.semanticweb.org/roche/ontologies/2025/3/untitled-ontology-5) : [http://www.semanticweb....

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-5 (http://www.semanticweb.org/roche/ontologies/2025/3/untitled-ontology-5)

Active ontology x Entities x Individuals by class x DL Query x Classes x

Class hierarchy (inferred) Annotations Usage

Class hierarchy Annotations: owl:Thing

Class hierarchy: owl:Thing

Annotations +

Asserted

owl:Thing

Class Hierarchy

```

graph BT
    VoluteKrater[Volute krater] -- rdfs:subClassOf --> Krater[Krater]
    ColumnKrater[Column krater] -- rdfs:subClassOf --> Krater
    BellKrater[Bell krater] -- rdfs:subClassOf --> Krater
    CalyxKrater[Calyx krater] -- rdfs:subClassOf --> Krater
  
```

Instances

Target for Key +

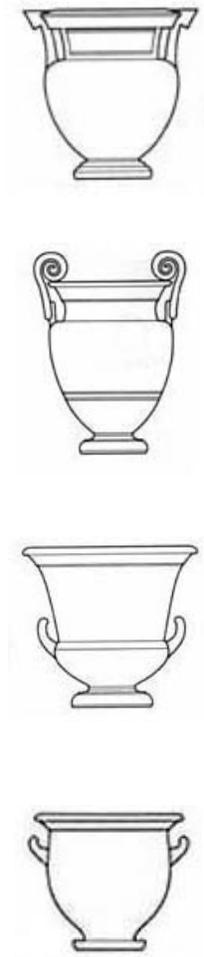
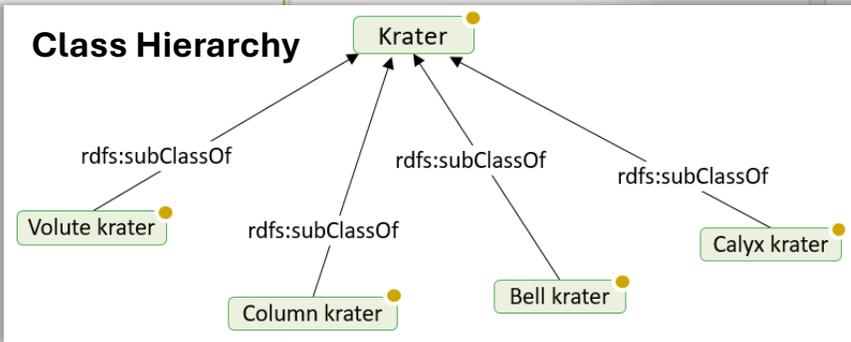
Disjoint With +

