

# The knowledge adaptive toolbox: a practical toolkit for the XXI century ontologist/knowledge engineer

Knowledge Graphs and Semantic Technologies

Stefano De Giorgis



## **Stefano De Giorgis**

Bachelor Degree in Italian Literature at University of Torino  
Master Degree in Linguistic Sciences at University of Bologna



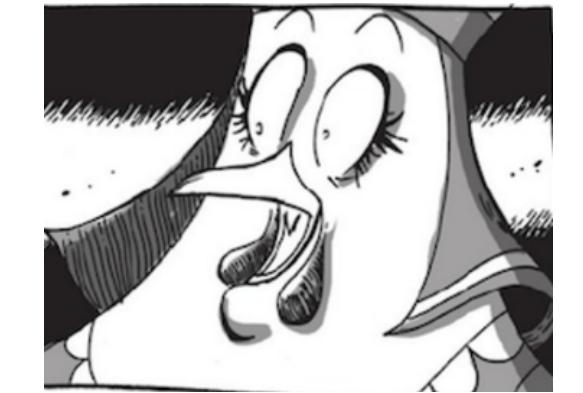


## **Stefano De Giorgis**

Bachelor Degree in Italian Literature at University of Torino  
Master Degree in Linguistic Sciences at University of Bologna



PhD in Digital Humanities at University of Bologna with prof. Aldo Gangemi





## Stefano De Giorgis

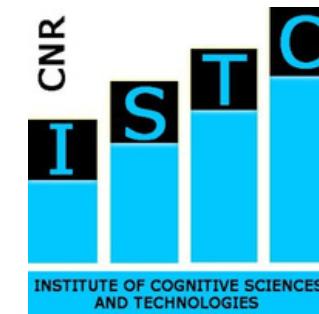
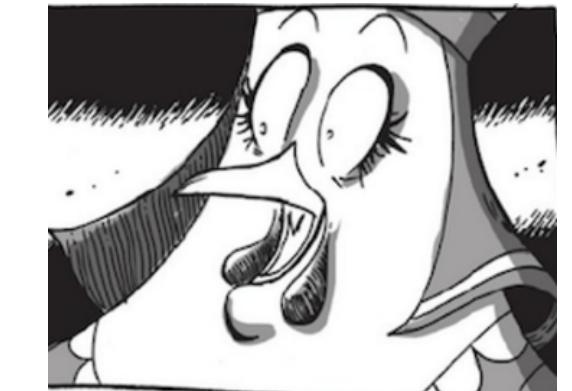
Bachelor Degree in Italian Literature at University of Torino  
Master Degree in Linguistic Sciences at University of Bologna



PhD in Digital Humanities at University of Bologna with prof. Aldo Gangemi

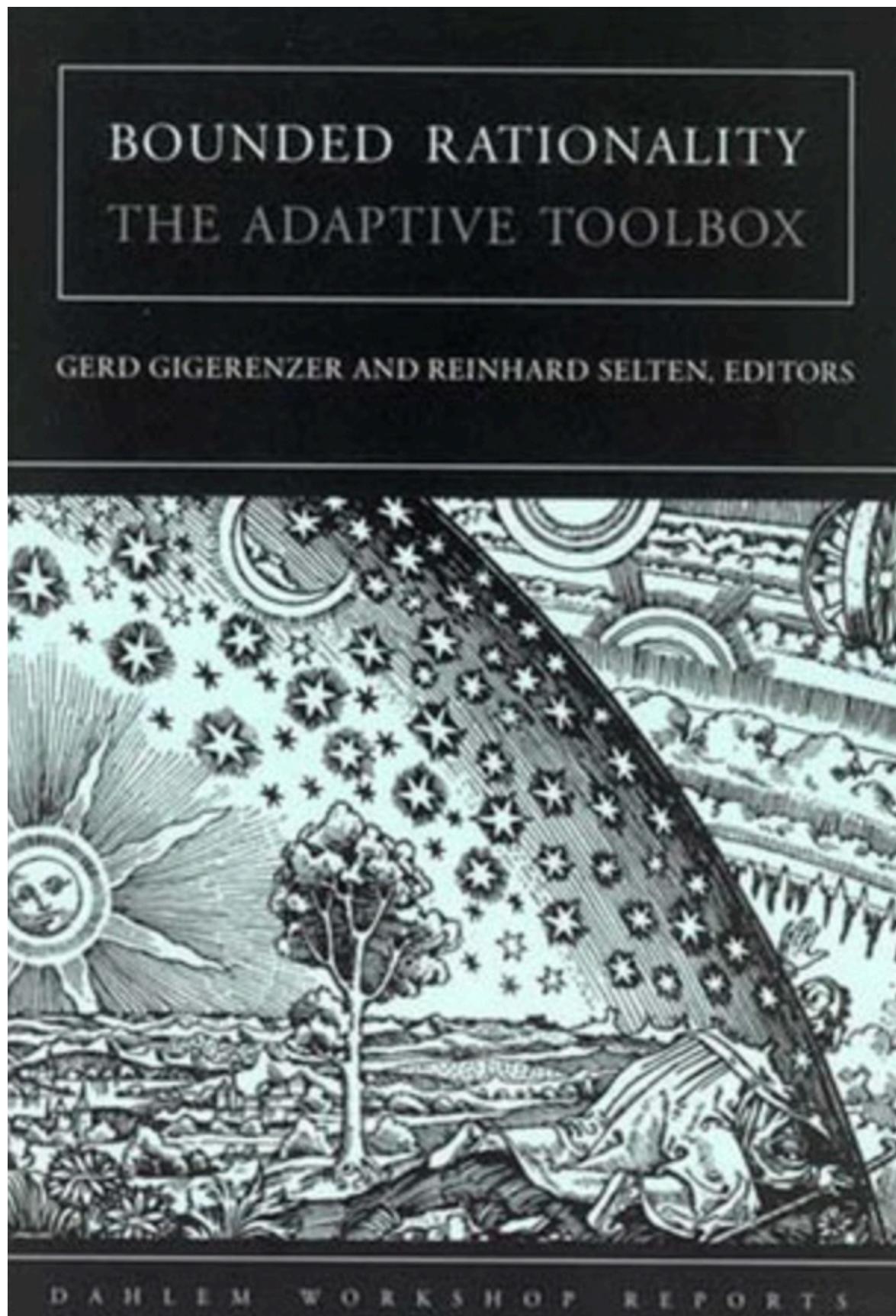


Researcher at the Institute for Cognitive Sciences and Technologies - National Research Council (ISTC-CNR) on "Bio-Socio-Cognitive AI for Grounded World Models"

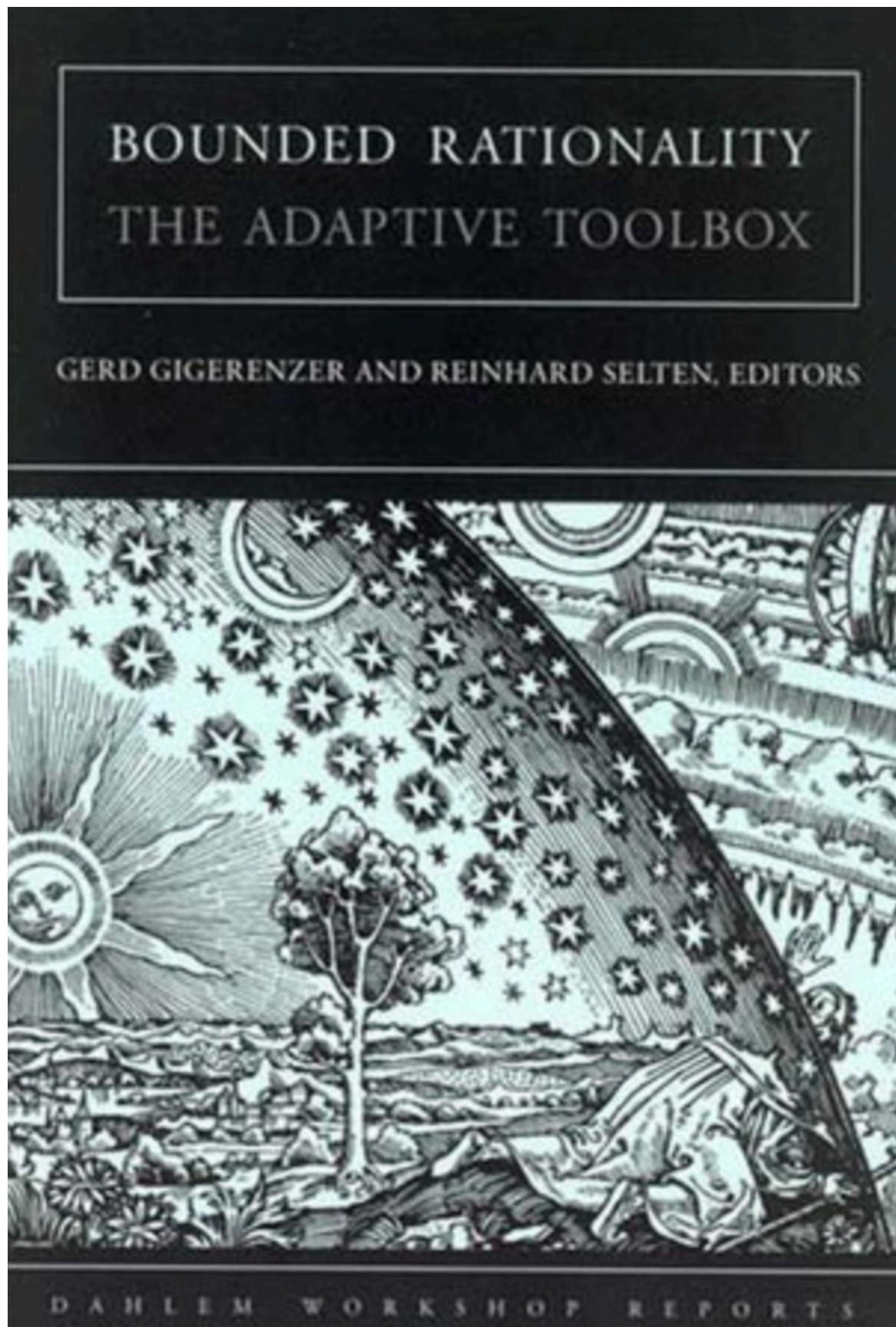


Post-doc at Vrije Universiteit Amsterdam with prof. Filip Ilievski and prof. Frank van Harmelen on the "Interpretation of Internet Memes"





# The knowledge adaptive toolbox: a practical toolkit for the XXI century ontologist/knowledge engineer



# The knowledge adaptive toolbox: a practical toolkit for the XXI century ontologist/knowledge engineer

Ontologies to the rescue!

# Toolkit

## Ontology Modeling Approaches

- Modeling Methodologies
- Foundational Ontologies

## Ontology / Graph visualisation & management

- Protégé
- PoolParty Semantic Suite
- TopBraid Composer/EDG
- Metaphactory
- Neo4j suite

## Query

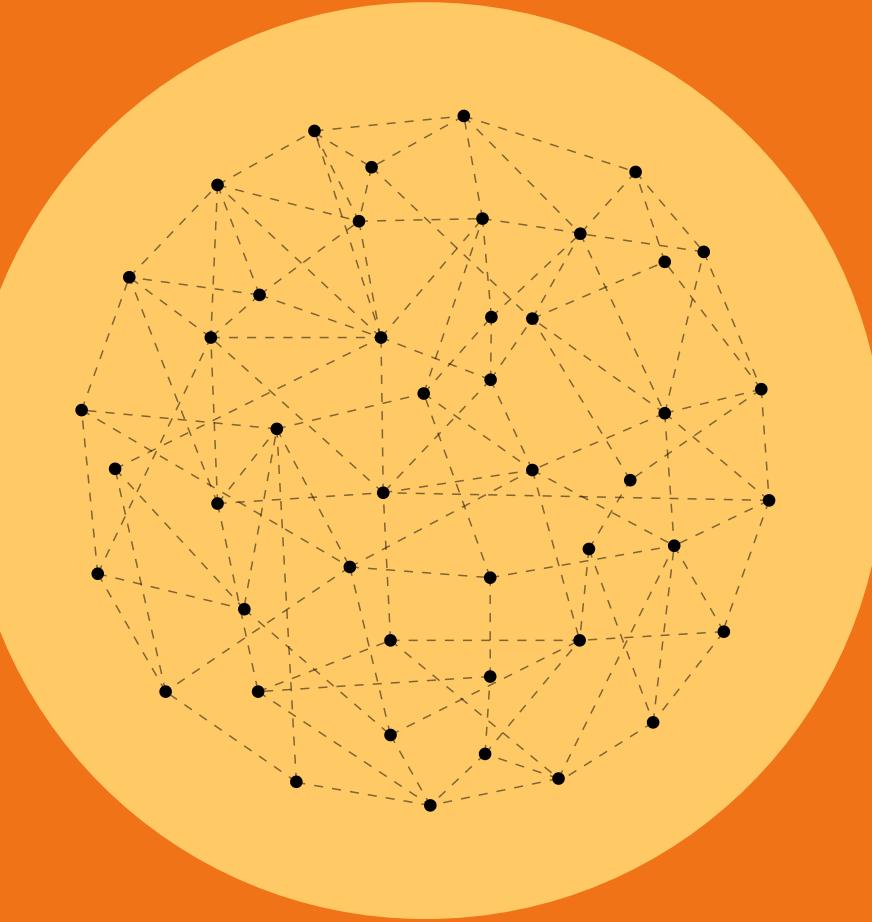
- SPARQL endpoint
  - Apache Jena Fuseki
  - Qlever
  - Blazegraph
  - GraphDB

## Conceptual Modeling, Visual Languages, and Diagrammatic Tools

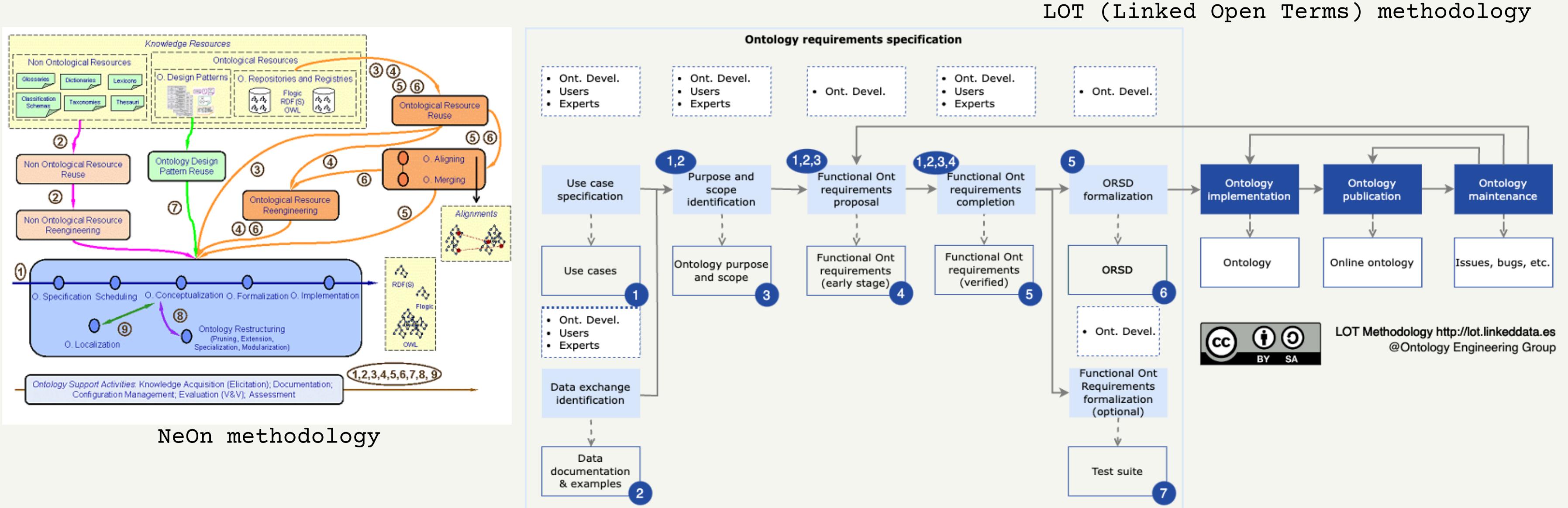
- OntoUML
- ChOWLk
- Graffoo (extension on diagrams.net)

# Modeling Approaches and Techniques

- Ontology 101
- NeOn methodology
- SAMOD
- XD - eXtreme Design
- MoMo - Modular Modeling



# Ontology design approaches - 1



## Ontology Development 101: A Guide to Creating Your First Ontology

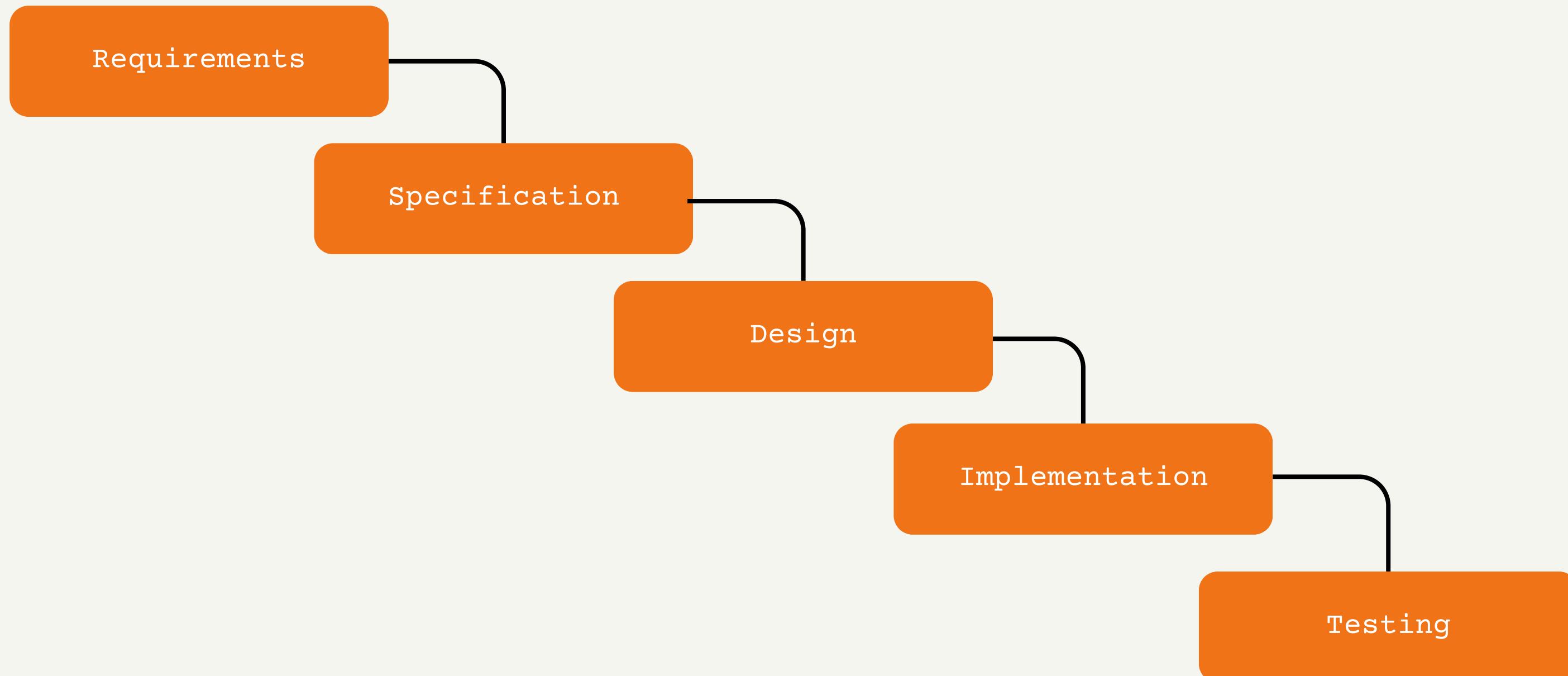
Natalya F. Noy and Deborah L. McGuinness

