



Ontology Engineering

Knut Hinkelmann

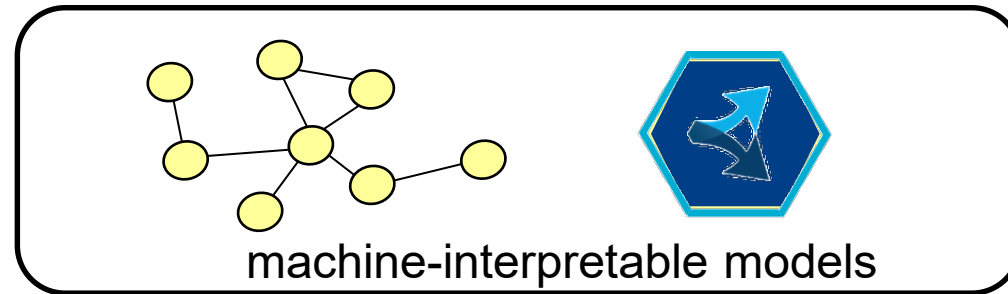


Knowledge-Representation and Reasoning

Reasoning



Knowledge Base



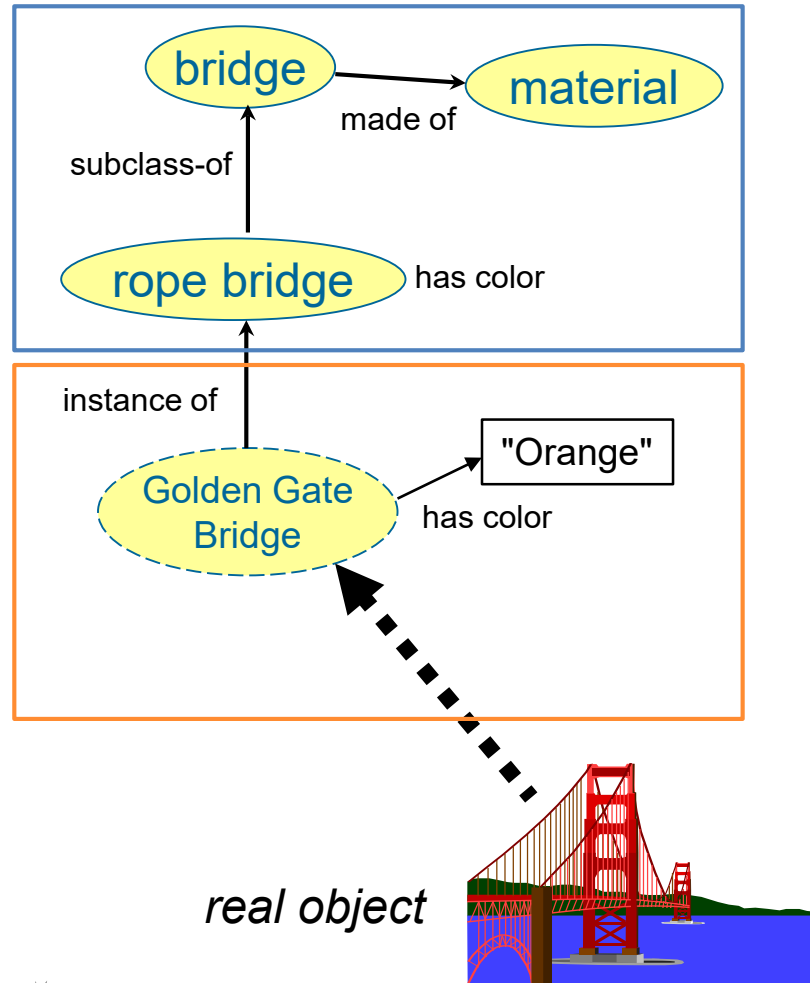