No Reasoner set. Select a reasoner from the Reasoner menu Show Inferences





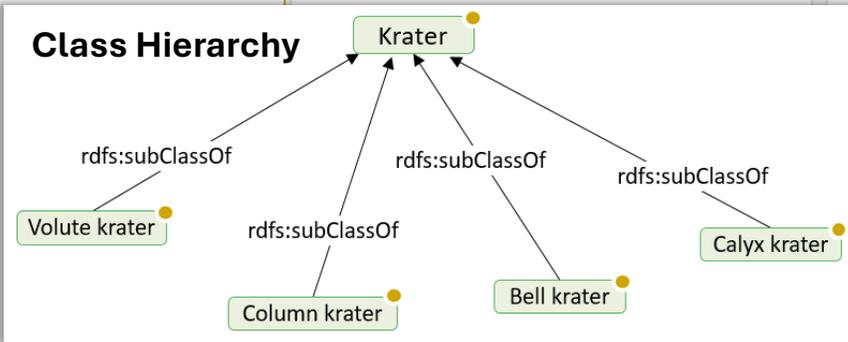
4. Ontology Building

4. Ontology Building: Class Hierarchy



The screenshot shows the Protégé ontology editor interface. On the left, the 'Class hierarchy (inferred)' panel displays a tree structure with 'owl:Thing' as the root and 'Krater' as a child. A red circle highlights the 'Add Class' icon (a plus sign inside a circle) in the toolbar. In the center, a 'Create a new Class' dialog box is open, with the 'Name' field containing 'Bell Krater' and the 'IRI' field containing 'http://www.semanticweb.org/roche/ontologies/2025/3/untitled-ontology-6#Bell_Krater'. The dialog has 'OK' and 'Annuler' buttons. On the right, the 'Annotations' panel shows 'Annotations: Krater' with a plus sign to add more annotations. Below the dialog, the 'General class axioms' panel shows 'SubClass Of (Anonymous Ancestor)'. At the bottom, the status bar indicates 'No Reasoner set. Select a reasoner from the Reasoner menu' and 'Show Inferences' is checked.





4. Ontology Building



Class hierarchy: owl:Thing
 Krater
 Bell_Krater
 Calyx_Krater
 Column_Krater
 Volute_Krater

Annotations: Volute_Krater

Description: Volute_Krater

- Equivalent To
- SubClass Of: Krater
- General class axioms
- SubClass Of (Anonymous Ancestor)
- Instances
- Target for Key
- Disjoint With

Class Hierarchy Diagram:

```
graph BT; Volute_krater[Volute krater] -- rdfs:subClassOf --> Krater; Column_krater[Column krater] -- rdfs:subClassOf --> Krater; Bell_krater[Bell krater] -- rdfs:subClassOf --> Krater; Calyx_krater[Calyx krater] -- rdfs:subClassOf --> Krater;
```





4. Ontology Building



Class hierarchy:

- owl:Thing
 - Krater
 - Bell_Krater
 - Calyx_Krater
 - Column_Krater
 - Volute_Krater

Annotations: Volute_Krater

Annotations +

Description: Volute_Krater

Equivalent To +

SubClass Of +

- Krater

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

Class Hierarchy

```

graph BT
    VoluteKrater[Volute krater] -- rdfs:subClassOf --> Krater[Krater]
    ColumnKrater[Column krater] -- rdfs:subClassOf --> Krater
    BellKrater[Bell krater] -- rdfs:subClassOf --> Krater
    CalyxKrater[Calyx krater] -- rdfs:subClassOf --> Krater
    
```

Definition?





4. Ontology Building



Class hierarchy: Volute_Krater Annotations: Volute_Krater

owl:Thing
└─ Krater
 └─ Bell_Krater
 └─ Calyx_Krater
 └─ Column_Krater
 └─ Volute_Krater

Annotations

Description: Volute_Krater

SubClass Of +
● Krater

General class axiom

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Definition?