Stanford University, Stanford, CA, 94305

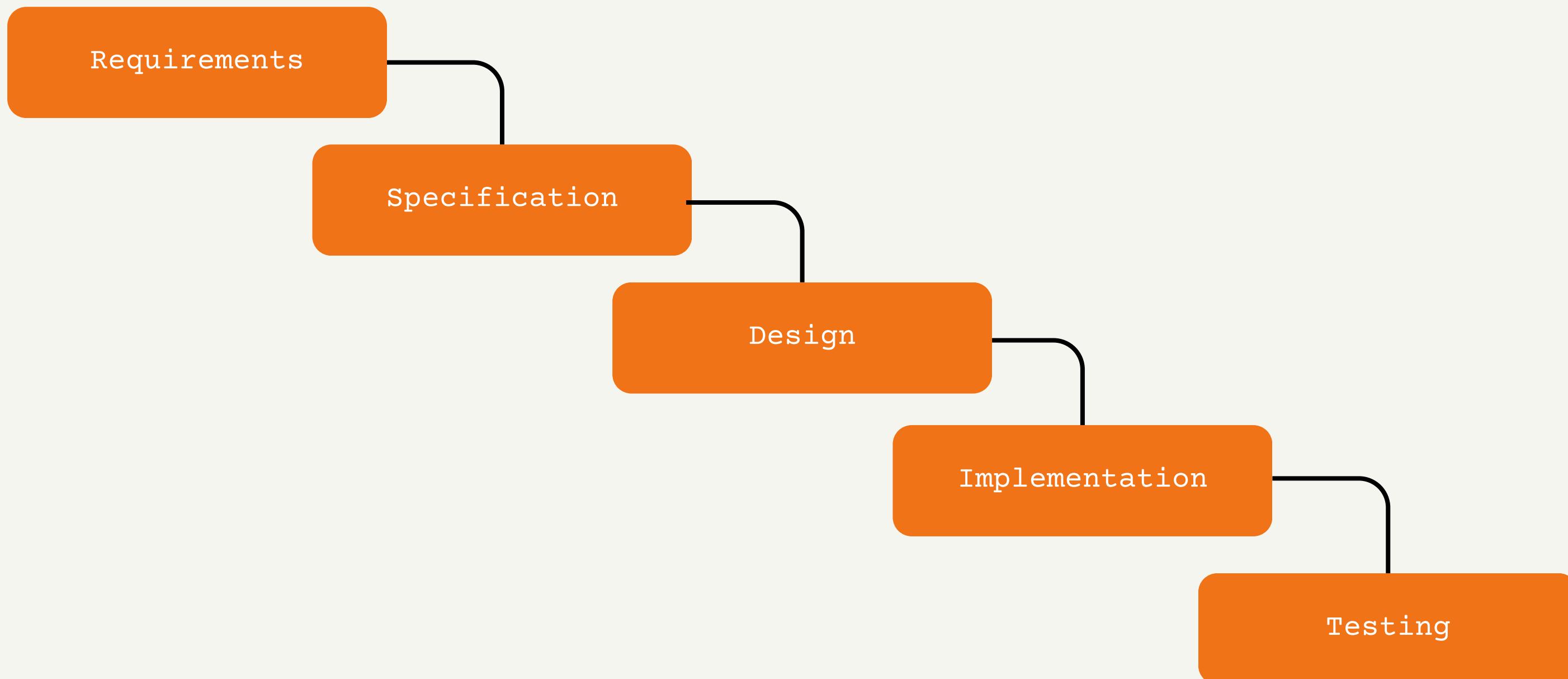
[noy@smi.stanford.edu](mailto:noy@smi.stanford.edu) and [dlm@ksl.stanford.edu](mailto:dlm@ksl.stanford.edu)

Ontology\_101

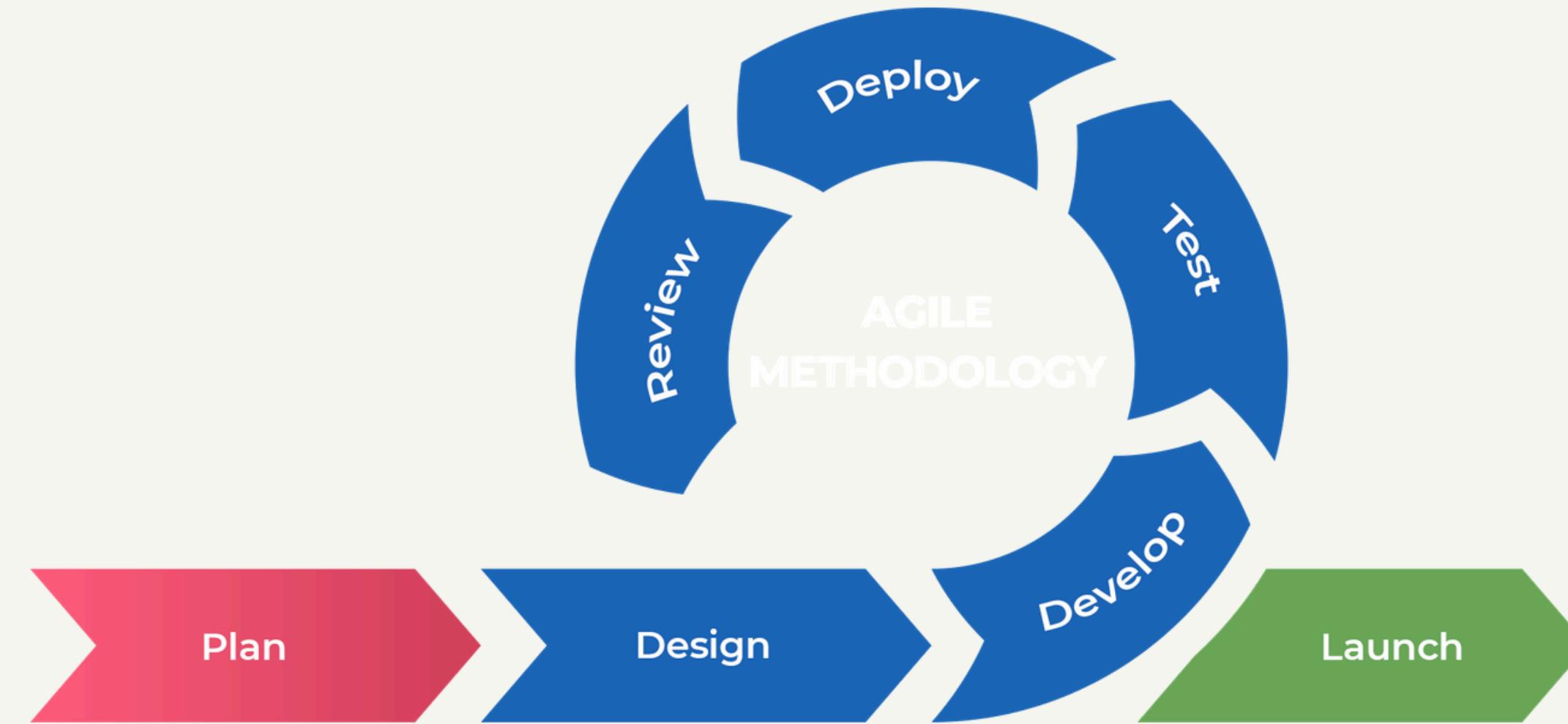
# NeOn Methodology



# Waterfall model for Software Engineering (1970)



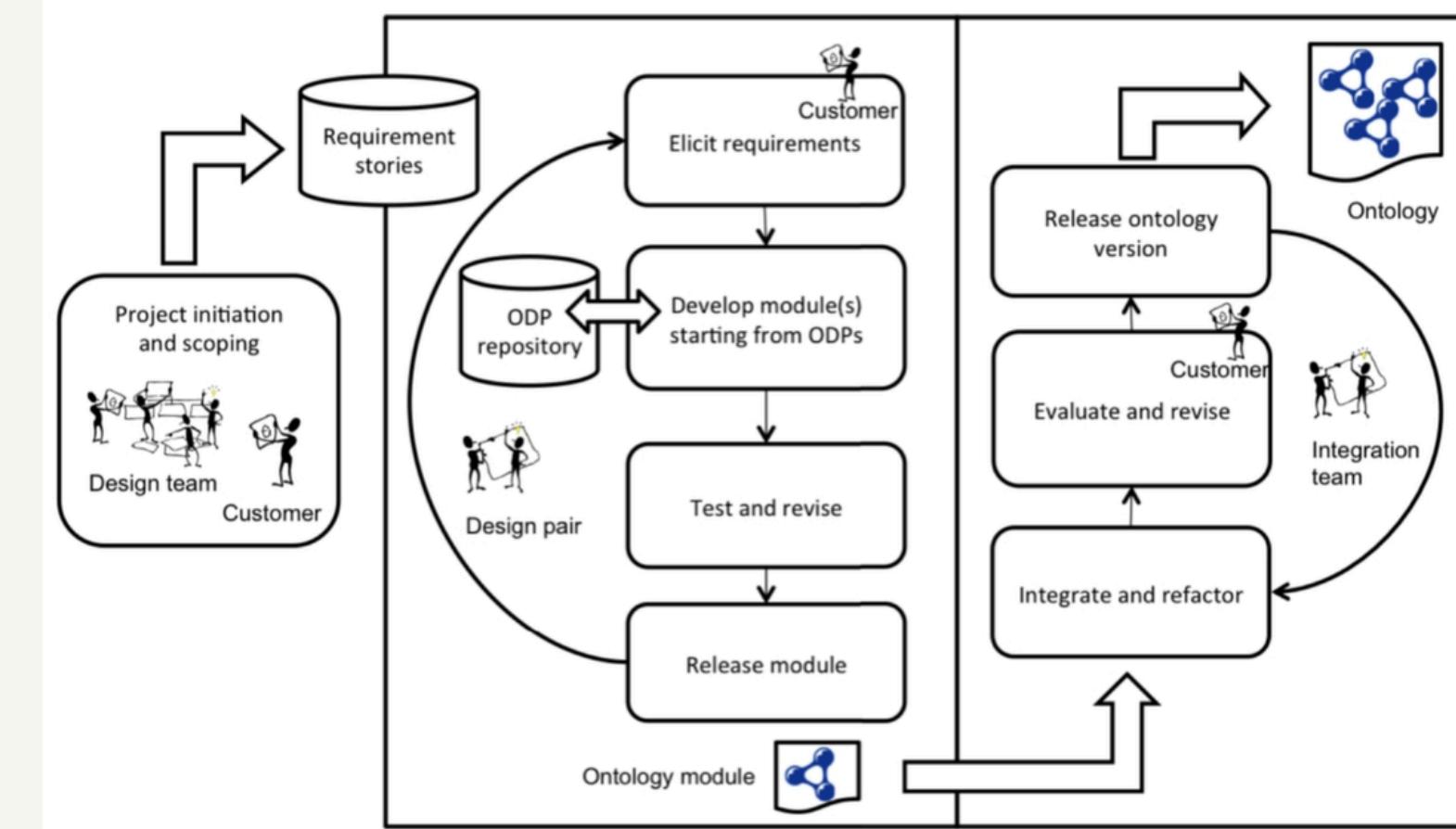
# Agile Software Development (2001)



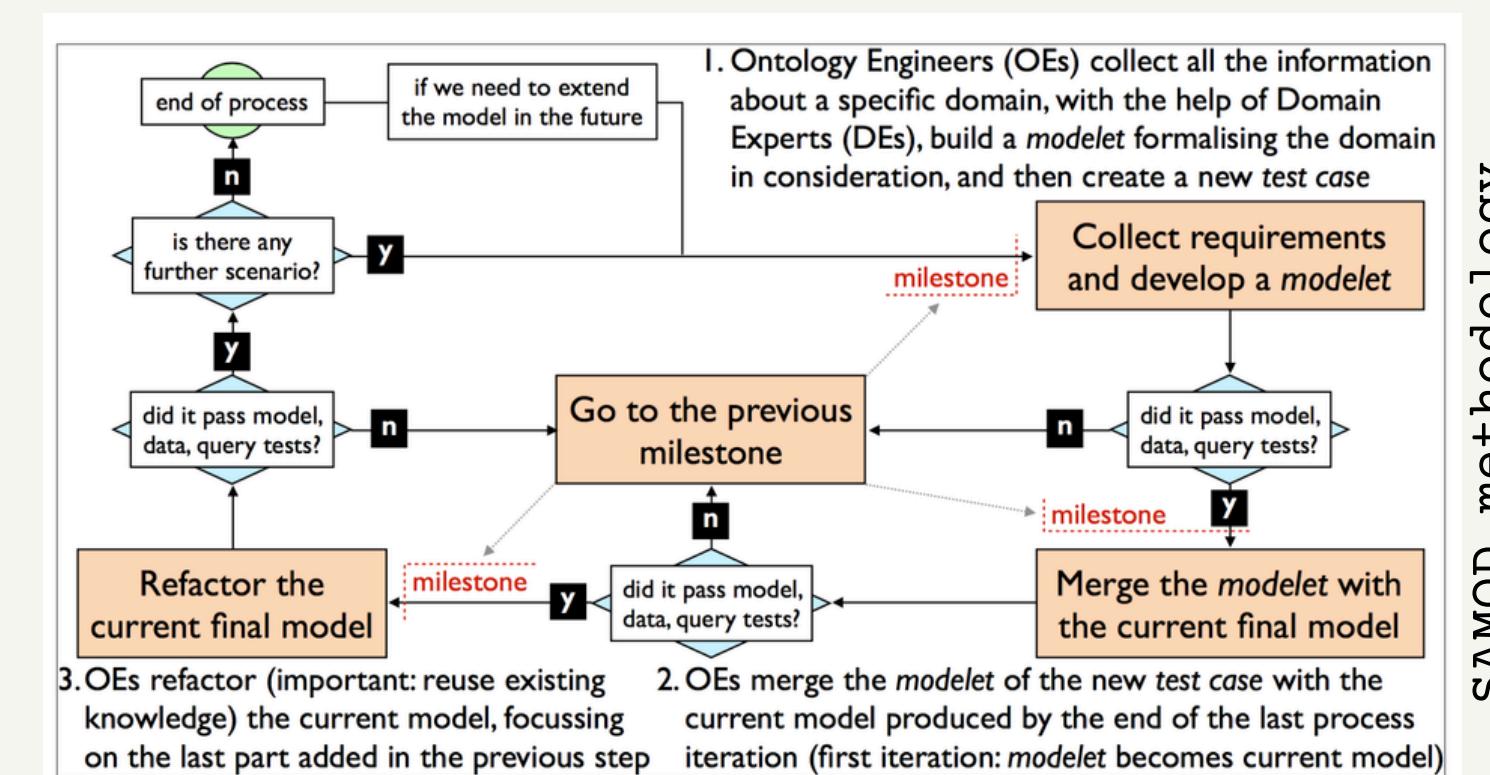
# Ontology design approaches - 2

Step	Responsible	Output
1. Describe use cases & data sources	Entire team	Use case descriptions
2. Gather competency questions	Entire team	List of CQs
3. Identify key notions	Entire team	List of key notions
4. Identify existing ODPs	Ontology engineers	Selected ODP(s) for each key notion.
5. Create module diagrams	Entire team	Diagrammatic representation of the solution module.
6. Document modules & axioms	Ontology engineers & domain experts	Module documentation with embedded schema diagrams, axiomatization, etc. (e.g., in LaTeX, Word, HTML format).
7. Create ontology diagram	Ontology engineers	Diagrammatic representation of the whole composed ontology.
8. Add spanning axioms	Ontology engineers	Documentation of the entire ontology with embedded schema diagrams, axiomatization, etc. (e.g., in LaTeX, Word, HTML format).
9. Review naming & axioms	Ontology engineers	Updated module and ontology documentation.
10. Create OWL file & axioms	Ontology engineers	An OWL file for publication and use.

MOMo workflow

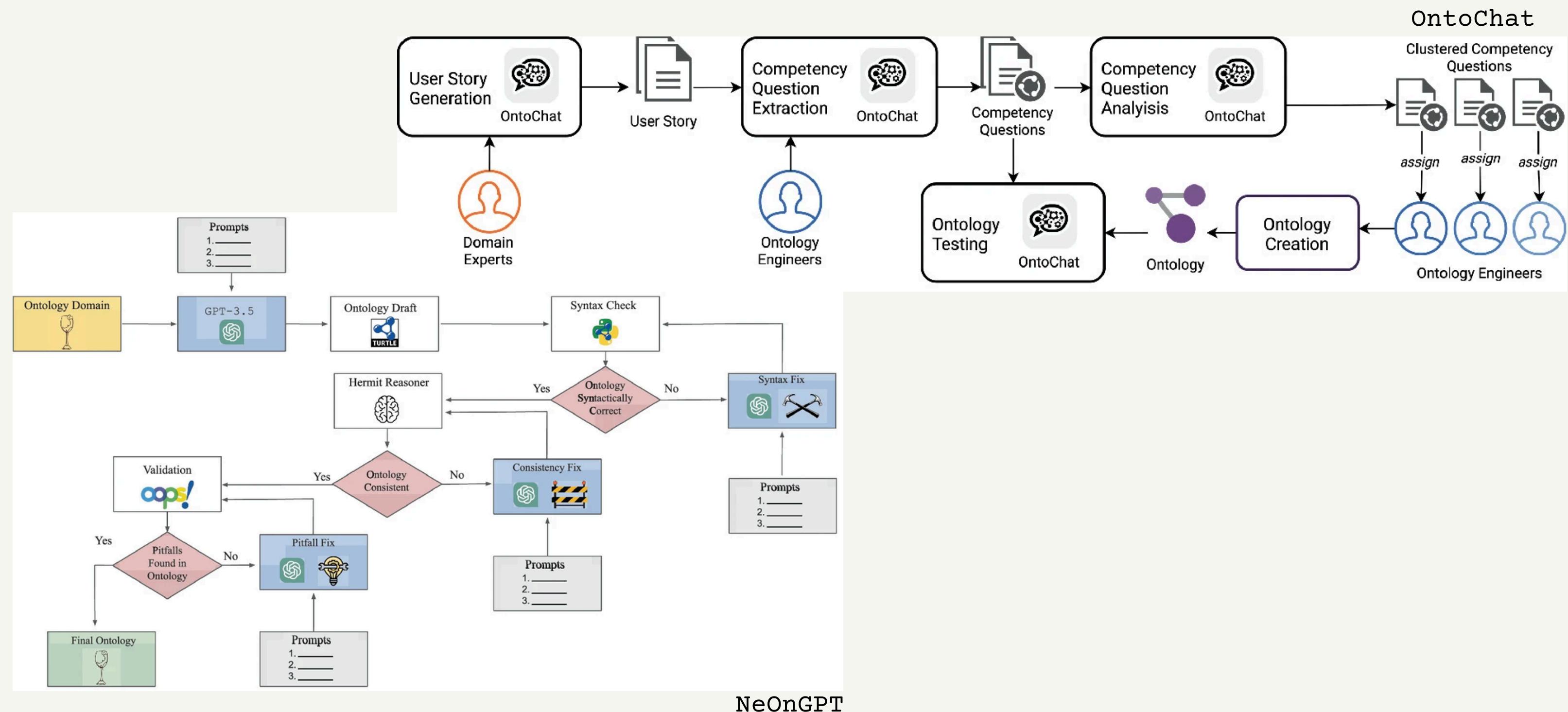


extreme Design



SAMOD methodology

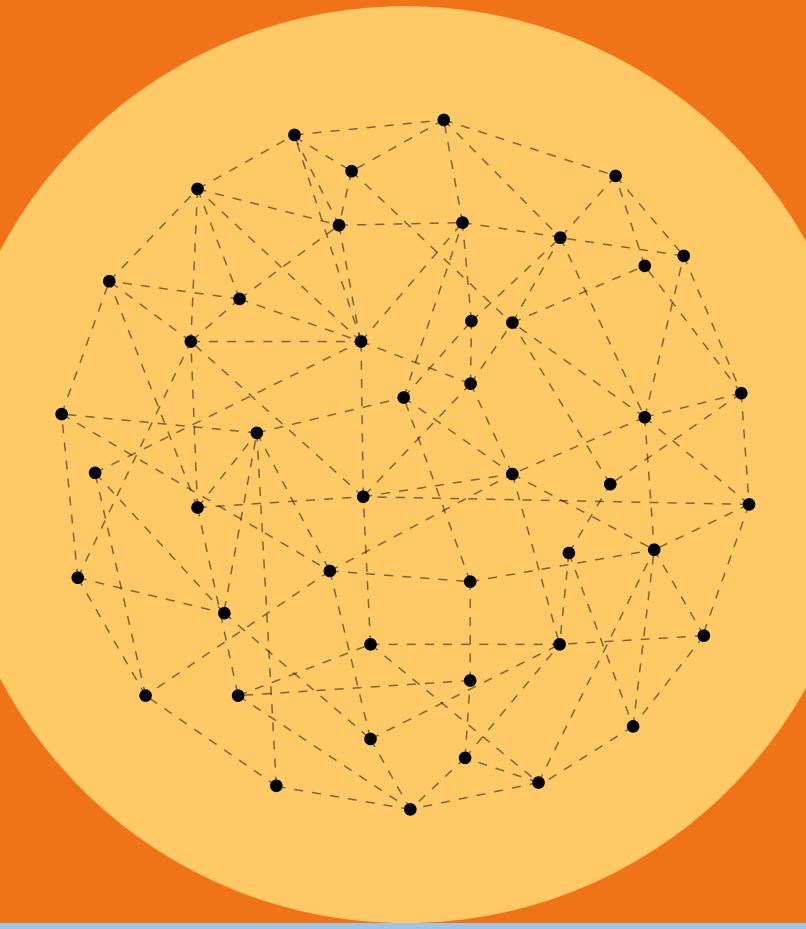
# LLM-based ontology methodologies



NeOnGPT

# Modeling types and challenges

---



TOP LEVEL

**Foundational  
Mereological**

commitment towards conflicting foundational theories  
commitment towards conflicting mereological theories