Reality



An Ontology – very informal

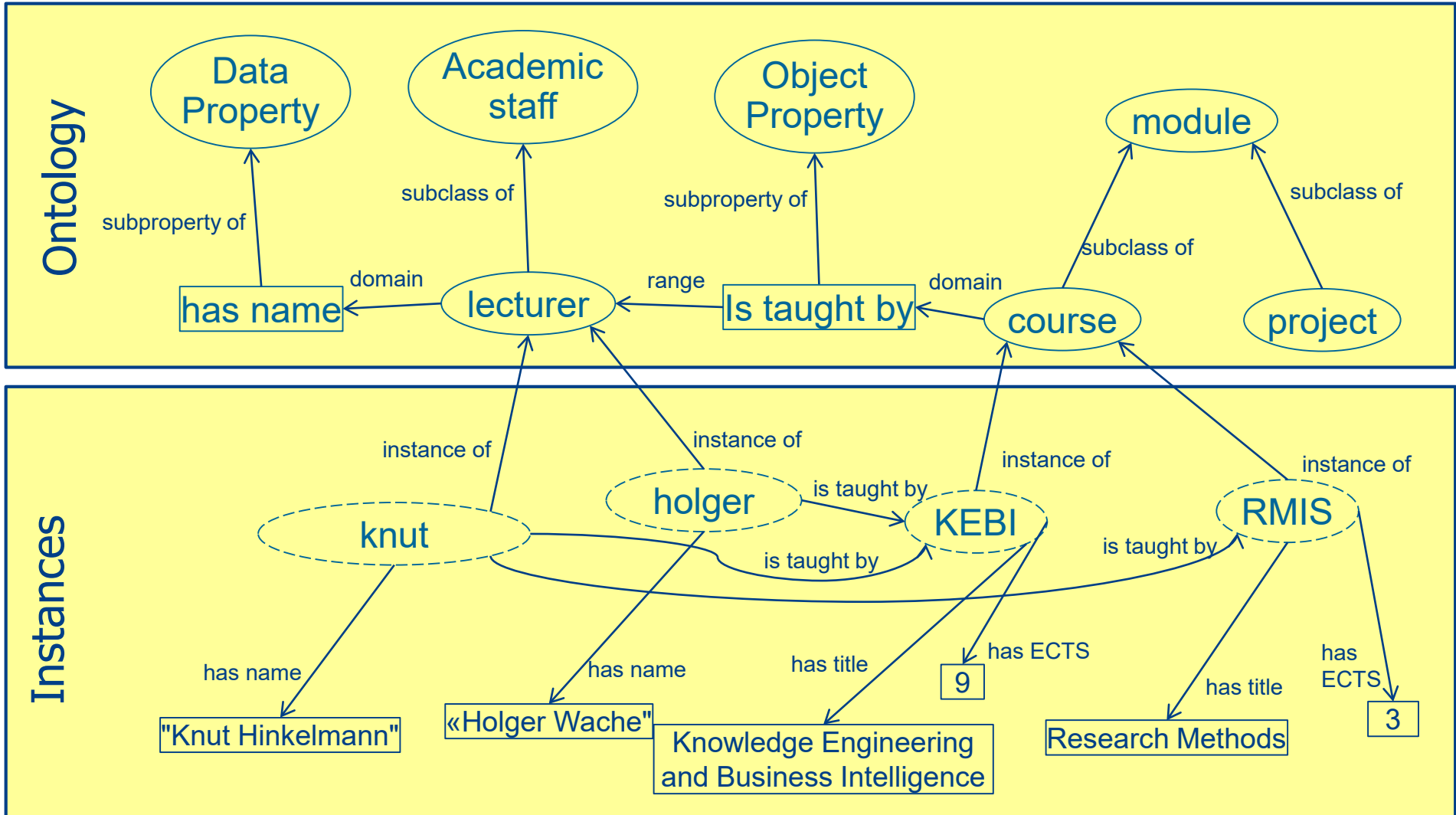
An ontology is a formal explicit description of concepts in a domain of discourse