Disjoint With +
● Calyx_Krater, Column_Krater, Bell_Krater

Disjoint Union Of +

Class Hierarchy

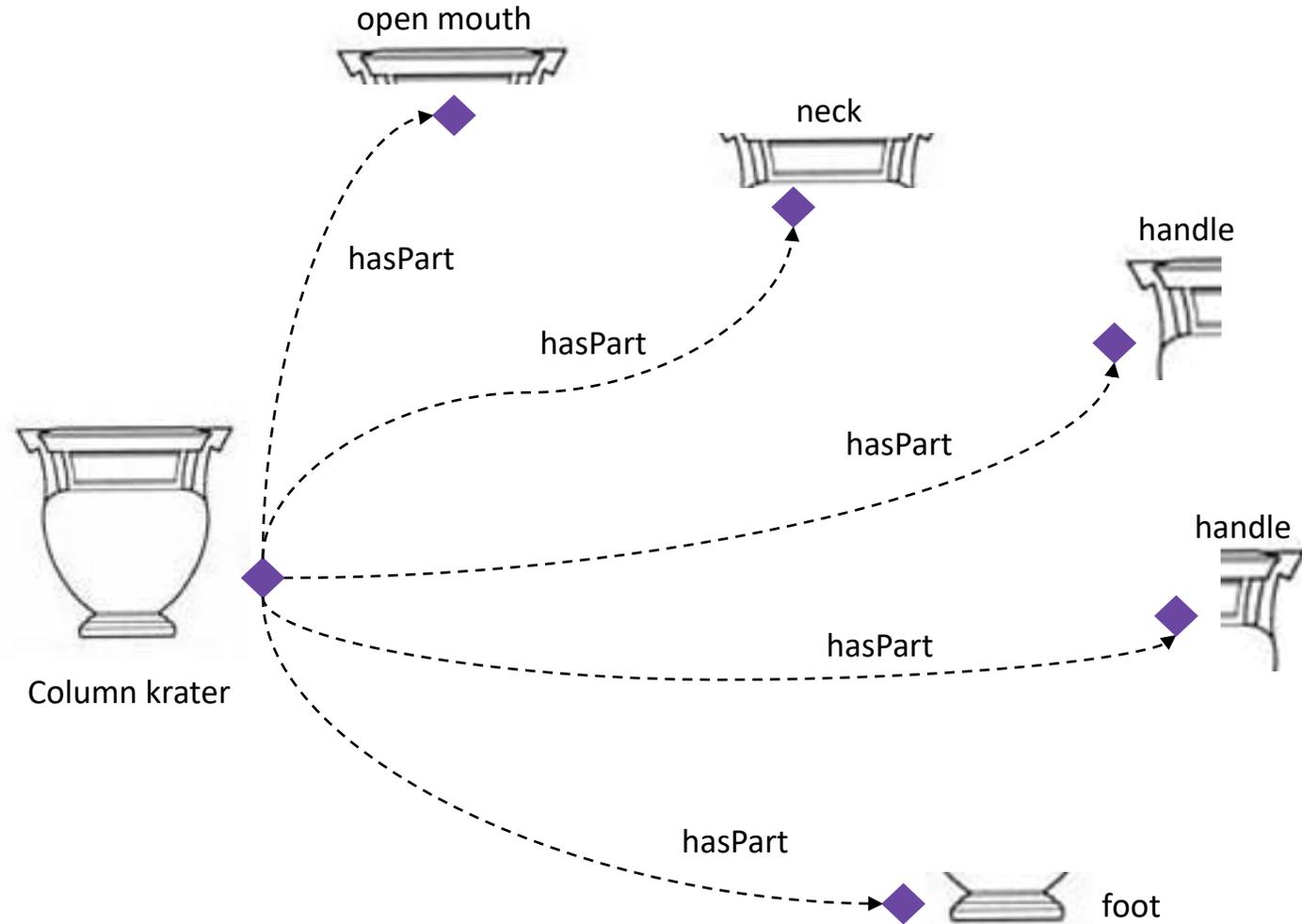
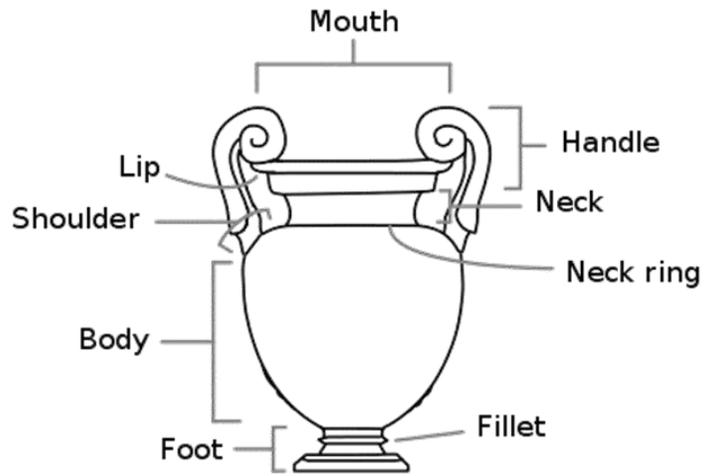
```
graph BT; Volute_krater[Volute krater] -- rdfs:subClassOf --> Krater; Column_krater[Column krater] -- rdfs:subClassOf --> Krater; Bell_krater[Bell krater] -- rdfs:subClassOf --> Krater; Calyx_krater[Calyx krater] -- rdfs:subClassOf --> Krater;
```