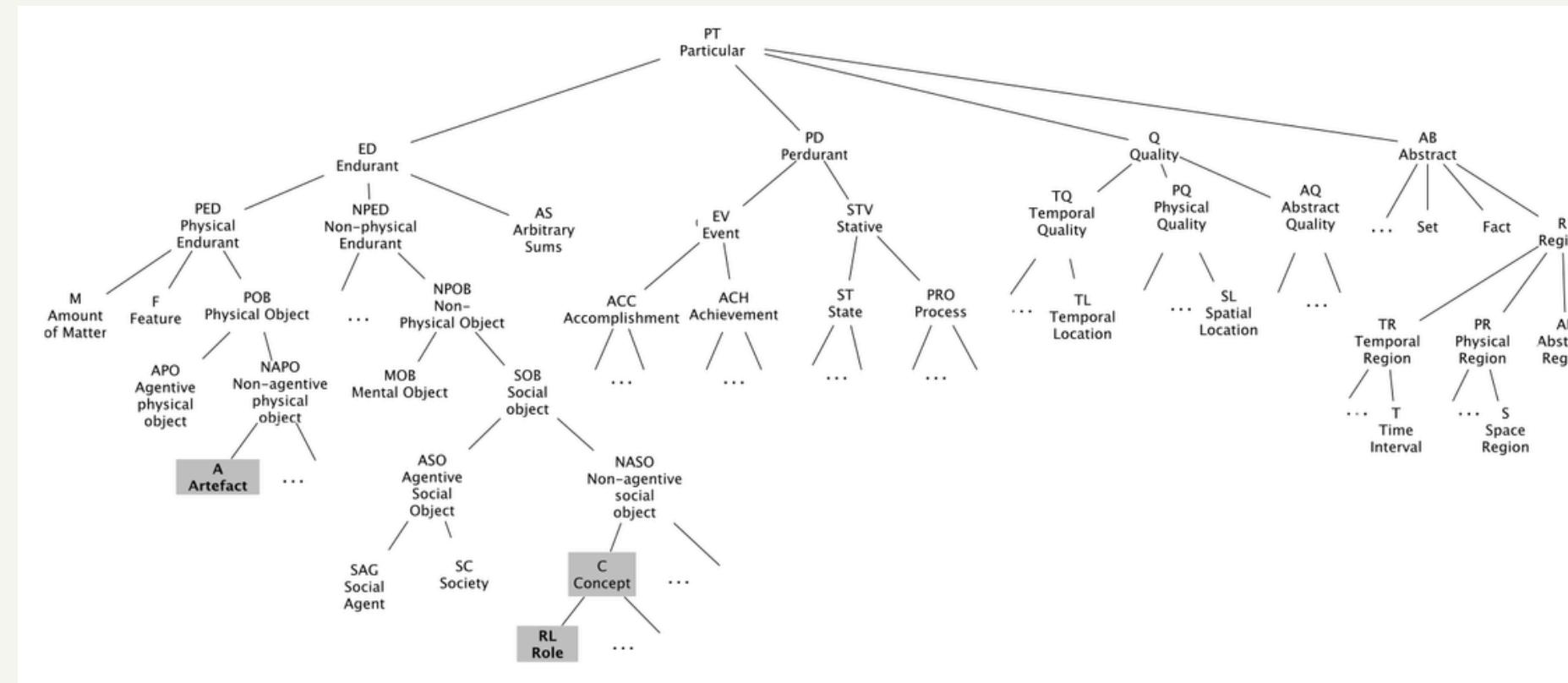
**SEMANTICS  
NEGOTIATION  
ISSUES**



TOP LEVEL

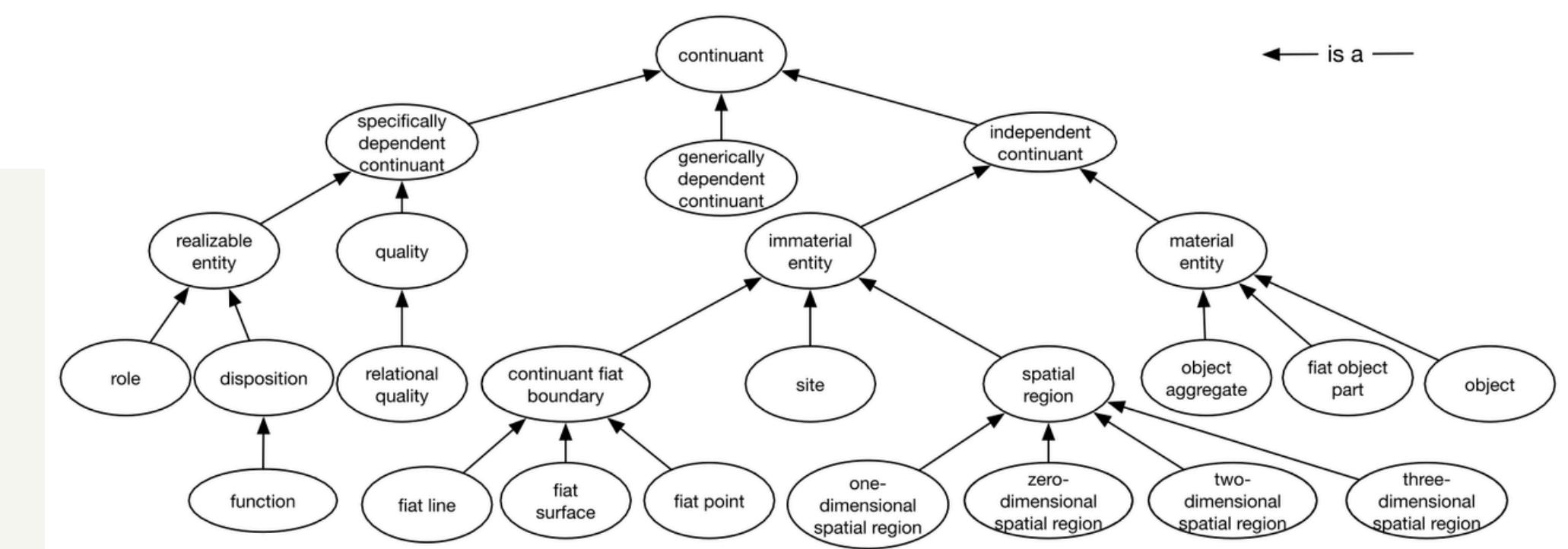
## Foundational Mereological

commitment towards conflicting foundational theories  
commitment towards conflicting mereological theories



BFO - Basic Formal Ontology

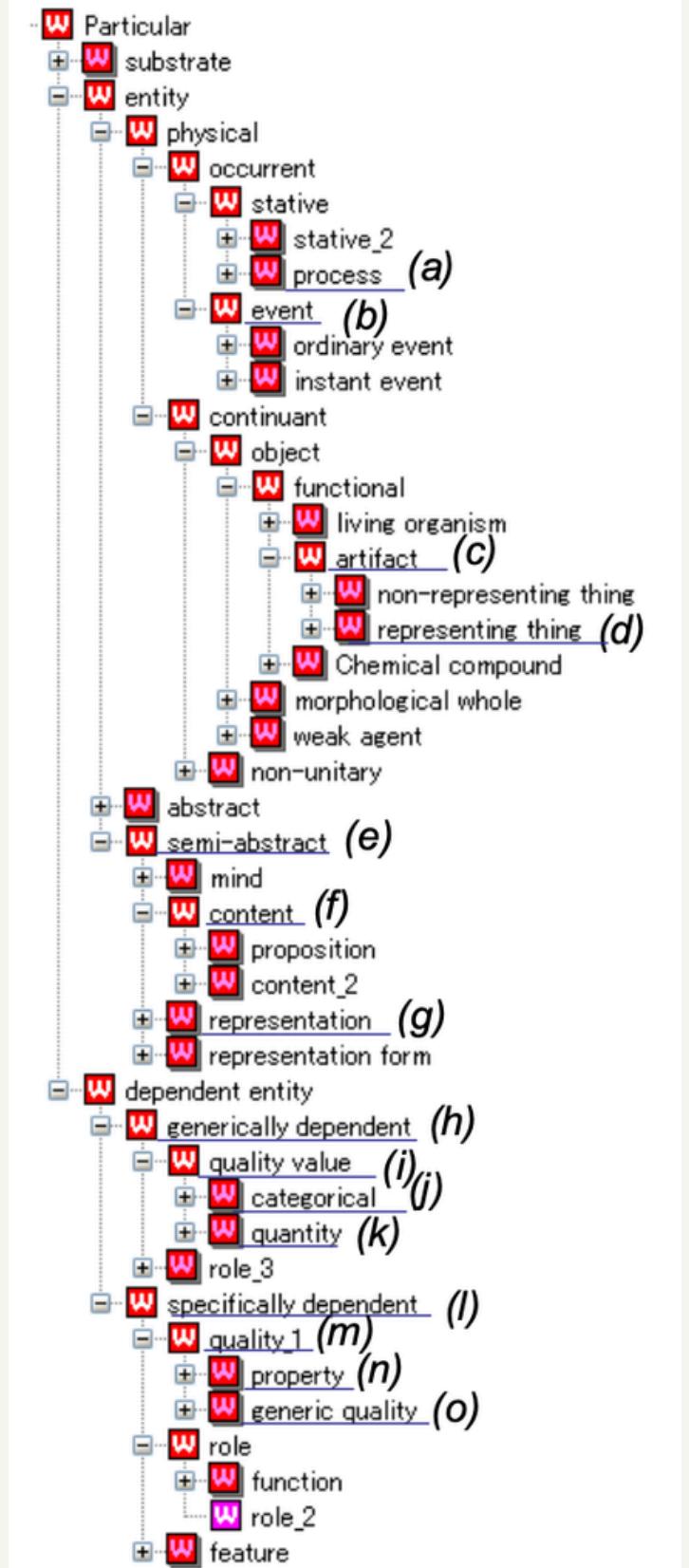
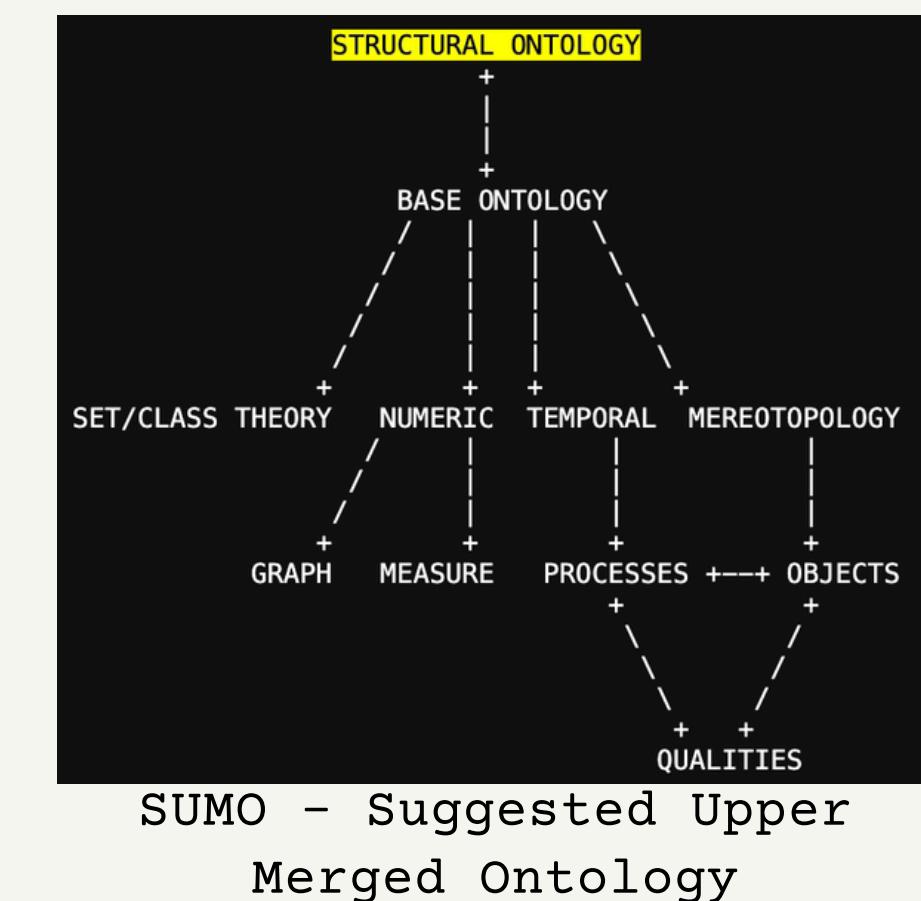
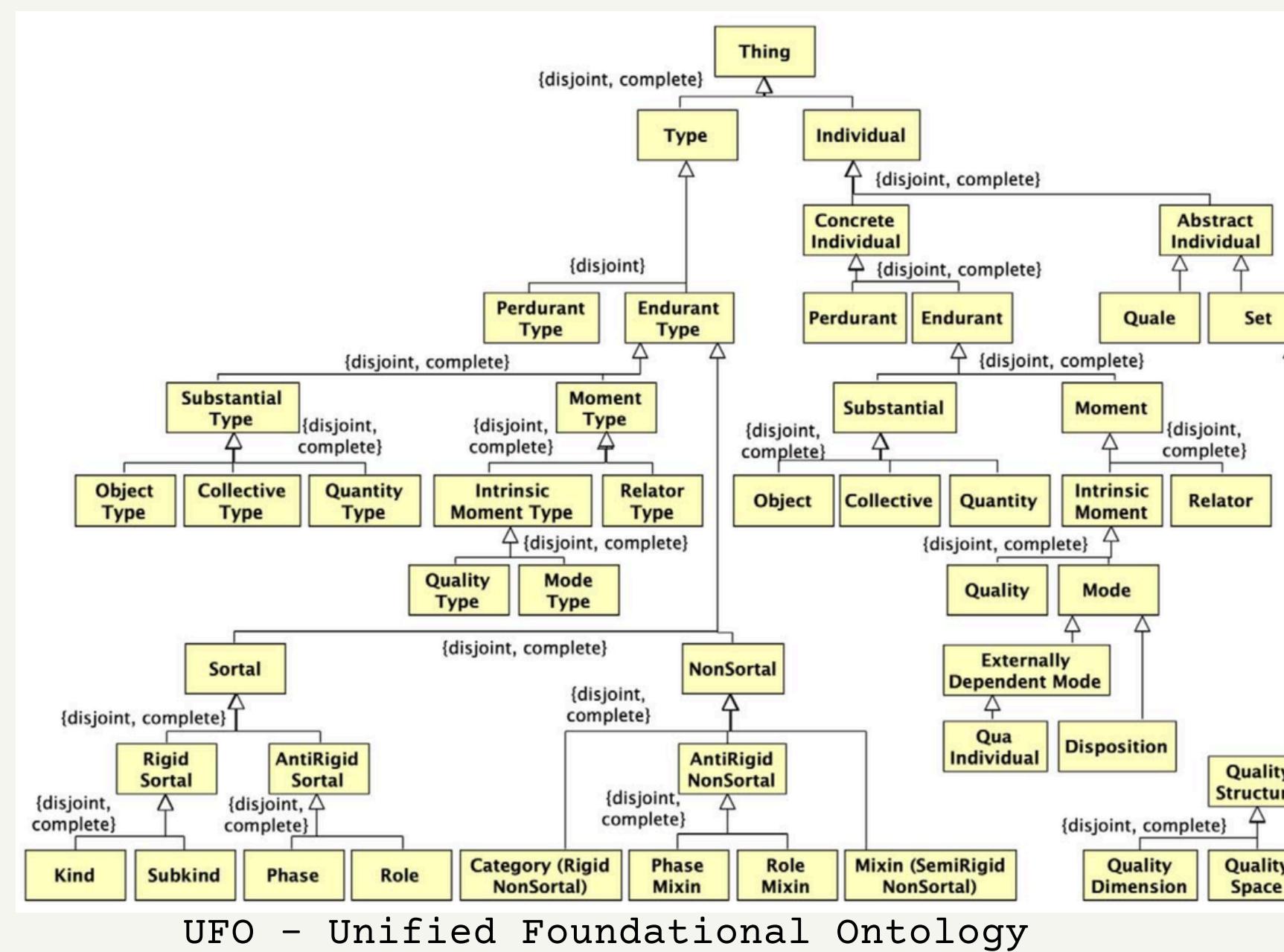
## DOLCE - Descriptive Ontology for Linguistic and Cognitive Engineering





## Foundational Mereological

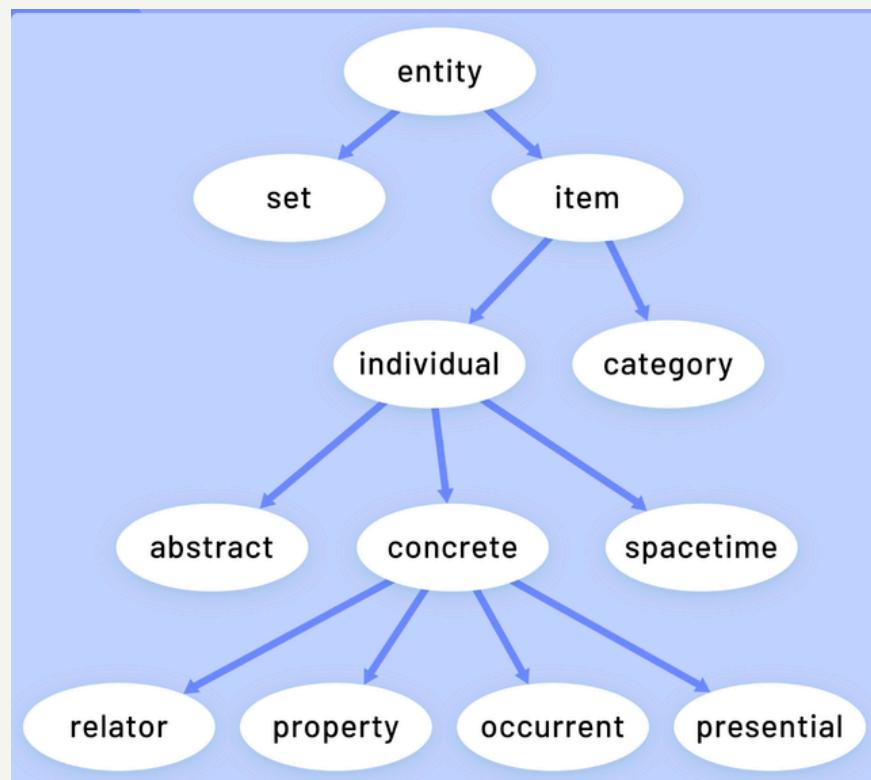
commitment towards conflicting foundational theories  
commitment towards conflicting mereological theories