- An **ontology** consists of
 - ◆ Concepts (Classes),
 - ◆ Relationships (Object Properties) between concepts
 - ◆ Attribute (Data Properties) of concepts
 - ◆ Constraints that hold between/for the concepts,as a representation of a particular domain
- An ontology together with a set of individual instances constitutes a **knowledge base**



ontology engineering
is
knowledge engineering

Example of an Ontology



Ontology Representation Formalisms

■ Representations of Ontologies

◆ ***RDF(S)***

Our focus

◆ OWL

◆ Neo4J

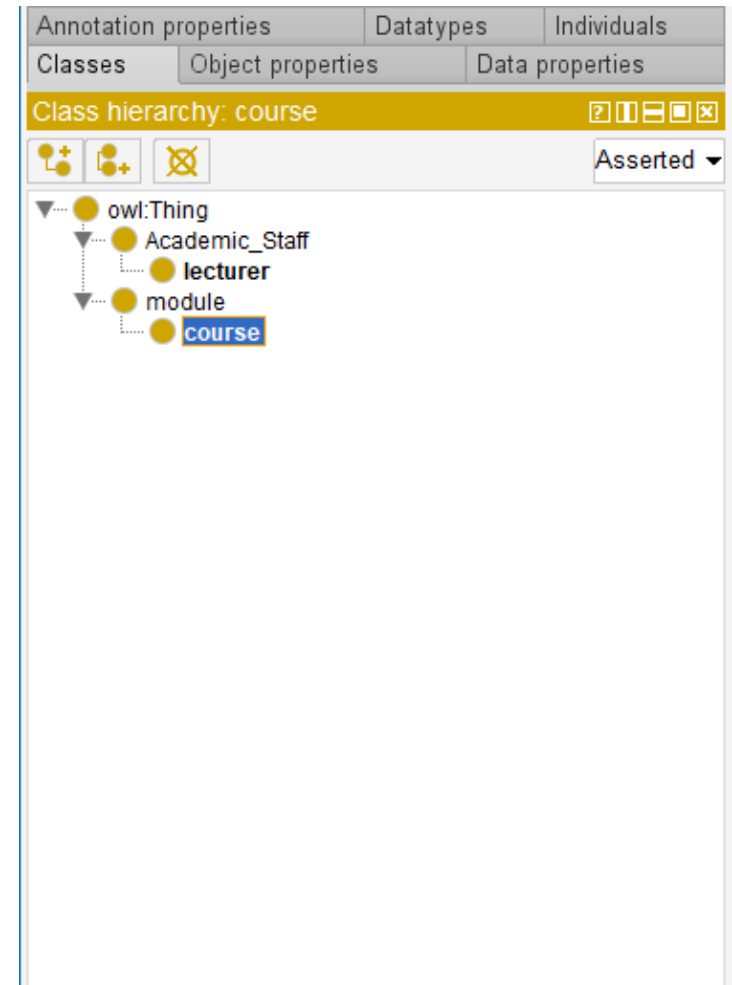
◆ ...

Strategy

- Defining classes in the ontology
- Arranging the classes in a taxonomic (subclass-superclass) hierarchy
- Defining properties and describing allowed values for the properties
- Creating instances and filling the values for properties