Object Properties

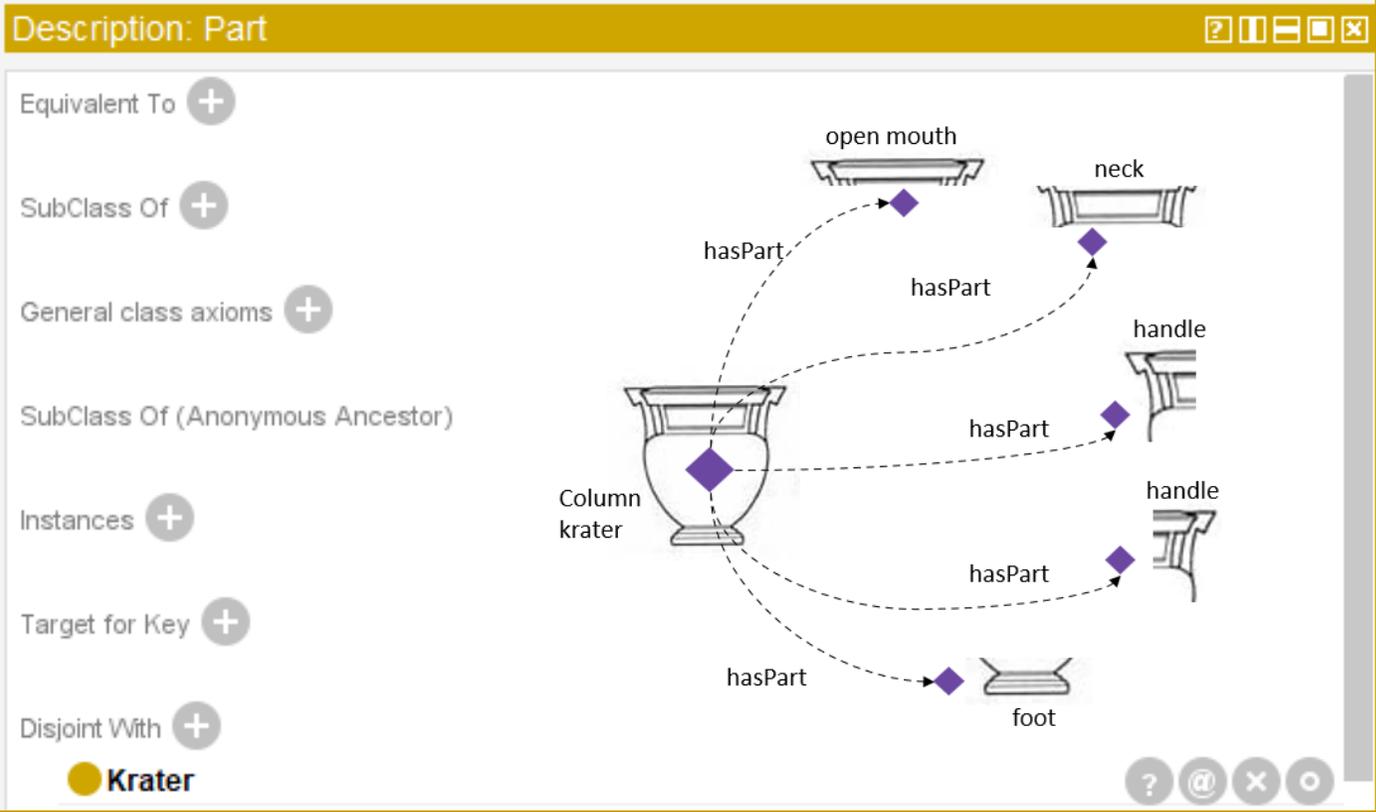
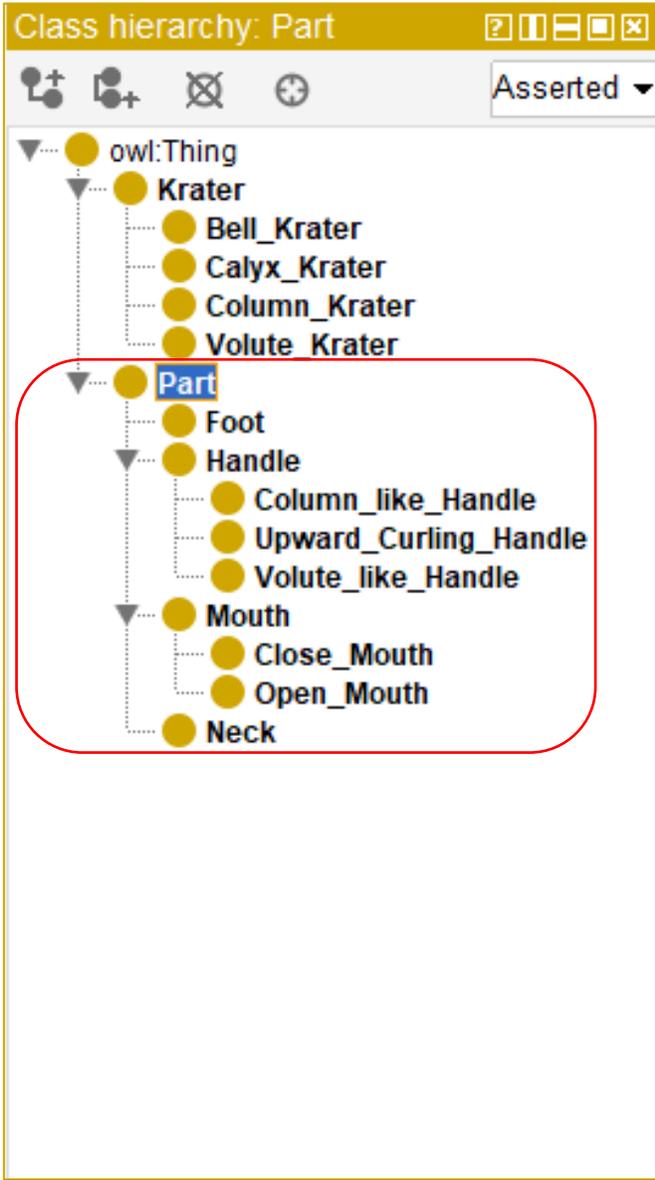