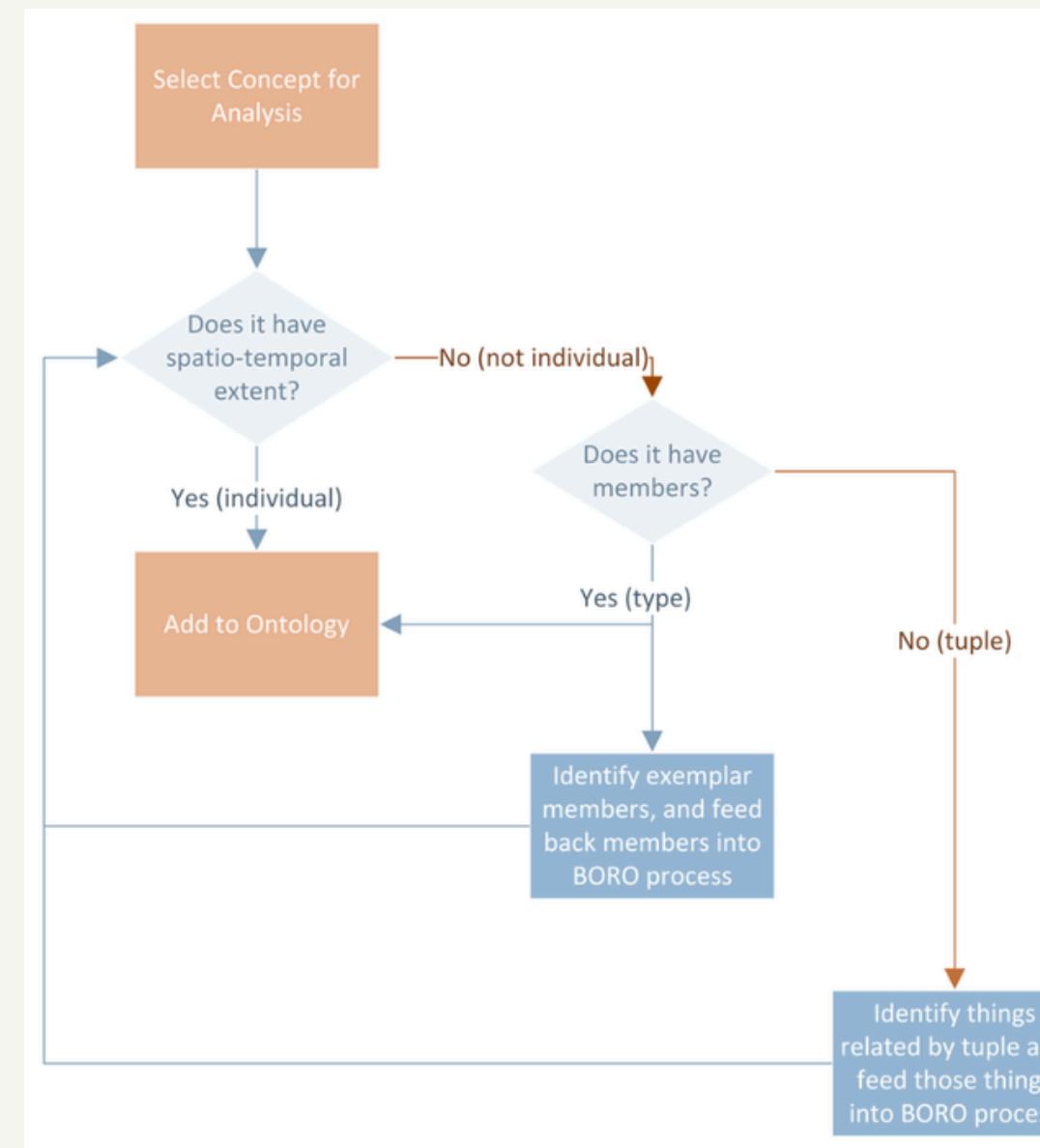


## Foundational Mereological

commitment towards conflicting foundational theories  
commitment towards conflicting mereological theories



GFO – General Formal  
Ontology



BORO – Business Objects  
Reference Ontology



EMMO – Elementary  
Multiperspective  
Material Ontology

4Dimensionalism



TOP LEVEL

**Foundational  
Mereological**

commitment towards conflicting foundational theories  
commitment towards conflicting mereological theories

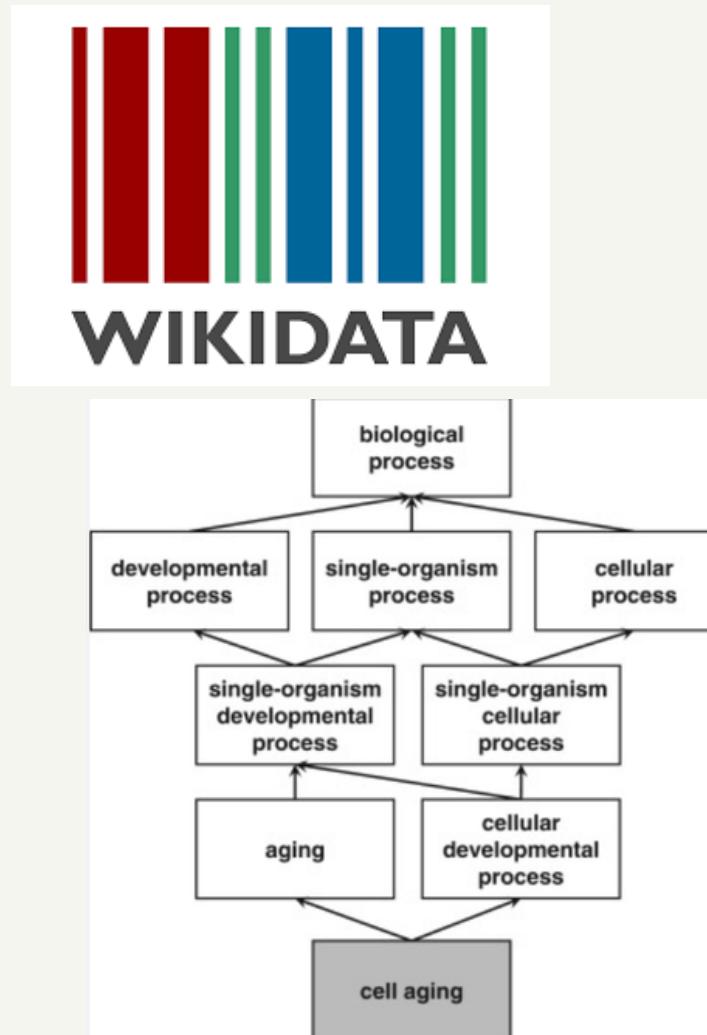


DOMAIN  
LEVEL

**Domain Theory  
Status of Element**

core/domain theories conflictuality  
specific conflicting scientific theories

**SEMANTICS  
NEGOTIATION  
ISSUES**

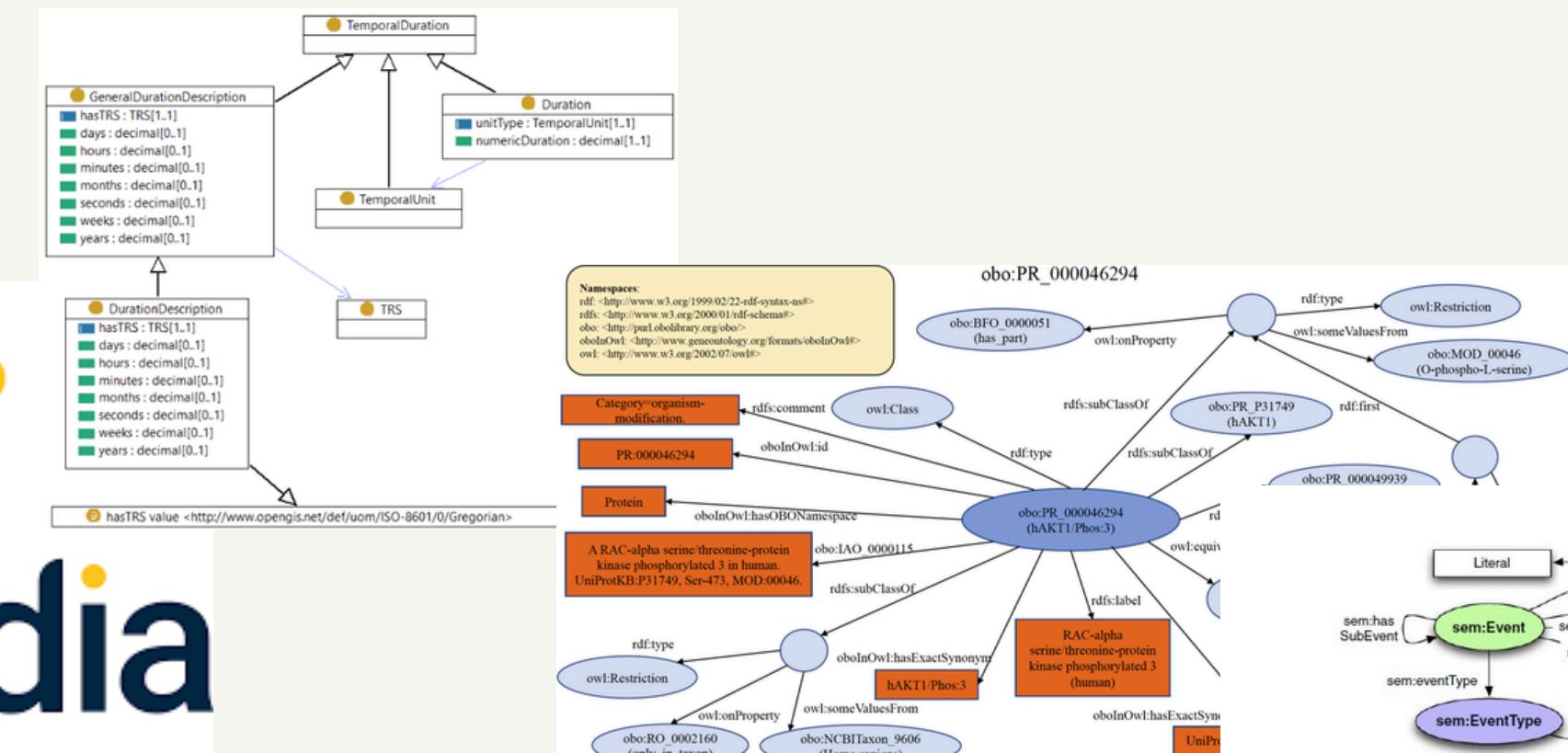


## Foundational Mereological

commitment towards conflicting foundational theories  
commitment towards conflicting mereological theories

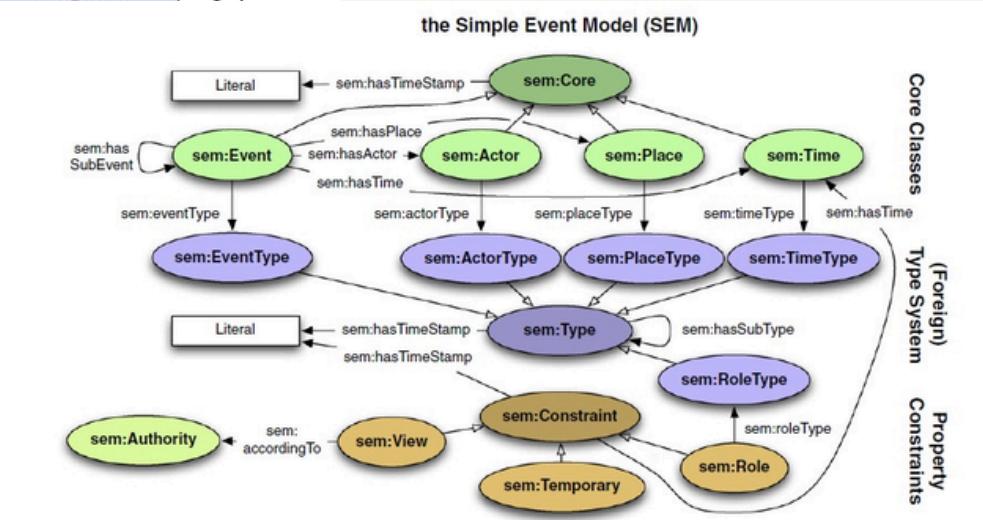
## Domain Theory Status of Element

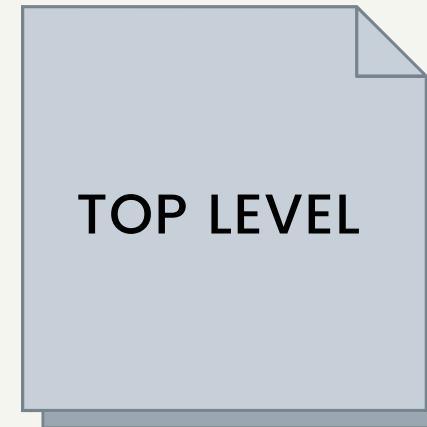
core/domain theories conflictuality  
specific conflicting scientific theories



Gene ontology  
Protein Ontology  
Wikidata  
DBpedia  
Time Ontology  
Simple Event Model

...





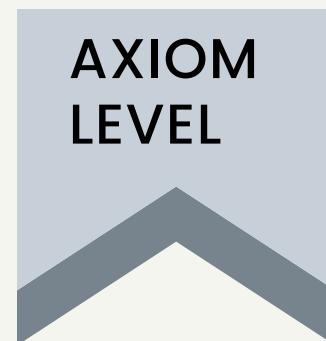
TOP LEVEL

**Foundational  
Mereological**

commitment towards conflicting foundational theories  
commitment towards conflicting mereological theories