Define Classes and Class Hierarchy

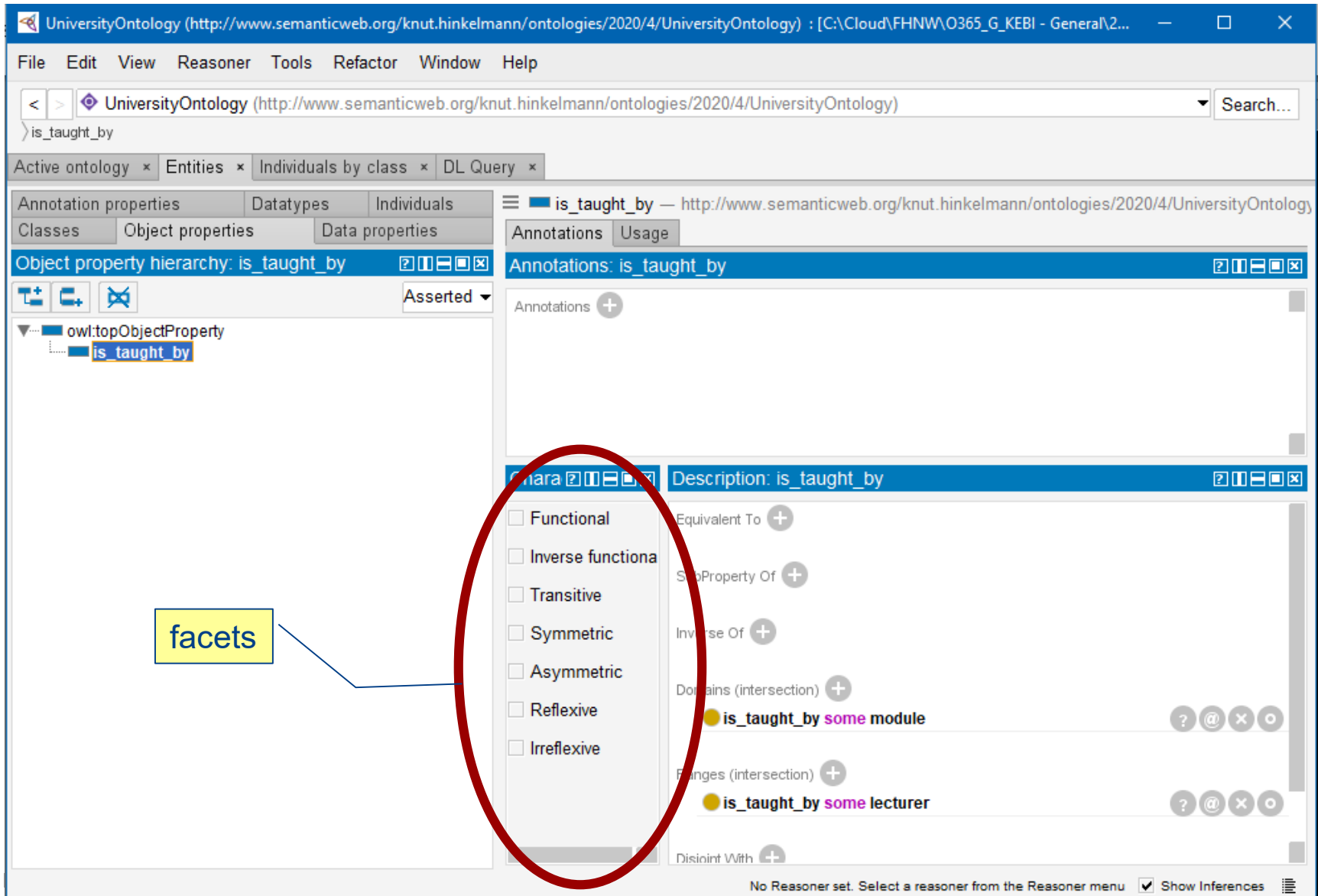
- There are several approaches
 - ◆ Top-down: Start with the most general concept, and work your way down
 - ◆ Bottom-up: Start with the most specific, and work your way up
 - ◆ Combination



Define Properties of Classes

- Describe the internal structure of concepts
 - ◆ Data Properties: Attributes
 - Range are data types like String, Integer, ...
 - ◆ Object Properties: Relations to other concepts
 - Range are Classes
- Describe facets: Characteristics of Properties
- Inheritance to Subclasses

Object Property



The screenshot shows the Protégé interface for the 'UniversityOntology'. The 'is_taught_by' property is selected, and its facets are being configured. The facets list includes:

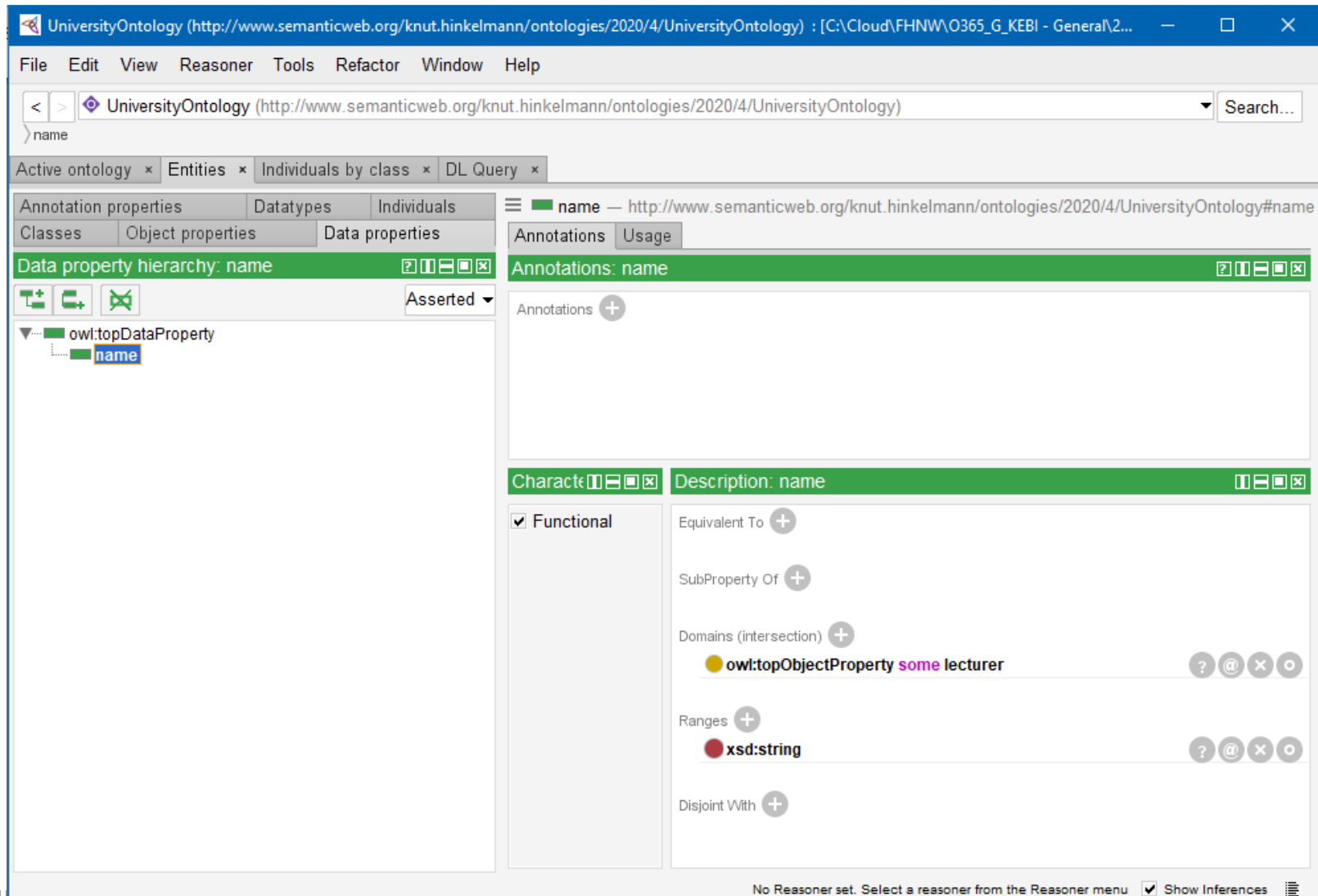
- Functional
- Inverse functional
- Transitive
- Symmetric
- Asymmetric
- Reflexive
- Irreflexive

The 'Description: is_taught_by' panel shows two facets:

- is_taught_by some module**
- is_taught_by some lecturer**

A yellow box labeled 'facets' points to the facets list.