Relationships between individuals

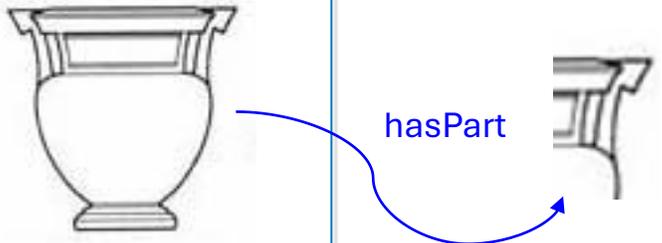




4. Ontology Building



4. Ontology Building



Object property hierarchy: hasPart

- owl:topObjectProperty
 - hasPart

Annotations: hasPart

Annotations +

Description: hasPart

- Functional
- Inverse function
- Transitive
- Symmetric
- Asymmetric
- Reflexive
- Irreflexive

Equivalent To +

SubProperty Of +

Inverse Of +

Domains (intersection) +

- Krater

Ranges (intersection) +

- Part



Definition of Krater Classes: Property Restrictions



Class hierarchy: Column_krat Annotations: Column_krater

- owl:Thing
 - Greek_God
 - Krater
 - Bell_krater
 - Calyx_krater
 - Column_krater
 - Volute_krater
 - Part
 - Foot
 - Handle
 - Column_like_handle
 - Upward_curling_handle
 - Volute_like_handle
 - Lip
 - Mouth
 - Neck

Annotations:

- rdfs:label [language: en] column krater
- rdfs:label [language: fr] cratère à colonnettes

Description: Column_krater

SubClass Of **+** Krater Add

General class axioms **+**

SubClass Of (Anonymous Ancestor)

Column_krater

Data restriction creator Class expression editor Object restriction creator Class hierarchy