DOMAIN  
LEVEL**Domain Theory  
Status of Element**

core/domain theories conflictuality  
specific conflicting scientific theories

AXIOM  
LEVEL**Ontological Level  
Language Family**

axiomatic conflictuality  
language expressivity

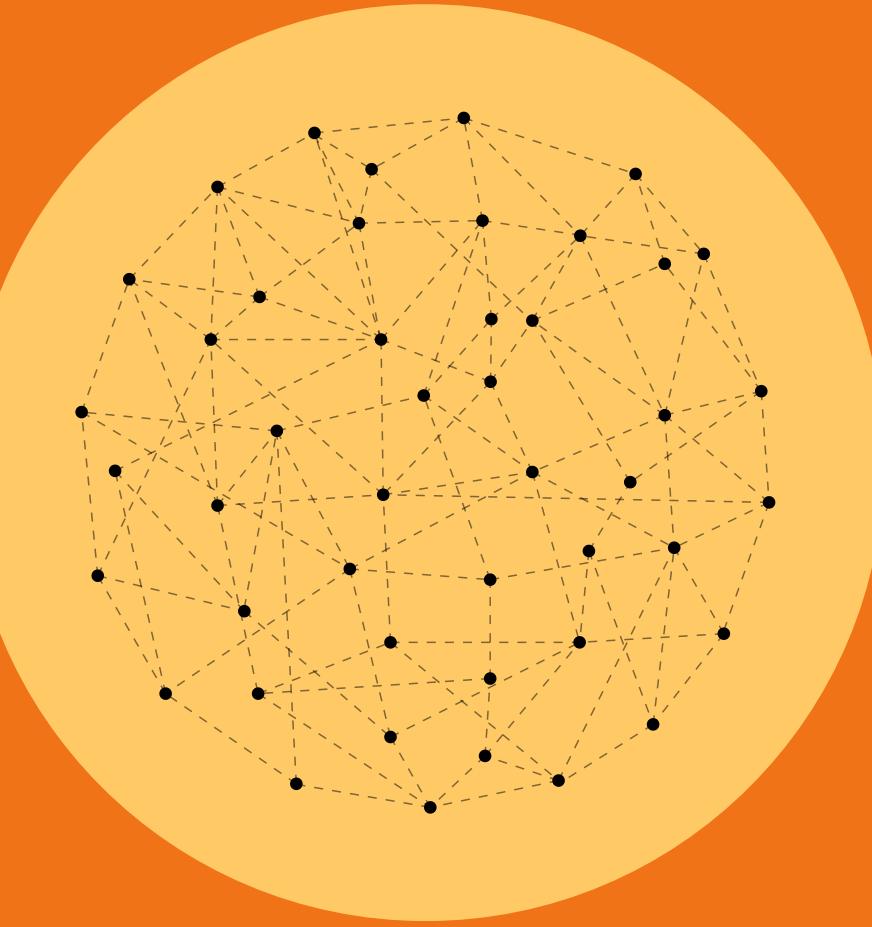
**SEMANTICS  
NEGOTIATION  
ISSUES**



**SEMANTICS  
NEGOTIATION  
ISSUES**

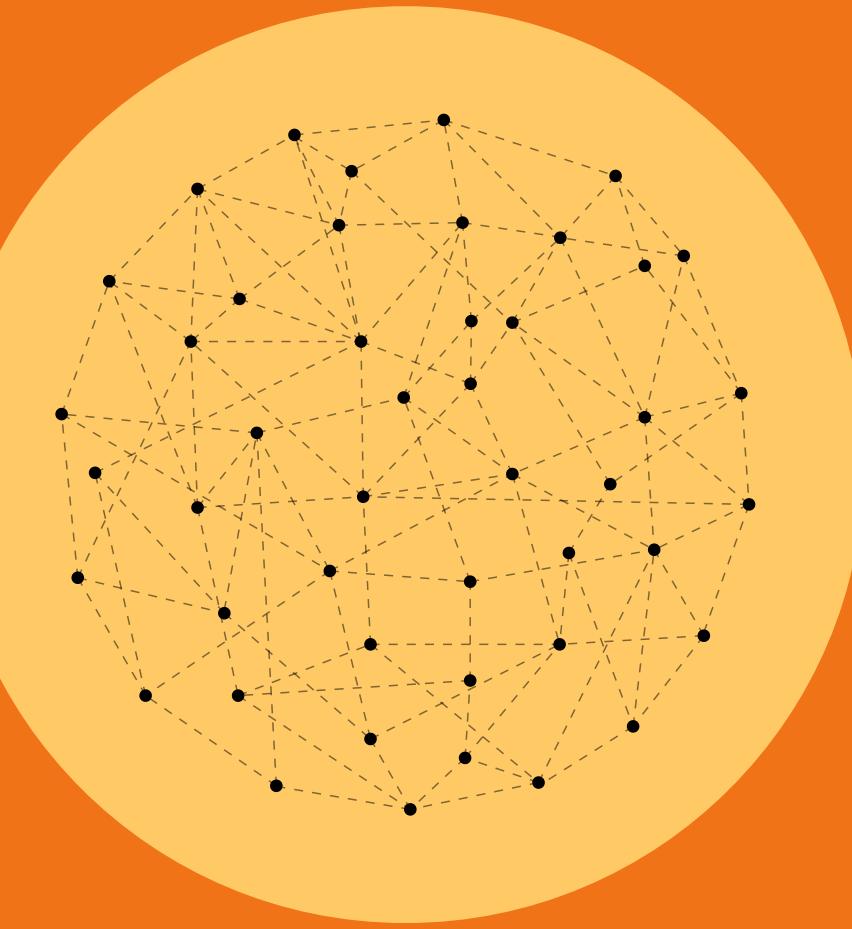
# Ontology testing

- OOPS! - OntOlogy Pitfall Scanner



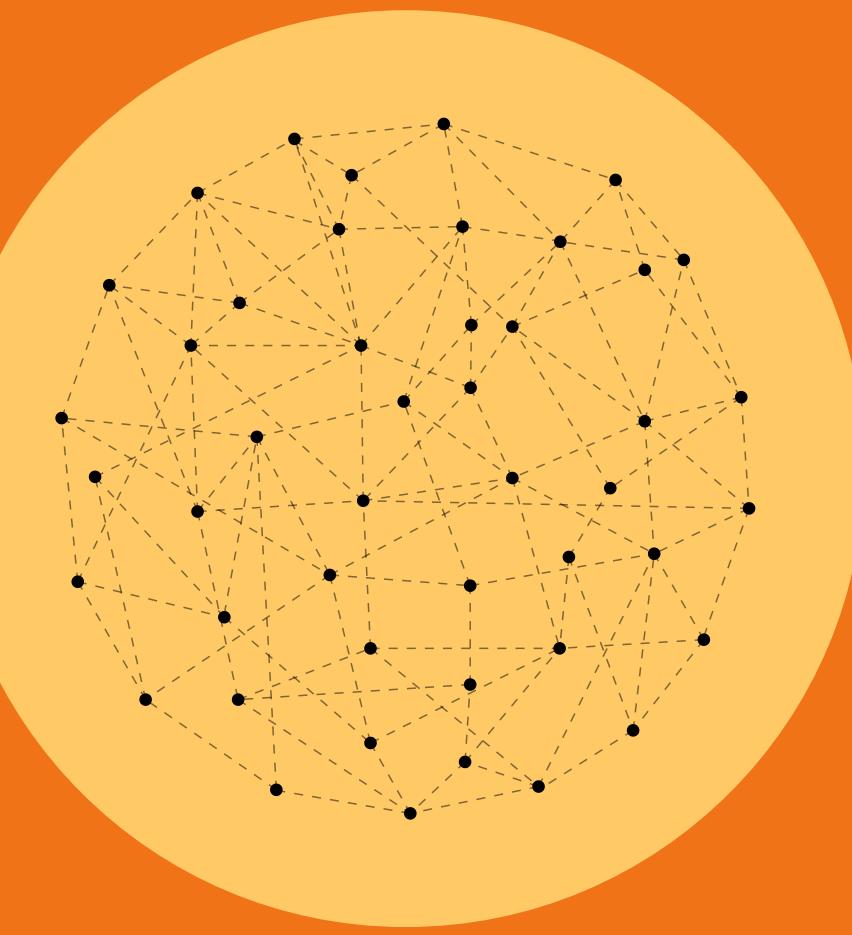
# Ontology visual Modeling

- ChOWLk
- Graffoo



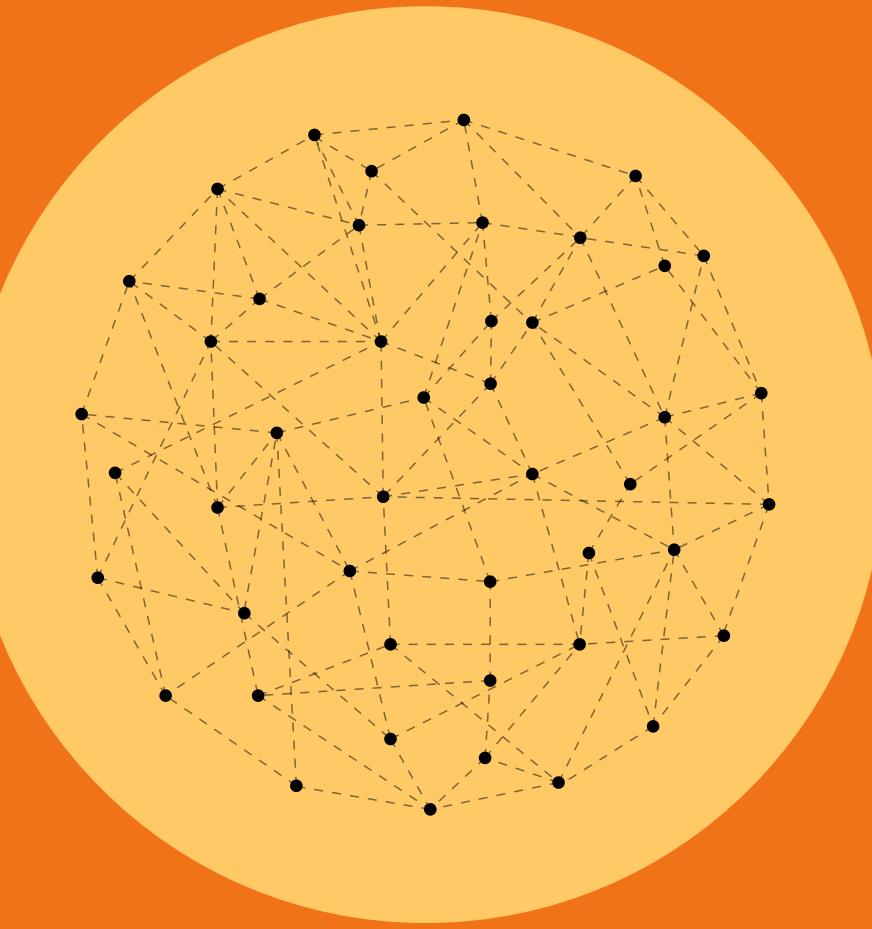
# Ontology visualization tools

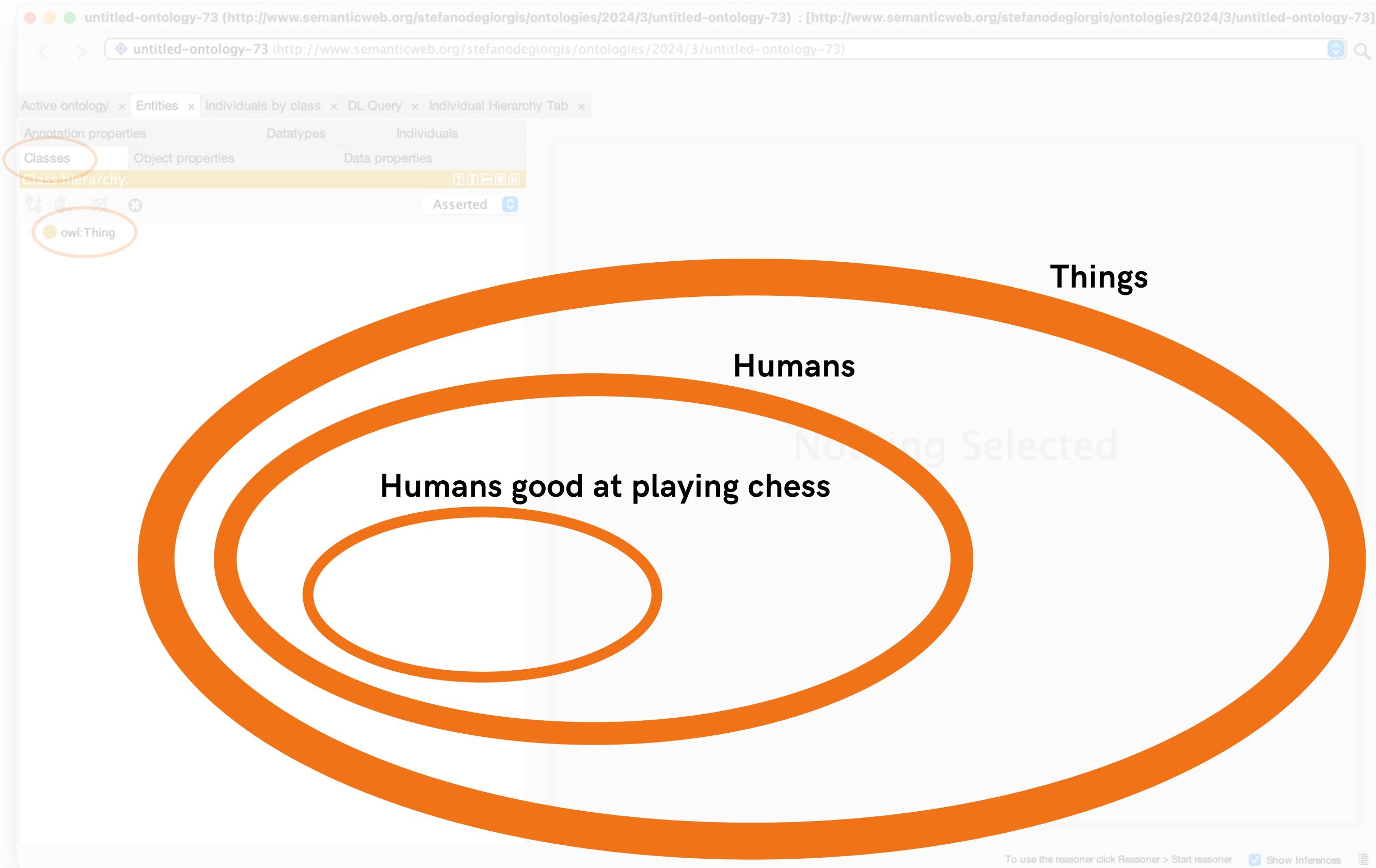
- Protégé
- PoolParty suite
- TopBraid
- Neo4j suite

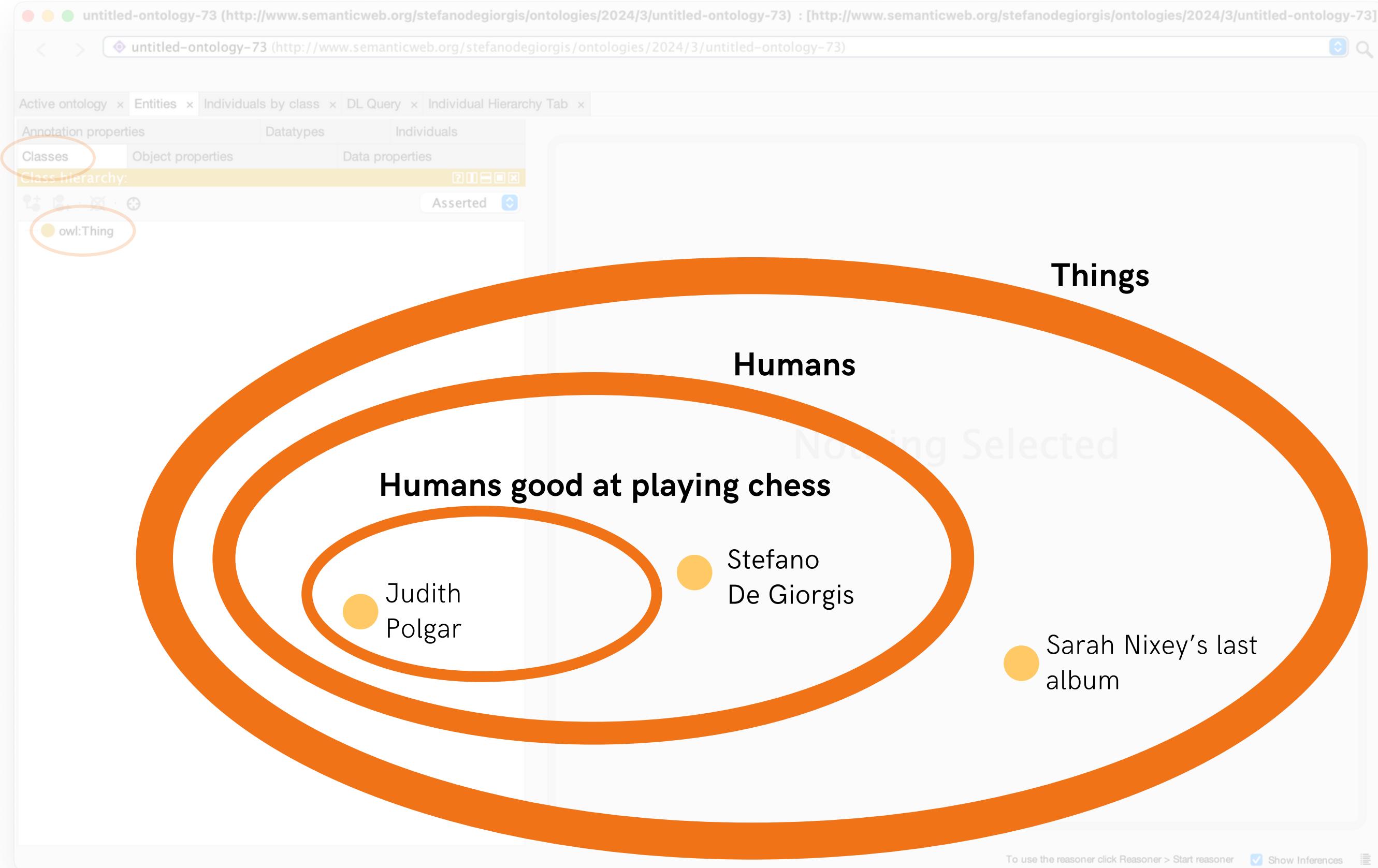


# Protégé

---







untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>) : [<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>]

< > untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>)

Active ontology x Entities x Individuals by class x DL Query x Individual Hierarchy Tab x

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Class hierarchy: owl:Thing

owl:Thing

Asserted

Annotations +

Annotations: owl:Thing

Description: owl:Thing

Equivalent To +

SubClass Of +

General class axioms +

SubClass Of (Anonymous Ancestor)

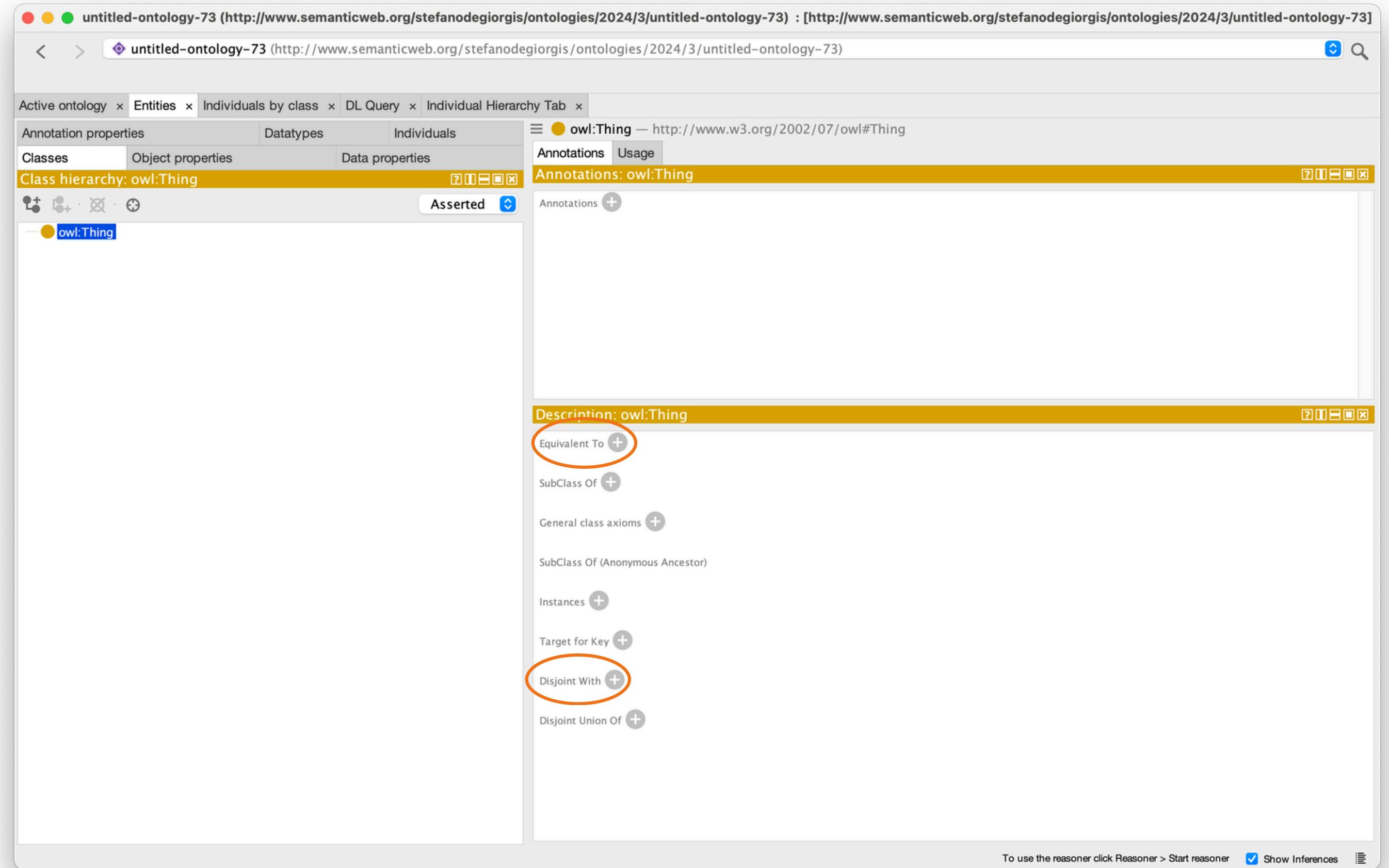
Instances +

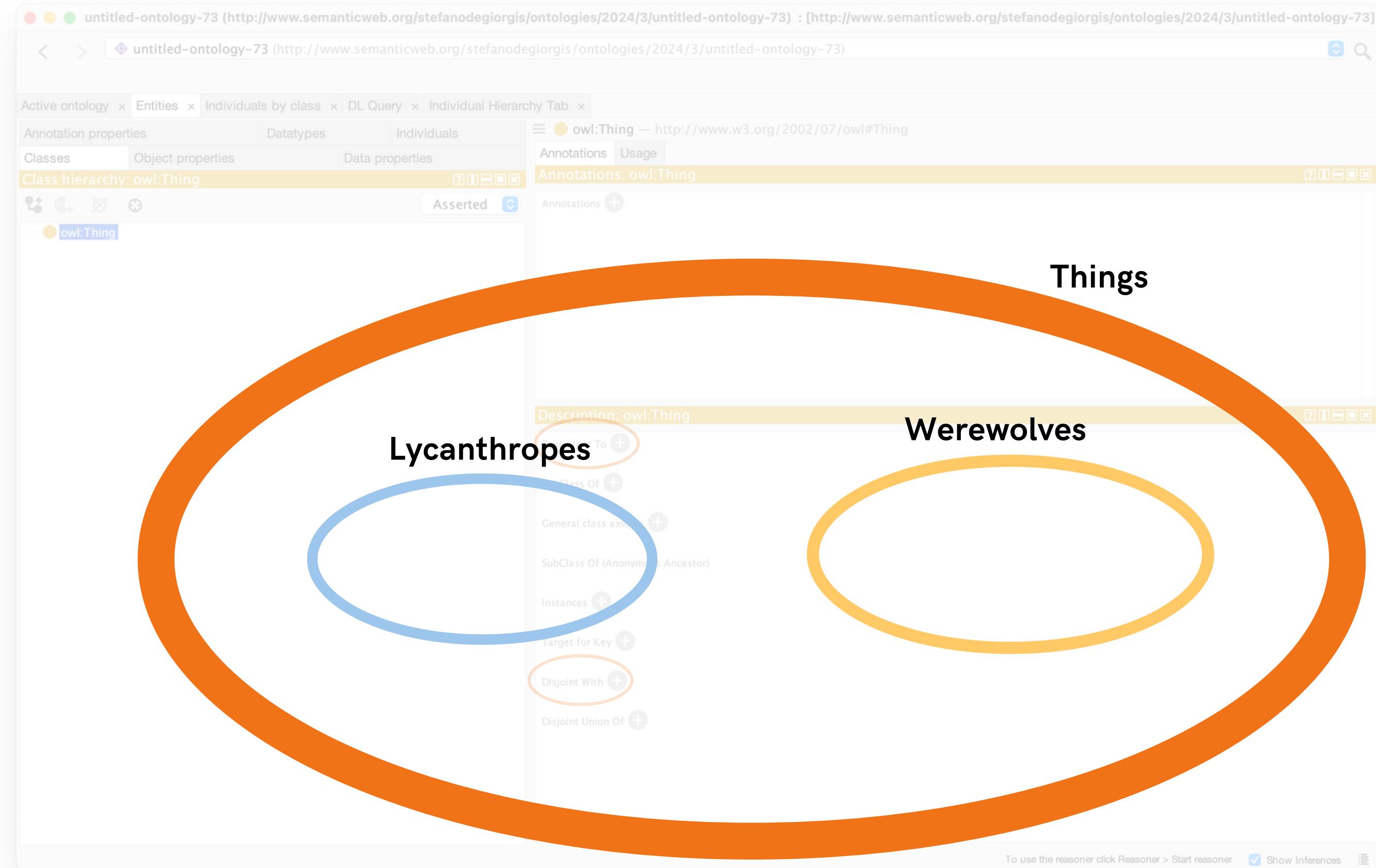
Target for Key +

Disjoint With +

Disjoint Union Of +

To use the reasoner click Reasoner > Start reasoner  Show Inferences





untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>) : [<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>]

< > untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>)

Active ontology x Entities x Individuals by class x DL Query x Individual Hierarchy Tab x