Restricted property

- owl:topObjectProperty
 - figured
 - hasPart

Restriction filler

- owl:Thing
 - Greek_God
 - Krater
 - Part
 - Foot
 - Handle
 - Column_like_handle
 - Upward_curling_handle
 - Volute_like_handle
 - Lip
 - Mouth
 - Neck

Restriction type

Exactly (exact cardinality) Cardinality 2

OK Annuler



Definition of Krater Classes: Property Restrictions



Class hierarchy: Column_krat

- owl:Thing
 - Greek_God
 - Krater
 - Bell_krater
 - Calyx_krater
 - Column_krater
 - Volute_krater
 - Part
 - Foot
 - Handle
 - Column_like_handle
 - Upward_curling_handle
 - Volute_like_handle
 - Lip
 - Mouth
 - Neck

Annotations: Column_krater

- rdfs:label [language: en] column krater
- rdfs:label [language: fr] cratère à colonnettes

Description: Column_krater

- SubClass Of
 - hasPart exactly 2 Column_like_handle
 - Krater
- General class axioms
- SubClass Of (Anonymous Ancestor)



Class hierarchy: Volute_krate

- owl:Thing
 - Greek_God
 - Krater
 - Bell_krater
 - Calyx_krater
 - Column_krater
 - Volute_krater
 - Part
 - Foot
 - Handle
 - Column_like_handle
 - Upward_curling_handle
 - Volute_like_handle
 - Lip
 - Mouth
 - Neck

Annotations: Volute_krater

- rdfs:label [language: en] volute krater
- skos:definition [language: fr] Krater with a clearly defined neck and volute-like hand

Description: Volute_krater

- Equivalent To
- SubClass Of
 - hasPart exactly 2 Volute_like_handle
 - Krater
- General class axioms
- SubClass Of (Anonymous Ancestor)





4. Ontology Building: Annotating

The screenshot shows the Protégé ontology editor interface. On the left, a class hierarchy tree shows 'owl:Thing' as the root, with 'Krater' as a subclass, and 'Bell_krater', 'Calyx_krater', 'Column_krater', and 'Volute_krater' as subclasses of 'Krater'. The 'Bell_krater' class is selected and highlighted in blue.

The main window is divided into several panes:

- Annotations: Bell_krater**: A pane showing the list of annotations for the selected class. A red arrow points to a green '+' icon, indicating the option to add a new annotation.
- Description: Bell_krater**: A pane showing the class's description and relationships. It lists 'Equivalent To', 'SubClass Of' (with 'Krater' selected), 'General class axioms', 'SubClass Of (Anonymous Ancestor)', 'Instances', 'Target for Key', and 'Disjoint With' (with 'Volute_krater, Calyx_krater, Column_krater' listed).
- Bell_krater**: A dialog box for adding a new annotation. It shows a list of available properties, with 'rdfs:label' selected and highlighted in blue. A red arrow points to this selection. The dialog also shows the 'Literal' tab with the text 'bell krater' entered, and the 'Lang' dropdown set to 'en'. Buttons for 'OK' and 'Annuler' are visible at the bottom.

Annotations are represented as triples: `"bell krater"@en` and `"cratère en cloche"@fr`. A magnifying glass icon is positioned over the 'Bell krater' text, with a dashed blue arrow pointing to the 'rdfs:label' property name in the dialog box.



4. Ontology Building: Annotating

Class hierarchy: Asserted

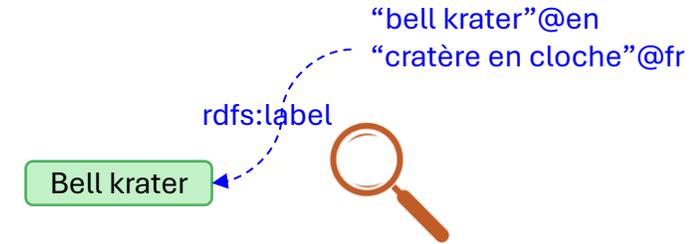
- owl:Thing
 - Krater
 - Bell_Krater**
 - Calyx_Krater
 - Column_Krater
 - Volute_Krater
 - Part