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Class hierarchy: owl:Thing

owl:Thing

Annotations Usage

Annotations: owl:Thing

Annotations

Asserted

Annotations

owl:Thing

Things

Description: owl:Thing

Lycanthropes - Werewolves

SubClass Of

General class axioms

SubClass Of (Anonymous Ancestor)

Implies

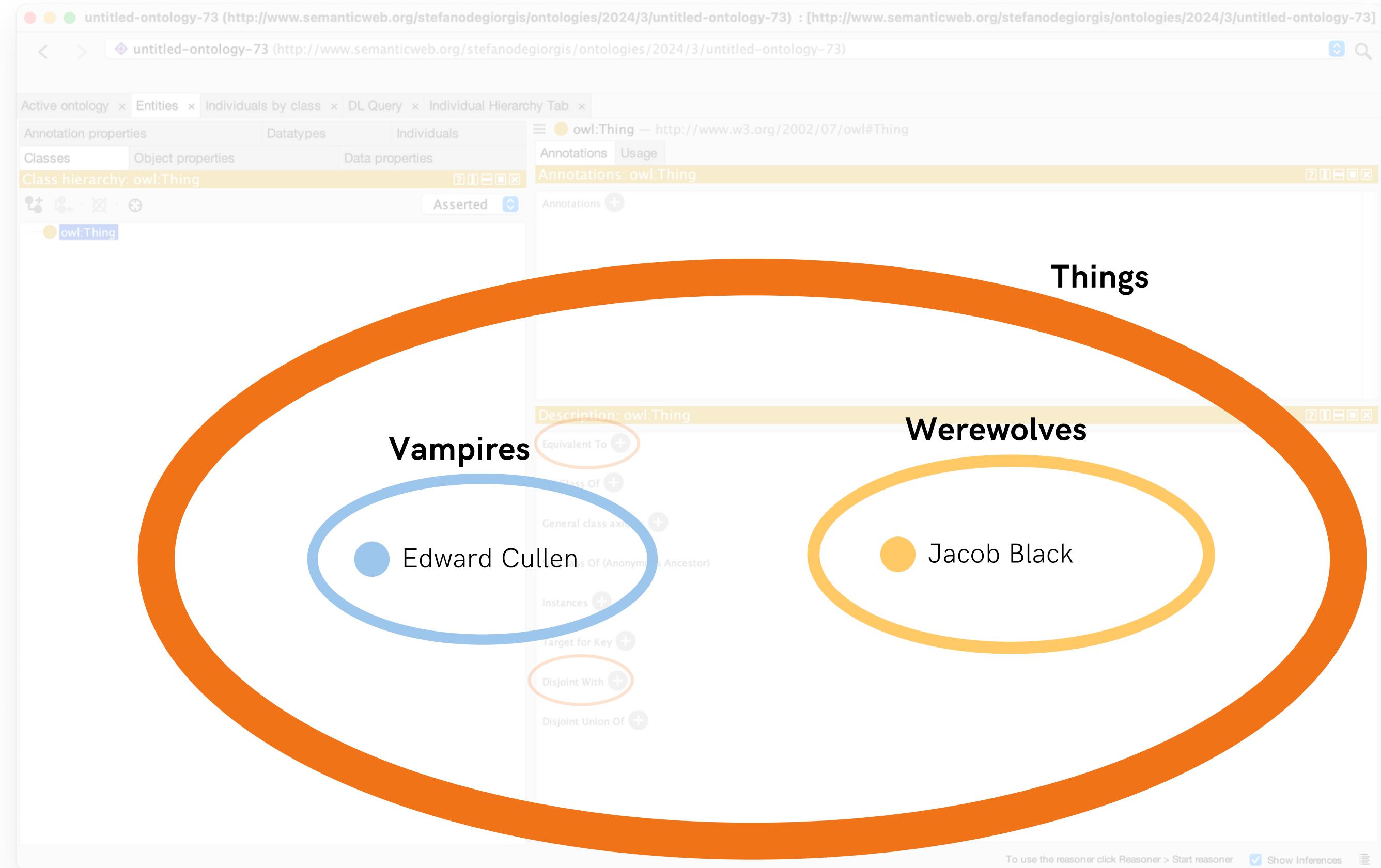
Target for Key

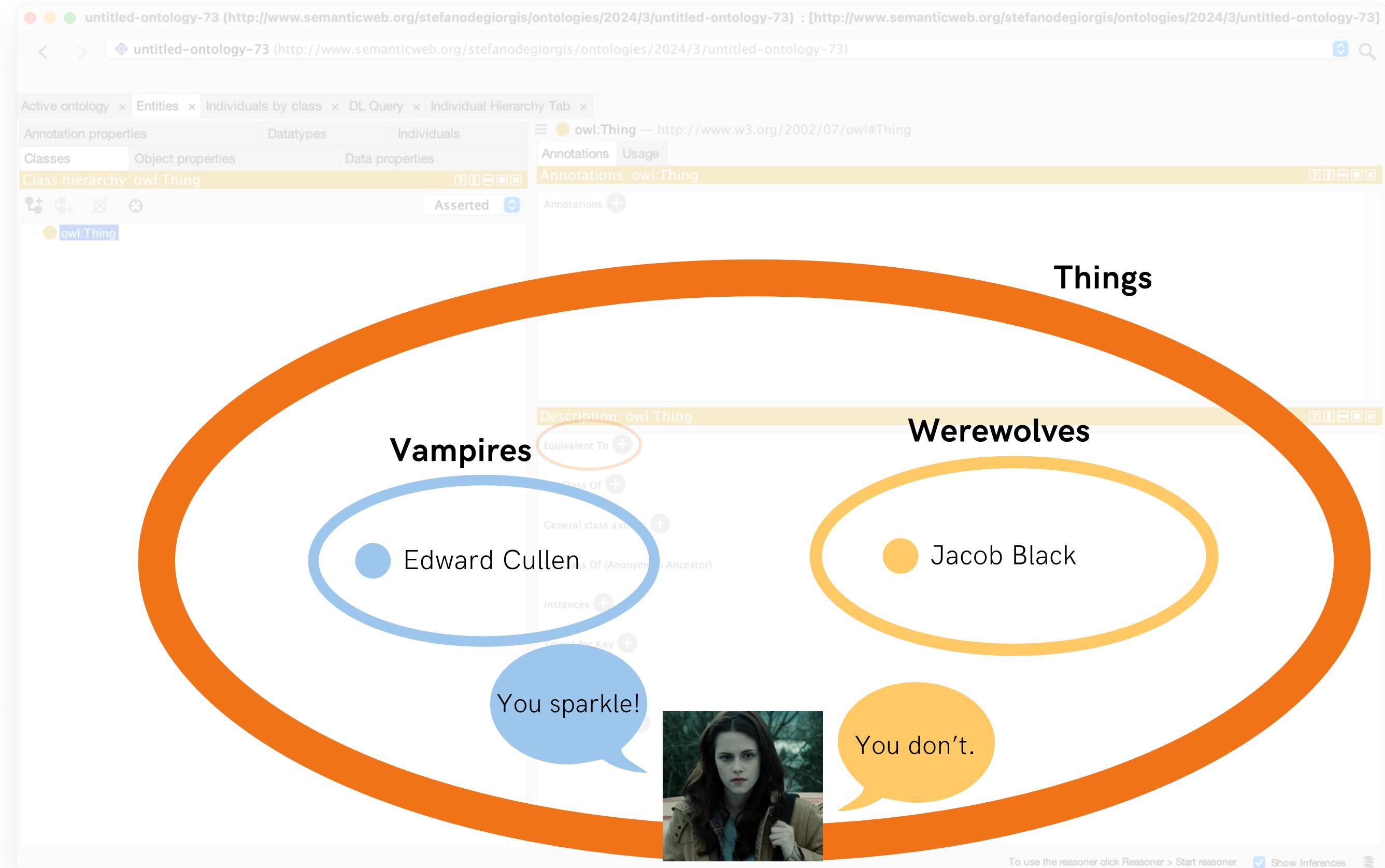
Disjoint With

Disjoint Union Of

To use the reasoner click Reasoner > Start reasoner  Show Inferences

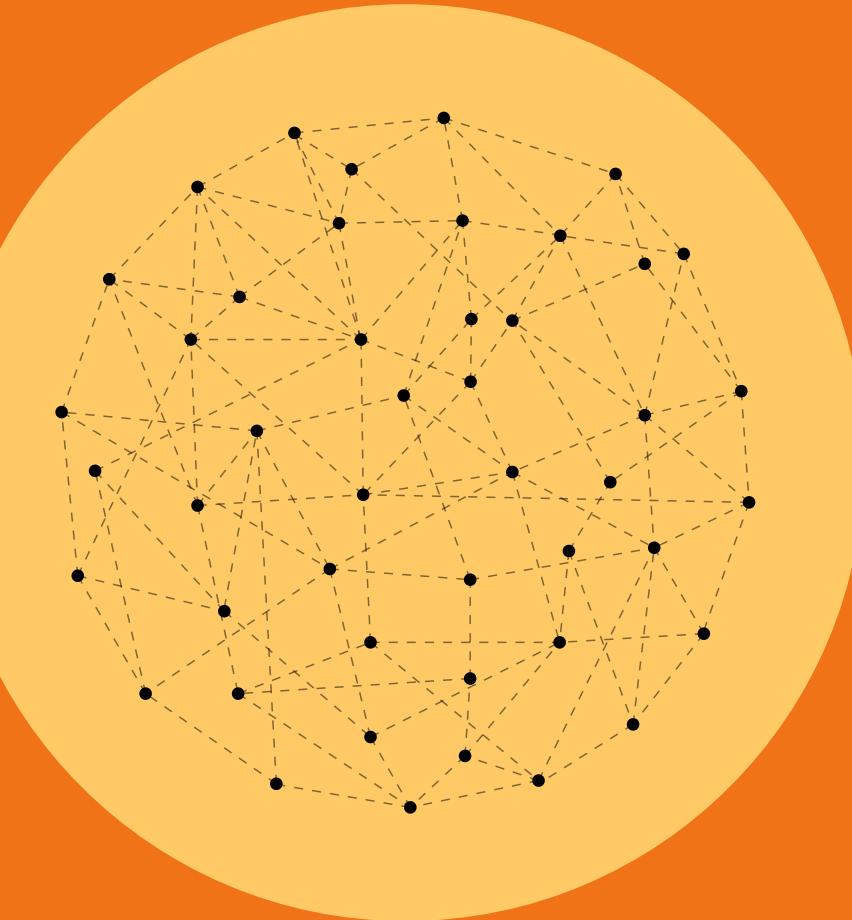
The screenshot shows the Protégé ontology editor interface. The top navigation bar includes tabs for Active ontology, Entities, Individuals by class, DL Query, and Individual Hierarchy Tab. Below the tabs are sections for Annotation properties, Datatypes, and Individuals, with sub-sections for Classes, Object properties, and Data properties. The main content area displays the 'Class hierarchy: owl:Thing' with a tree view showing 'owl:Thing' at the root. On the right, there are tabs for Annotations and Usage, with 'Annotations' currently selected. The main workspace shows a large orange circle highlighting the entire content area. Inside this circle, a green oval highlights the 'Annotations' section for the owl:Thing class. The text 'Lycanthropes - Werewolves' is centered within this green oval. Below it, a list of annotations is shown: 'Description: owl:Thing', 'SubClass Of', 'General class axioms', 'SubClass Of (Anonymous Ancestor)', 'Implies', 'Target for Key', 'Disjoint With', and 'Disjoint Union Of'. The bottom of the screen has a footer with the text 'To use the reasoner click Reasoner > Start reasoner' and a checked checkbox for 'Show Inferences'.





# OWL properties

- functionality
- simmetry
- transitivity
- ...



untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>) : [<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>]

< > [untitled-ontology-73 \(<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>\)](#) [☰](#) [🔍](#)

Active ontology x Entities x Individuals by class x DL Query x Individual Hierarchy Tab x

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Object property hierarchy: owl:topObjectProperty

owl:topObjectProperty

Annotations Usage

Annotations: owl:topObjectProperty

Annotations +

Characterist [+] Description: owl:topObjectProperty

Functional

Inverse functional

Transitive

Symmetric

Asymmetric

Reflexive

Irreflexive

equivalent To +

subProperty Of +

inverse Of +

domains (intersection) +

ranges (intersection) +

disjoint With +

superProperty Of (Chain) +

To use the reasoner click Reasoner > Start reasoner  Show Inferences [☰](#)

The screenshot shows the Protégé ontology editor interface. The main title bar indicates the active ontology is 'untitled-ontology-73'. The top navigation bar includes tabs for Active ontology, Entities, Individuals by class, DL Query, and Individual Hierarchy Tab. Below this is a secondary navigation bar with tabs for Annotation properties, Datatypes, and Individuals, and sub-tabs for Classes, Object properties, and Data properties. The main content area displays the 'Object property hierarchy' for 'owl:topObjectProperty'. On the left, a tree view shows a single node 'owl:topObjectProperty'. On the right, the 'Annotations' tab is selected, showing the 'Annotations: owl:topObjectProperty' panel. This panel contains a list of annotations with '+' signs next to them for expansion. A vertical sidebar on the left has an orange border around its top section, highlighting the annotations panel. At the bottom, there is a footer with the text 'To use the reasoner click Reasoner > Start reasoner' and a checked checkbox for 'Show Inferences'.

Property Type	Description	De Facto
<b>Functional</b>	A functional property is one where if property P connects a subject S to an object O1, and P also connects S to O2, then O1 and O2 must be the same entity.	"Each subject can be associated with only one object."
<b>Inverse Functional</b>	An inverse functional property means that if property P connects S1 to O, and P also connects S2 to O, then S1 and S2 must be the same entity.	"Each object can be associated with only one subject."
<b>Transitive</b>	A property is transitive if whenever property P connects S to O, and O to another entity O2, then P also connects S to O2.	"Relationship extends over a chain of instances."
<b>Symmetric</b>	A symmetric property is one where if P connects S to O, then P must also connect O to S.	"If a relationship exists in one direction, it exists in the reverse as well."
<b>Asymmetric</b>	An asymmetric property is such that if P connects S to O, then P cannot connect O to S. It is the opposite of symmetric.	"A relationship in one direction prevents its reverse."
<b>Reflexive</b>	A reflexive property requires that if P connects S to O, then S must also be connected to itself via P.	"Each instance must be related to itself."
<b>Irreflexive</b>	An irreflexive property is one where if P connects S to O, then S cannot be connected to itself via P. It is the opposite of reflexive.	"No instance can be related to itself."

Property Type	Description	De Facto
Functional	A functional property is one where if property P connects a subject S to an object O <sub>1</sub> , and P also connects S to O <sub>2</sub> , then O <sub>1</sub> and O <sub>2</sub> must be the same entity.	"Each subject can be associated with only one object."
Inverse Functional	An inverse functional property means that if property P connects S <sub>1</sub> to O, and P also connects S <sub>2</sub> to O, then S <sub>1</sub> and S <sub>2</sub> must be the same entity.	"Each object can be associated with only one subject."
Transitive	A property is transitive if whenever property P connects S to O, and O to another entity O <sub>2</sub> , then P also connects S to O <sub>2</sub> .	"Relationship extends over a chain of instances."
Symmetric	A symmetric property is one where if P connects S to O, then P must also connect O to S.	"If a relationship exists in one direction, it exists in the reverse as well."
Asymmetric	An asymmetric property is such that if P connects S to O, then P cannot connect O to S. It is the opposite of symmetric.	"A relationship in one direction prevents its reverse."
Reflexive	A reflexive property requires that if P connects S to O, then S must also be connected to itself via P.	"Each instance must be related to itself."
Irreflexive	An irreflexive property is one where if P connects S to O, then S cannot be connected to itself via P. It is the opposite of reflexive.	"No instance can be related to itself."

:Person1 :hasBirthPlace :Place1  
 :Person1 :hasBirthTime :Time1

Property Type	Description	De Facto
Functional	A functional property is one where if property P connects a subject S to an object O <sub>1</sub> , and P also connects S to O <sub>2</sub> , then O <sub>1</sub> and O <sub>2</sub> must be the same entity.	"Each subject can be associated with only one object."
Inverse Functional	An inverse functional property means that if property P connects S <sub>1</sub> to O, and P also connects S <sub>2</sub> to O, then S <sub>1</sub> and S <sub>2</sub> must be the same entity.	"Each object can be associated with only one subject."
Transitive	A property is transitive if whenever property P connects S to O, and O to another entity O <sub>2</sub> , then P also connects S to O <sub>2</sub> .	"Relationship extends over a chain of instances."
Symmetric	A symmetric property is one where if P connects S to O, then P must also connect O to S.	"If a relationship exists in one direction, it exists in the reverse as well."
Asymmetric	An asymmetric property is such that if P connects S to O, then P cannot connect O to S. It is the opposite of symmetric.	"A relationship in one direction prevents its reverse."
Reflexive	A reflexive property requires that if P connects S to O, then S must also be connected to itself via P.	"Each instance must be related to itself."
Irreflexive	An irreflexive property is one where if P connects S to O, then S cannot be connected to itself via P. It is the opposite of reflexive.	"No instance can be related to itself."

:Person1 :hasBirthPlace :Place1  
 :Person1 :hasBirthTime :Time1

:Person1 :biologicalMotherOf :Person2

Property Type	Description	De Facto
Functional	A functional property is one where if property P connects a subject S to an object O <sub>1</sub> , and P also connects S to O <sub>2</sub> , then O <sub>1</sub> and O <sub>2</sub> must be the same entity.	"Each subject can be associated with only one object."
Inverse Functional	An inverse functional property means that if property P connects S <sub>1</sub> to O, and P also connects S <sub>2</sub> to O, then S <sub>1</sub> and S <sub>2</sub> must be the same entity.	"Each object can be associated with only one subject."
Transitive	A property is transitive if whenever property P connects S to O, and O to another entity O <sub>2</sub> , then P also connects S to O <sub>2</sub> .	"Relationship extends over a chain of instances."
Symmetric	A symmetric property is one where if P connects S to O, then P must also connect O to S.	"If a relationship exists in one direction, it exists in the reverse as well."
Asymmetric	An asymmetric property is such that if P connects S to O, then P cannot connect O to S. It is the opposite of symmetric.	"A relationship in one direction prevents its reverse."
Reflexive	A reflexive property requires that if P connects S to O, then S must also be connected to itself via P.	"Each instance must be related to itself."
Irreflexive	An irreflexive property is one where if P connects S to O, then S cannot be connected to itself via P. It is the opposite of reflexive.	"No instance can be related to itself."

:Person1 :hasBirthPlace :Place1

:Person1 :hasBirthTime :Time1

:Person1 :biologicalMotherOf :Person2

:Region1 :subRegionOf :Region2

Property Type	Description	De Facto
Functional	A functional property is one where if property P connects a subject S to an object O1, and P also connects S to O2, then O1 and O2 must be the same entity.	"Each subject can be associated with only one object."
Inverse Functional	An inverse functional property means that if property P connects S1 to O, and P also connects S2 to O, then S1 and S2 must be the same entity.	"Each object can be associated with only one subject."
Transitive	A property is transitive if whenever property P connects S to O, and O to another entity O2, then P also connects S to O2.	"Relationship extends over a chain of instances."
Symmetric	A symmetric property is one where if P connects S to O, then P must also connect O to S.	"If a relationship exists in one direction, it exists in the reverse as well."
Asymmetric	An asymmetric property is such that if P connects S to O, then P cannot connect O to S. It is the opposite of symmetric.	"A relationship in one direction prevents its reverse."
Reflexive	A reflexive property requires that if P connects S to O, then S must also be connected to itself via P.	"Each instance must be related to itself."
Irreflexive	An irreflexive property is one where if P connects S to O, then S cannot be connected to itself via P. It is the opposite of reflexive.	"No instance can be related to itself."