Annotations: Bell_Krater

- Annotations +
- rdfs:label** [language: fr] cratère en cloche
- rdfs:label** [language: en] bell krater

Description: Bell_Krater

- Equivalent To +
- SubClass Of +
 - Krater**
- General class axioms +
- SubClass Of (Anonymous Ancestor)
- Instances +



4. Ontology Building: Annotating

Class hierarchy: Bell_Krater

- owl:Thing
 - Krater
 - Bell_Krater**
 - Calyx_Krater
 - Column_Krater
 - Volute_Krater
 - Part

Annotations: Bell_Krater

Annotations +

- rdfs:label** [language: fr] cratère à cloche
- rdfs:label** [language: en] bell krater
- skos:definition** [language: en] It is named for its bell-like shape, perhaps originating in wood. It has small horizontal upturned handles just over halfway up the body.
- rdfs:seeAlso** <https://www.carc.ox.ac.uk/carc/resources/Introduction-to-Greek-Pottery/Shapes/Kraters>

Description: Bell_Krater

SubClass Of +

- Krater

General class axioms +

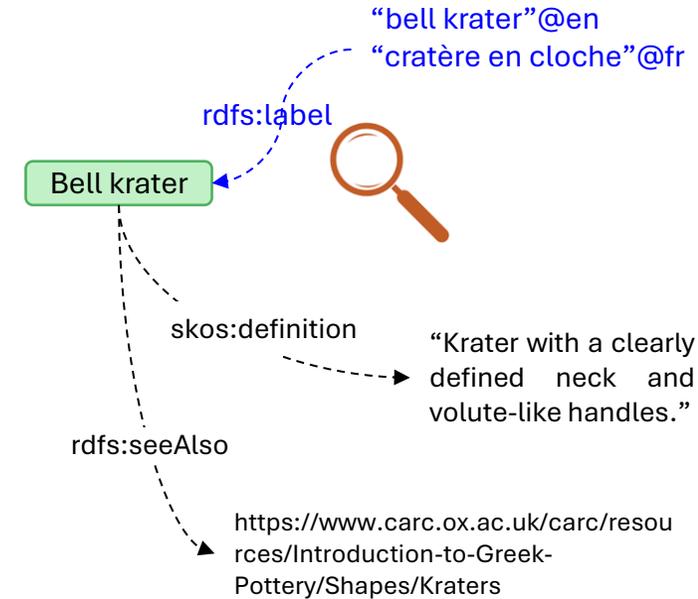
SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

- Calyx_Krater, Column_Krater, Volute_Krater



4. Ontology Building: Populating

The screenshot shows the Protégé ontology editor with the following components:

- Class Hierarchy:**
 - owl:Thing
 - Krater
 - Bell_krater
 - Calyx_krater** (selected)
 - Column_krater
 - Volute_krater

- Annotations for Beazley-215424:**
- rdfs:label: Beazley-215424
- rdfs:seeAlso: <http://ark.dasch.swiss/ark:/72163/080e-768258017ac42-7>
- rdfs:seeAlso: <http://www.beazley.ox.ac.uk/record/87F1A62D-DD35-4BA9-A5CB-E4E296788FB8>
- foaf:depiction:
- Property Assertions:**
- technique "Red-Figure"@en
- fabric "Athenian"@en


Vase Number: 215424
Fabric: ATHENIAN
Technique: RED-FIGURE
Shape Name: KRATER, CALYX
Date: -450 to -400

5. Open Questions

Definition: { essential characteristics }

Amphora



for storing and transport



without neck

Bell_Krater ::= { *for_mixing_wine_and_water* , *without_neck* ,
with_foot , *with_open_mouth* ,
with_upward_curling_handles }