:Person1 :hasBirthPlace :Place1

:Person1 :hasBirthTime :Time1

:Person1 :biologicalMotherOf :Person2

:Region1 :subRegionOf :Region2

:Region1 :adjacentTo :Region2

Property Type	Description	De Facto
Functional	A functional property is one where if property P connects a subject S to an object O1, and P also connects S to O2, then O1 and O2 must be the same entity.	"Each subject can be associated with only one object."
Inverse Functional	An inverse functional property means that if property P connects S1 to O, and P also connects S2 to O, then S1 and S2 must be the same entity.	"Each object can be associated with only one subject."
Transitive	A property is transitive if whenever property P connects S to O, and O to another entity O2, then P also connects S to O2.	"Relationship extends over a chain of instances."
Symmetric	A symmetric property is one where if P connects S to O, then P must also connect O to S.	"If a relationship exists in one direction, it exists in the reverse as well."
Asymmetric	An asymmetric property is such that if P connects S to O, then P cannot connect O to S. It is the opposite of symmetric.	"A relationship in one direction prevents its reverse."
Reflexive	A reflexive property requires that if P connects S to O, then S must also be connected to itself via P.	"Each instance must be related to itself."
Irreflexive	An irreflexive property is one where if P connects S to O, then S cannot be connected to itself via P. It is the opposite of reflexive.	"No instance can be related to itself."

:Person1 :hasBirthPlace :Place1

:Person1 :hasBirthTime :Time1

:Person1 :biologicalMotherOf :Person2

:Region1 :subRegionOf :Region2

:Region1 :adjacentTo :Region2

:Person1 :biologicalMotherOf :Person2

Property Type	Description	De Facto
Functional	A functional property is one where if property P connects a subject S to an object O <sub>1</sub> , and P also connects S to O <sub>2</sub> , then O <sub>1</sub> and O <sub>2</sub> must be the same entity.	"Each subject can be associated with only one object."
Inverse Functional	An inverse functional property means that if property P connects S <sub>1</sub> to O, and P also connects S <sub>2</sub> to O, then S <sub>1</sub> and S <sub>2</sub> must be the same entity.	"Each object can be associated with only one subject."
Transitive	A property is transitive if whenever property P connects S to O, and O to another entity O <sub>2</sub> , then P also connects S to O <sub>2</sub> .	"Relationship extends over a chain of instances."
Symmetric	A symmetric property is one where if P connects S to O, then P must also connect O to S.	"If a relationship exists in one direction, it exists in the reverse as well."
Asymmetric	An asymmetric property is such that if P connects S to O, then P cannot connect O to S. It is the opposite of symmetric.	"A relationship in one direction prevents its reverse."
Reflexive	A reflexive property requires that if P connects S to O, then S must also be connected to itself via P.	"Each instance must be related to itself."
Irreflexive	An irreflexive property is one where if P connects S to O, then S cannot be connected to itself via P. It is the opposite of reflexive.	"No instance can be related to itself."

:Person1 :hasBirthPlace :Place1  
 :Person1 :hasBirthTime :Time1  
 :Person1 :biologicalMotherOf :Person2  
 :Region1 :subRegionOf :Region2  
 :Region1 :adjacentTo :Region2  
 :Person1 :biologicalMotherOf :Person2  
 :Person1 :samIdentityAs :Person2

Property Type	Description	De Facto
Functional	A functional property is one where if property P connects a subject S to an object O <sub>1</sub> , and P also connects S to O <sub>2</sub> , then O <sub>1</sub> and O <sub>2</sub> must be the same entity.	"Each subject can be associated with only one object."
Inverse Functional	An inverse functional property means that if property P connects S <sub>1</sub> to O, and P also connects S <sub>2</sub> to O, then S <sub>1</sub> and S <sub>2</sub> must be the same entity.	"Each object can be associated with only one subject."
Transitive	A property is transitive if whenever property P connects S to O, and O to another entity O <sub>2</sub> , then P also connects S to O <sub>2</sub> .	"Relationship extends over a chain of instances."
Symmetric	A symmetric property is one where if P connects S to O, then P must also connect O to S.	"If a relationship exists in one direction, it exists in the reverse as well."
Asymmetric	An asymmetric property is such that if P connects S to O, then P cannot connect O to S. It is the opposite of symmetric.	"A relationship in one direction prevents its reverse."
Reflexive	A reflexive property requires that if P connects S to O, then S must also be connected to itself via P.	"Each instance must be related to itself."
Irreflexive	An irreflexive property is one where if P connects S to O, then S cannot be connected to itself via P. It is the opposite of reflexive.	"No instance can be related to itself."

:Person1 :hasBirthPlace :Place1  
 :Person1 :hasBirthTime :Time1  
 :Person1 :biologicalMotherOf :Person2  
 :Region1 :subRegionOf :Region2  
 :Region1 :adjacentTo :Region2  
 :Person1 :biologicalMotherOf :Person2  
 :Person1 :samIdentityAs :Person2  
 :Person1 :biologicalMotherOf :Person2

untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>) : [<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>]

< > [untitled-ontology-73 \(<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>\)](#) [☰](#) [🔍](#)

Active ontology x Entities x Individuals by class x DL Query x Individual Hierarchy Tab x

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Object property hierarchy: owl:topObjectProperty

owl:topObjectProperty

Annotations Usage

Annotations: owl:topObjectProperty

Annotations +

Characterist [ ] Description: owl:topObjectProperty [ ]

Functional

Inverse functional

Transitive

Symmetric

Asymmetric

Reflexive

Irreflexive

Equivalent To +

SubProperty Of +

Inverse Of +

Domains (intersection) +

Ranges (intersection) +

Disjoint With +

SuperProperty Of (Chain) +

To use the reasoner click Reasoner > Start reasoner  Show Inferences [☰](#)

The screenshot shows the Protégé ontology editor interface. The main focus is the 'Annotations' tab for the 'owl:topObjectProperty' object property. This tab is highlighted with an orange border. On the left, there's a sidebar titled 'Object property hierarchy: owl:topObjectProperty' which lists 'owl:topObjectProperty'. Below the sidebar, there are several characteristic checkboxes: Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, and Irreflexive. To the right of these checkboxes is a list of annotations with '+' icons: Equivalent To, SubProperty Of, Inverse Of, Domains (intersection), Ranges (intersection), Disjoint With, and SuperProperty Of (Chain). At the bottom of the screen, there are status messages: 'To use the reasoner click Reasoner > Start reasoner' and 'Show Inferences' with a checked checkbox.

untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>) : [<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>]

untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>)

Active ontology x Entities x Individuals by class x DL Query x Individual Hierarchy Tab x

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Object property hierarchy: owl:topObjectProperty

owl:topObjectProperty

Annotations Usage

Annotations: owl:topObjectProperty

Annotations +

Characterist Description: owl:topObjectProperty

Functional

Inverse functional

Transitive

Symmetric

Asymmetric

Reflexive

Irreflexive

Equivalent To +

SubProperty Of +

Inverse Of +

Domains (intersection) +

Ranges (intersection) +

Disjoint With +

SuperProperty Of (Chain) +

:parentOf :subPropertyOf :relativeOf

To use the reasoner click Reasoner > Start reasoner  Show Inferences

untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>) : [<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>]

untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>)

Active ontology x Entities x Individuals by class x DL Query x Individual Hierarchy Tab x

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Object property hierarchy: owl:topObjectProperty

owl:topObjectProperty

Annotations Usage

Annotations: owl:topObjectProperty

Annotations +

Characterist [ ] Description: owl:topObjectProperty

Functional

Inverse functional

Transitive

Symmetric

Asymmetric

Reflexive

Irreflexive

Equivalent To +

SubProperty Of +

Inverse Of +

Domains (intersection) +

Ranges (intersection) +

Disjoint With +

SuperProperty Of (Chain) +

:childOf :inverseOf :parentOf

To use the reasoner click Reasoner > Start reasoner  Show Inferences

The screenshot shows the Protégé ontology editor interface. The main window displays the annotations for the owl:topObjectProperty. A tooltip is shown over the Inverse Of button, containing the text ':childOf :inverseOf :parentOf'. The interface includes tabs for Active ontology, Entities, Individuals by class, DL Query, and Individual Hierarchy Tab. The Object property hierarchy section shows owl:topObjectProperty. The Annotations tab is selected in the main panel, which lists various annotations such as Functional, Inverse functional, Transitive, etc., and buttons for adding relationships like Equivalent To, SubProperty Of, and SuperProperty Of.

untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>) : [<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>]

< > [untitled-ontology-73 \(<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>\)](#) [Search](#)

Active ontology x Entities x Individuals by class x DL Query x Individual Hierarchy Tab x

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Object property hierarchy: owl:topObjectProperty

owl:topObjectProperty

Annotations Usage

Annotations: owl:topObjectProperty

Annotations +

Characterist [ ] Description: owl:topObjectProperty [ ]

Functional

Inverse functional

Transitive

Symmetric

Asymmetric

Reflexive

Irreflexive

Equivalent To +

SubProperty Of +

Inverse Of +

Domains (intersection) +

Ranges (intersection) +

Disjoint With +

SuperProperty Of (Chain) +

To use the reasoner click Reasoner > Start reasoner  Show Inferences

The screenshot shows the Protégé ontology editor interface. The main title bar indicates the active ontology is 'untitled-ontology-73'. The top navigation bar includes tabs for Active ontology, Entities, Individuals by class, DL Query, and Individual Hierarchy Tab. Below this is a secondary navigation bar with tabs for Annotation properties, Datatypes, and Individuals, and sub-tabs for Classes, Object properties, and Data properties. The main content area displays the 'Object property hierarchy: owl:topObjectProperty' with a single entry 'owl:topObjectProperty'. On the right, the 'Annotations' tab is selected for 'owl:topObjectProperty', showing a list of annotations. These annotations include 'Functional', 'Inverse functional', 'Transitive', 'Symmetric', 'Asymmetric', 'Reflexive', 'Irreflexive', 'Equivalent To', 'SubProperty Of', 'Inverse Of', 'Domains (intersection)', 'Ranges (intersection)', 'Disjoint With', and 'SuperProperty Of (Chain)'. The 'Domains (intersection)' and 'Ranges (intersection)' annotations are highlighted with red horizontal bars. A large orange rectangle highlights the entire 'Annotations' section. At the bottom, there is a note to 'To use the reasoner click Reasoner > Start reasoner' and a checked checkbox for 'Show Inferences'.

untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>) : [<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>]

Active ontology x Entities x Individuals by class x DL Query x Individual Hierarchy Tab x

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Object property hierarchy: owl:topObjectProperty

owl:topObjectProperty

Annotations: owl:topObjectProperty

Annotations +

PROPERTY

Description: owl:topObjectProperty

Characterist [ ]

Functional

Inverse functional

Transitive

Symmetric

Asymmetric

Reflexive

Irreflexive

Equivalent To +

SubProperty Of +

Inverse Of +

Domains (intersection) +

Ranges (intersection) +

Disjoint With +

SuperProperty Of (Chain) +

DOMAIN

RANGE

To use the reasoner click Reasoner > Start reasoner  Show Inferences

The diagram illustrates the concept of owl:topObjectProperty. It features two large circles: a yellow one on the left labeled 'DOMAIN' and a blue one on the right labeled 'RANGE'. A curved arrow points from the 'DOMAIN' circle to the 'RANGE' circle. Between the two circles is a central box with a light orange border. The box contains the text 'PROPERTY' at the top, followed by 'Description: owl:topObjectProperty'. Below this, there is a list of property characteristics with checkboxes: Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, and Irreflexive. To the right of this list is a vertical column of buttons with labels: 'Equivalent To +', 'SubProperty Of +', 'Inverse Of +', 'Domains (intersection) +', 'Ranges (intersection) +', 'Disjoint With +', and 'SuperProperty Of (Chain) +'. An arrow points from the 'PROPERTY' box towards the 'RANGE' circle.

untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>) : [<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>]

Active ontology x Entities x Individuals by class x DL Query x Individual Hierarchy Tab x

Annotation properties Datatypes Individuals

Classes Object properties Data properties

Object property hierarchy: owl:topObjectProperty

owl:topObjectProperty

Annotations: owl:topObjectProperty

Annotations +

PROPERTY

Characterist Description: owl:topObjectProp

Functional

Inverse functional

Transitive

Symmetric

Asymmetric

Reflexive

Irreflexive

Equivalent To +

SubProperty Of +

Inverse Of +

Domains (intersection) +

Ranges (intersection) +

Disjoint With +

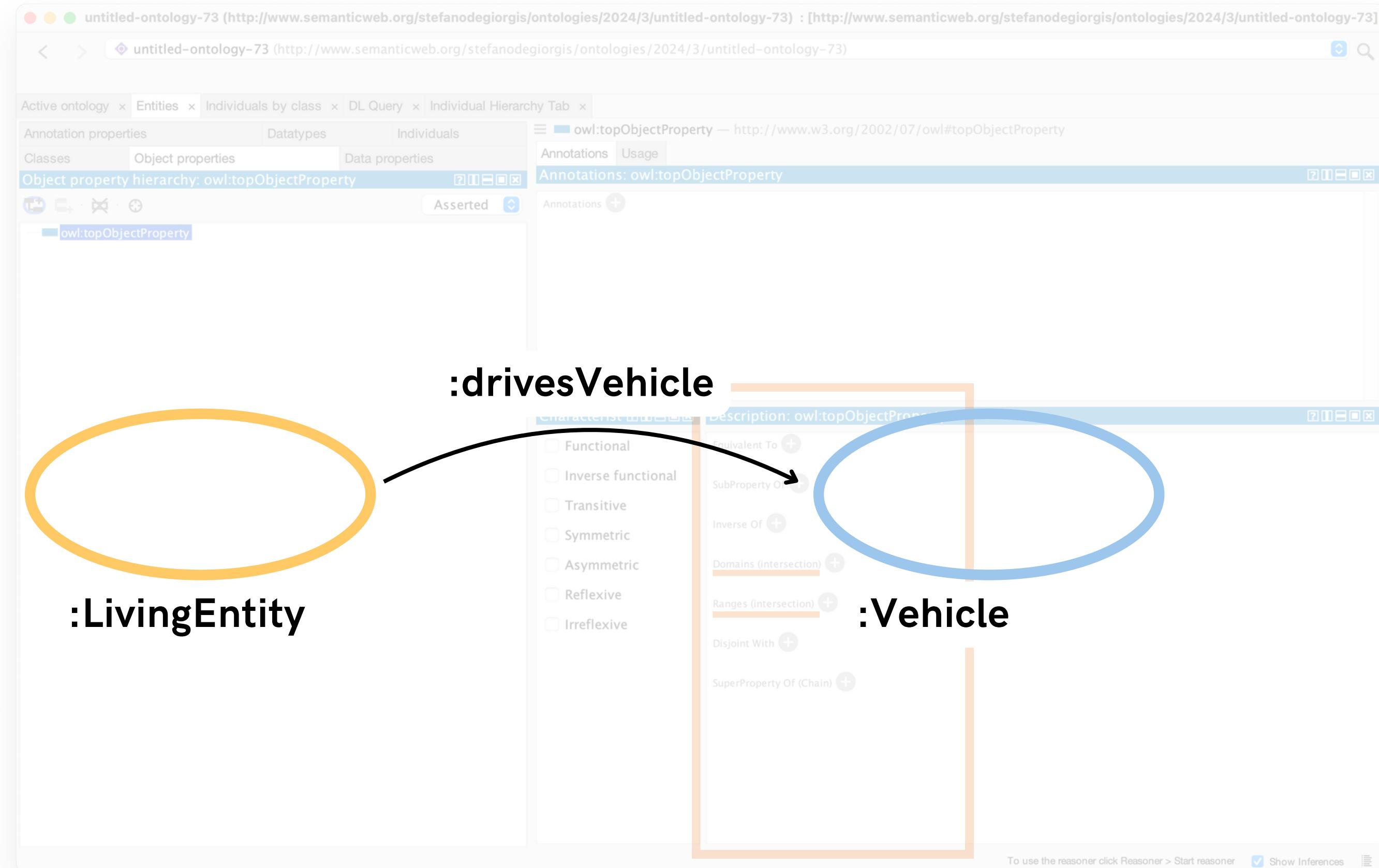
SuperProperty Of (Chain) +

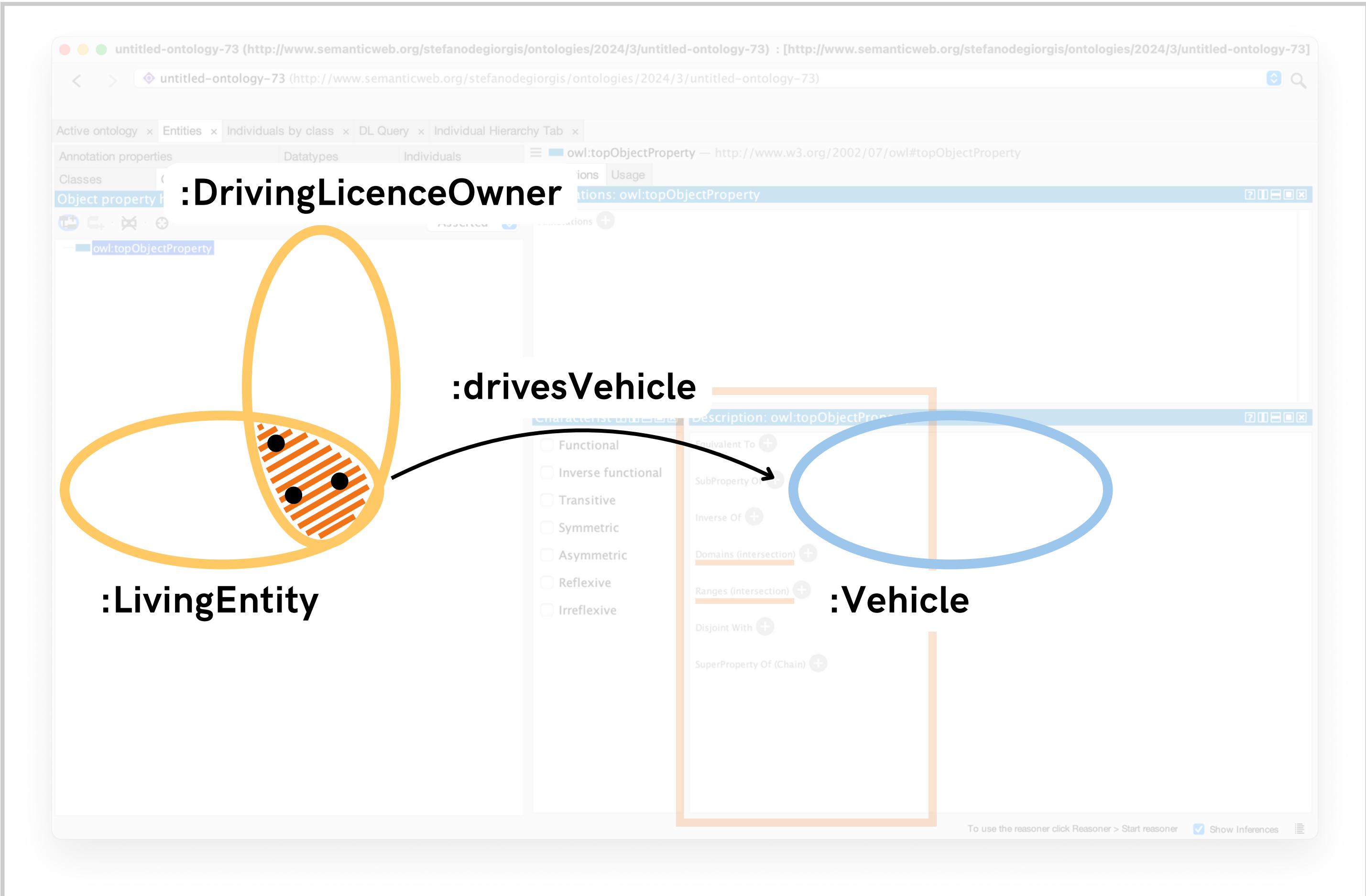
DOMAIN

RANGE

To use the reasoner click Reasoner > Start reasoner  Show Inferences

The diagram illustrates the concept of `owl:topObjectProperty`. It features two large ovals: one orange oval on the left labeled "DOMAIN" and one blue oval on the right labeled "RANGE". A black curved arrow points from the "DOMAIN" oval to the "RANGE" oval. Between the two ovals is a central vertical box with a light orange border, labeled "PROPERTY" at the top. This box contains several options: "Characterist" (which is expanded to show "Description: owl:topObjectProp"), "Functional", "Inverse functional", "Transitive", "Symmetric", "Asymmetric", "Reflexive", and "Irreflexive". Below these, there are several buttons with plus signs: "Equivalent To +", "SubProperty Of +", "Inverse Of +", "Domains (intersection) +", "Ranges (intersection) +", "Disjoint With +", and "SuperProperty Of (Chain) +". An arrow points from the "SubProperty Of +" button towards the "RANGE" oval.





untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>) : [<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>]

## Object property hierarchy

Characteristics: drive

Description: drivesVehicle

Asserted

owl:topObjectProperty

**drivesVehicle**

Functional

Inverse functional

Transitive

Symmetric

Asymmetric

Reflexive

Irreflexive

Equivalent To +

SubProperty Of +

Inverse Of +

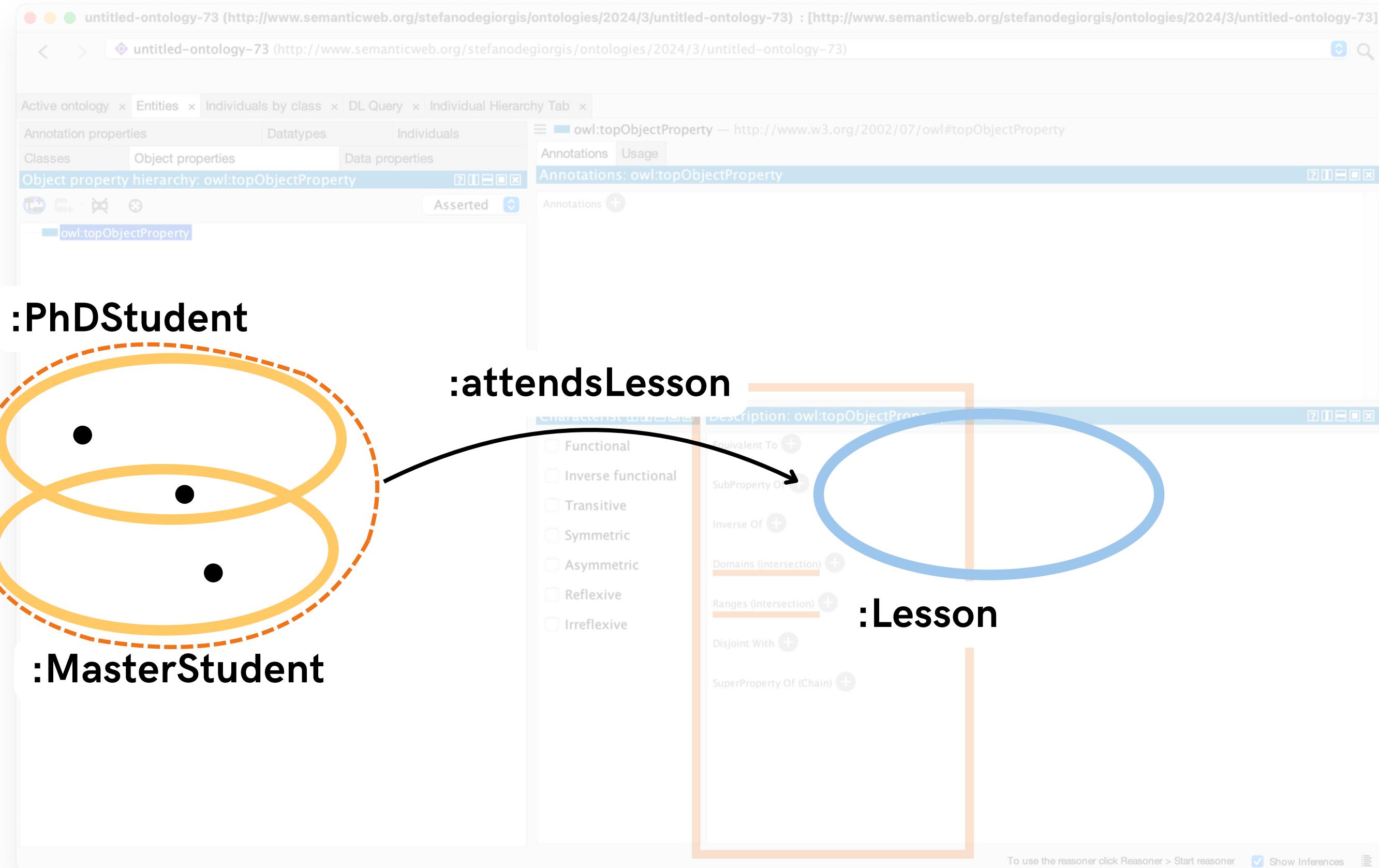
Domains (intersection) +

**DrivingLicenceOwner**  
and **LivingEntity**

Ranges (intersection) +

**Vehicle**

To use the reasoner click Reasoner > Start reasoner  Show Inferences



untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>) : [<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>]

## Object property hierarchy

Characteristics: asserted

Description: attendsLesson

Asserted

owl:topObjectProperty

attendsLesson

drivesVehicle

Functional

Inverse functional

Transitive

Symmetric

Asymmetric

Reflexive

Irreflexive

Equivalent To

SubProperty Of

Inverse Of

Domains (intersection)

MasterStudent or PhDStudent

Ranges (intersection)

Lesson

To use the reasoner click Reasoner > Start reasoner

Show Inferences

untitled-ontology-73 (<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>) : [<http://www.semanticweb.org/stefanodegiorgis/ontologies/2024/3/untitled-ontology-73>]

## Object property hierarchy

Characteristics: drive

Description: drivesVehicle

Asserted

owl:topObjectProperty

drivesVehicle

Functional

Inverse functional

Transitive

Symmetric

Asymmetric

Reflexive

Irreflexive

Equivalent To

SubProperty Of

Inverse Of

Domains (intersection)

DrivingLicenceOwner

LivingEntity

Ranges (intersection)

Vehicle

To use the reasoner click Reasoner > Start reasoner

Show Inferences

Data property hierarchy: hasStudentCardNumber

Annotations: hasStudentCardNumber

owl:topDataProperty

hasBirthDate

hasStudentCardNumber

Characteristics: hasStudentCardNumber

Description: hasStudentCardNumber

Functional

Equivalent To

SubProperty Of

Domains (intersection)

MasterStudent or PhDStudent

Ranges

xsd:string

Annotations

Asserted

Annotations

Characteristics

Description

Functional

Equivalent To

SubProperty Of

Domains (intersection)

MasterStudent or PhDStudent

Ranges

xsd:string

Data property hierarchy: hasBirthDa

Annotations: hasBirthDate

Annotations

Characteristics: hasI

Description: hasBirthDate

owl:topDataProperty

Functional

Equivalent To

SubProperty Of

Domains (intersection)

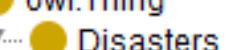
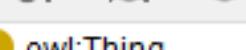
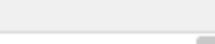
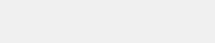
MasterStudent or PhDStudent

Ranges

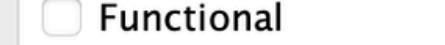
xsd:dateTime

hasBirthDate

hasStudentCardNumber

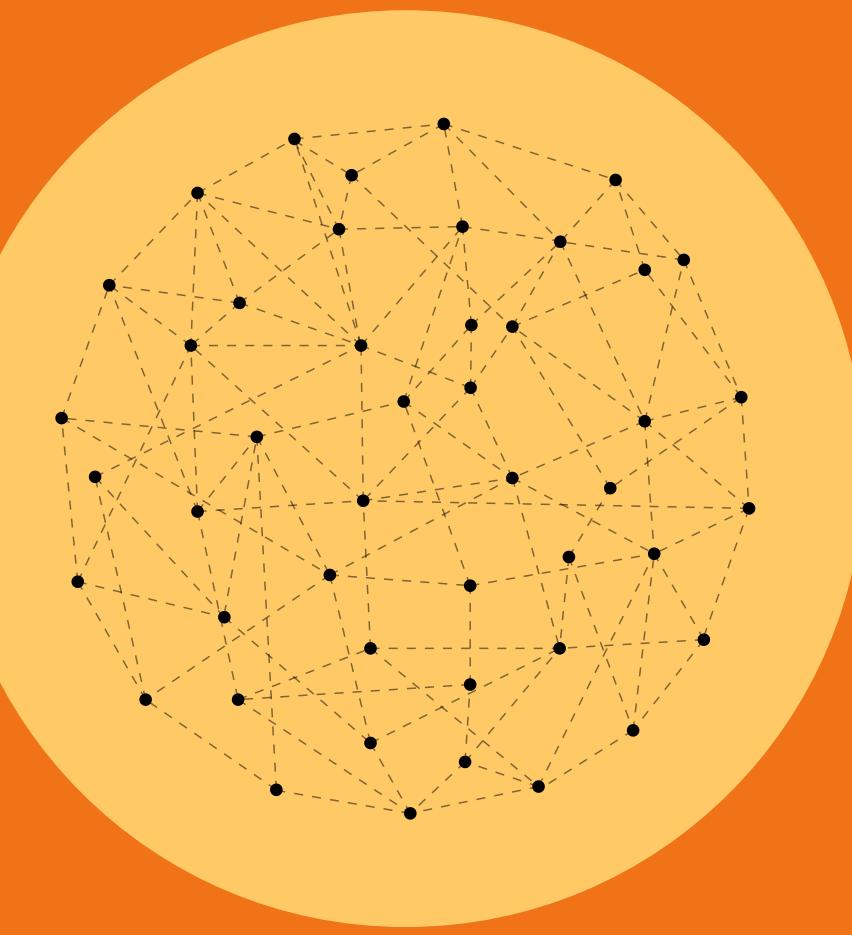
[Annotation properties](#) [Datatypes](#) [Individuals](#)[Classes](#)[Object properties](#)[Data properties](#)[Class hierarchy: Hazard](#)[owl:Thing](#)[Disasters](#)[EnvironmentalDisaster](#)[NaturalDisasters](#)[Events](#)[Hazard](#)[Earthquake](#)[NaturalPhenomenon](#)[Object](#)[environment](#)[society](#)[people](#)[property](#)[Annotations](#) [Usage](#)[Usage: Hazard](#)Show:  this  disjoint  named sub/superclasses

Found 6 uses of Hazard

[Earthquake](#)[Earthquake SubClassOf Hazard](#)[Description: Hazard](#)[Equivalent To](#) [damage some property](#)[SubClass Of](#) [Events](#)[Characteris](#)  Functional Inverse functional Transitive Symmetric Asymmetric Reflexive Irreflexive[Equivalent To](#) [SubProperty Of](#) [Inverse Of](#) [Domains \(intersection\)](#) [NaturalDisaster](#)[Ranges \(intersection\)](#) [Community](#)[Society](#)[Disjoint With](#) [SuperProperty Of \(Chain\)](#)

# Ontology visualization tools

- Protégé
- PoolParty suite
- TopBraid
- Neo4j suite



# PoolParty

The screenshot shows the PoolParty interface for managing semantic web concepts. The top navigation bar includes links for PROJECT, CORPORA, TOOLS, ADVANCED, and a search bar labeled "Search Thesaurus Concepts". The main content area displays the details for the concept "Cambridge".

**Left Sidebar (Jobs and Skills Demo Application):**

- Company Type (5)
- Job Roles (6)
- Location (3)
  - England (4)
    - Bristol (0)
    - Cambridge (0)
    - London (0)
    - Manchester (0)
  - Saudi Arabia (1)
  - United States of America (8)
    - California (1)
    - Massachusetts (1)
    - Minnesota (1)
    - New Jersey (0)
    - Pennsylvania (1)
    - Texas (1)
- Skills (6)
- Studies (3)
- Lists
- SKOS-XL Label
- Collections

# TopBraid

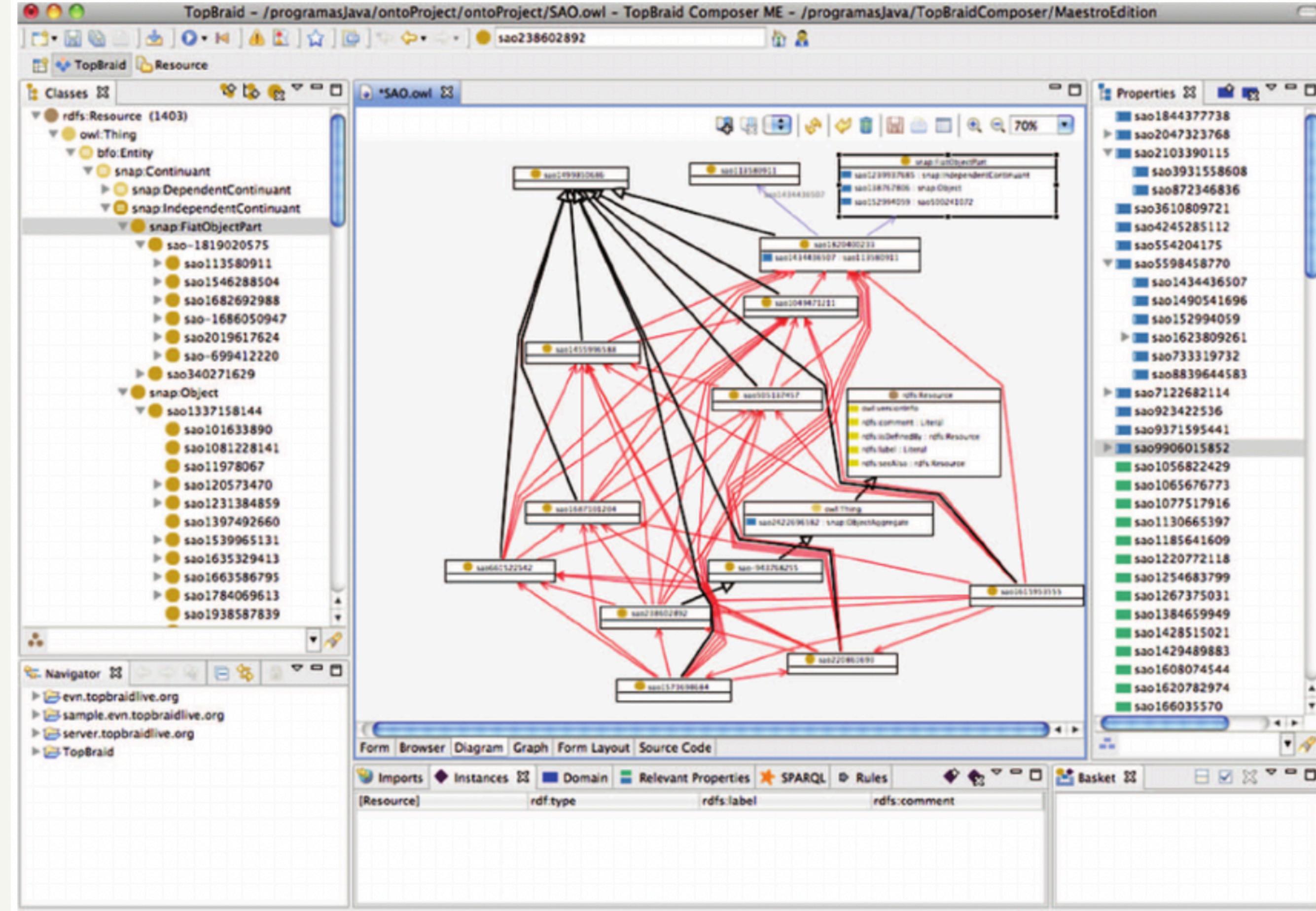


Image from: García-Peña, Francisco J., et al. "Using OWL-VisMod through a decision-making process for reusing OWL ontologies." Behaviour & Information Technology 33.5 (2014): 426-442.

# Neo4j

neo4j Docs

Docs ▾ Labs ▾ Get Help ▾ GraphAcademy ▾ Get Started Free

**Working with data**

- Model a dataset
- Connect data sources
- Import your data
- Query a database
- Visualize and explore
- Monitor data changes
- Use graph algorithms
- Create applications
- Integrate GenAI functions

**Fully-managed cloud (Aura)**

- Create an instance
- Import your data
- Monitor an instance
- Manage a deployment
- Backup, export, and restore

**Self-managed deployment**

- Installation
- Manage a database
- Import your data
- Deploy and manage a cluster
- Database internals

## Neo4j documentation

### Get started with Neo4j

Explore the capabilities of graph databases.

[Get started](#)

#### Deployment options

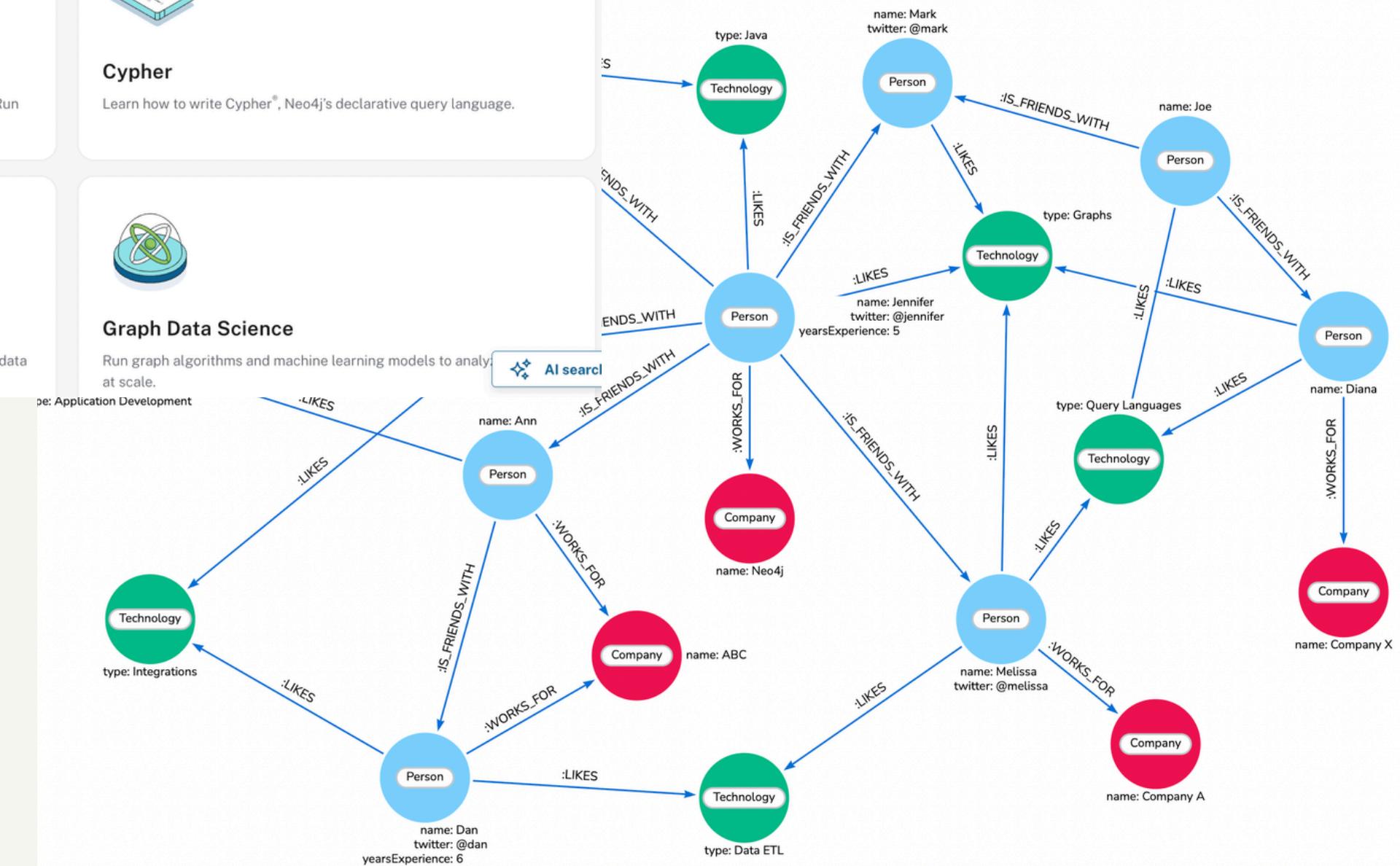
Choose from fully and self-managed local and cloud deployments. Run Neo4j on Docker or Kubernetes.

#### Cypher

Learn how to write Cypher®, Neo4j's declarative query language.

#### Graph Data Science

Run graph algorithms and machine learning models to analyze data at scale.



# Modeling Tools

---