Krater

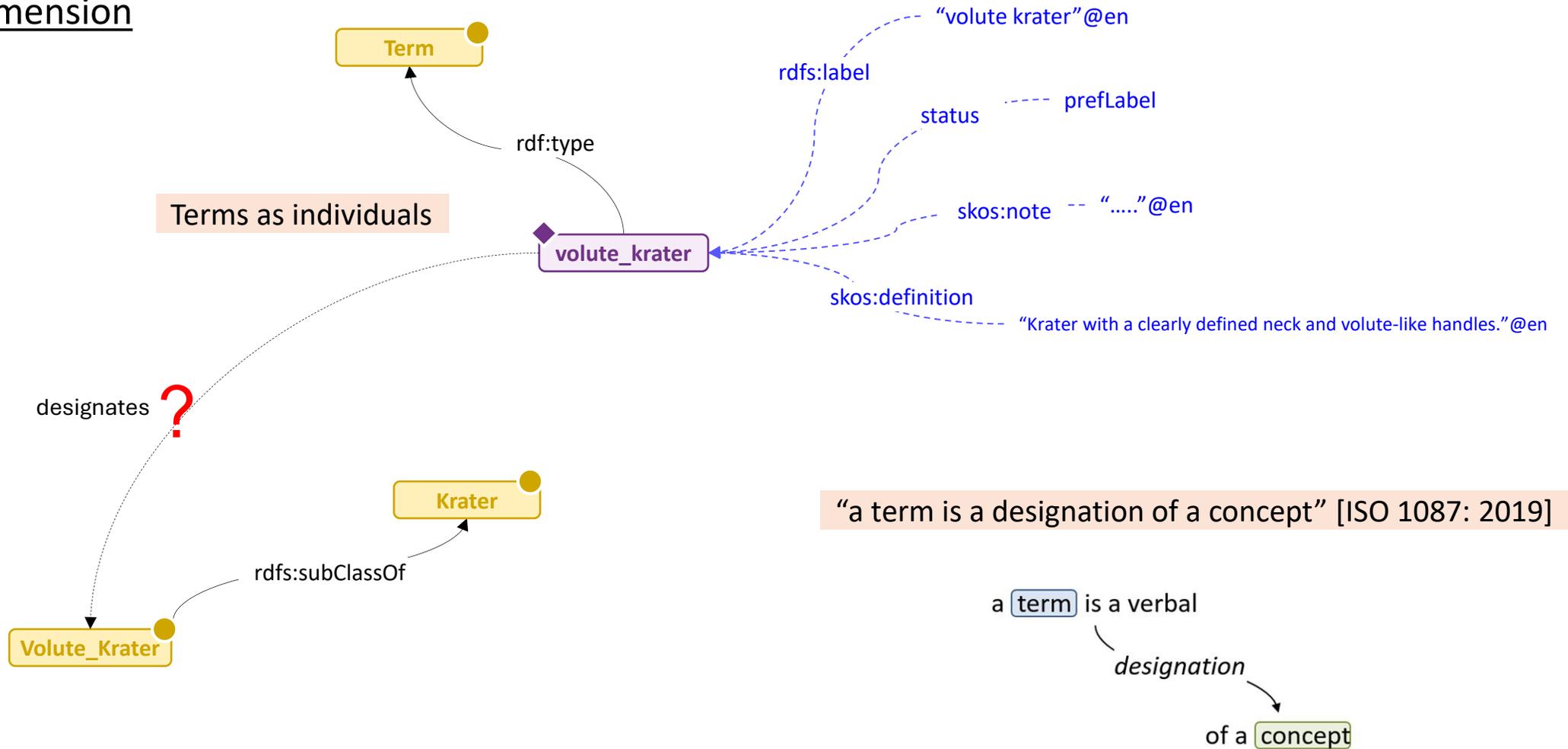


for mixing wine and water



5. Open Questions

Linguistic Dimension



6. Conclusion

- ✓ Free environment
- ✓ Large Community
- ✓ Definition based on relations between objects (Class)
- ✓ Description Logic Reasoners
- ✓ W3C Standards compliant



- How to represent essential characteristics (Concept)?
- How to represent the linguistic dimension?
- How to take into account the way of thinking of Experts?