Data Property

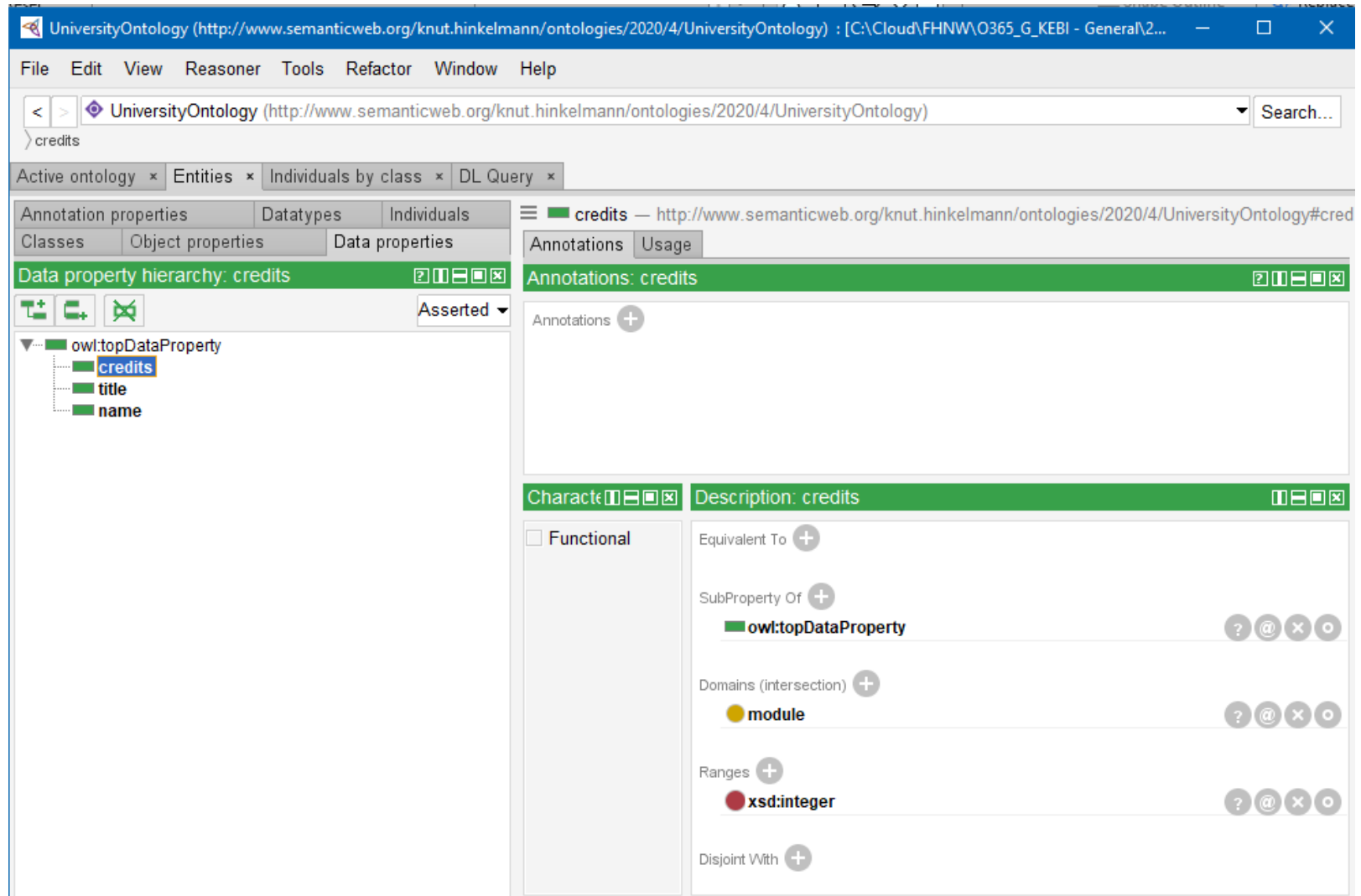


The screenshot shows the Protege interface for editing the 'name' data property in the 'UniversityOntology'.

- Left Panel (Data property hierarchy):** Shows a tree view under 'owl:topDataProperty' with the property 'name' selected.
- Top Panel (Annotations):** Shows the URI for the property: `http://www.semanticweb.org/knut.hinkelmann/ontologies/2020/4/UniversityOntology#name`.
- Right Panel (Description):** Shows the configuration for the property:
 - Functional**
 - Equivalent To:** +
 - SubProperty Of:** +
 - Domains (intersection):** +
 - `owl:topObjectProperty some lecturer`
 - Ranges:** +
 - `xsd:string`
 - Disjoint With:** +

At the bottom, the status bar indicates: "No Reasoner set. Select a reasoner from the Reasoner menu" and "Show Inferences" is checked.

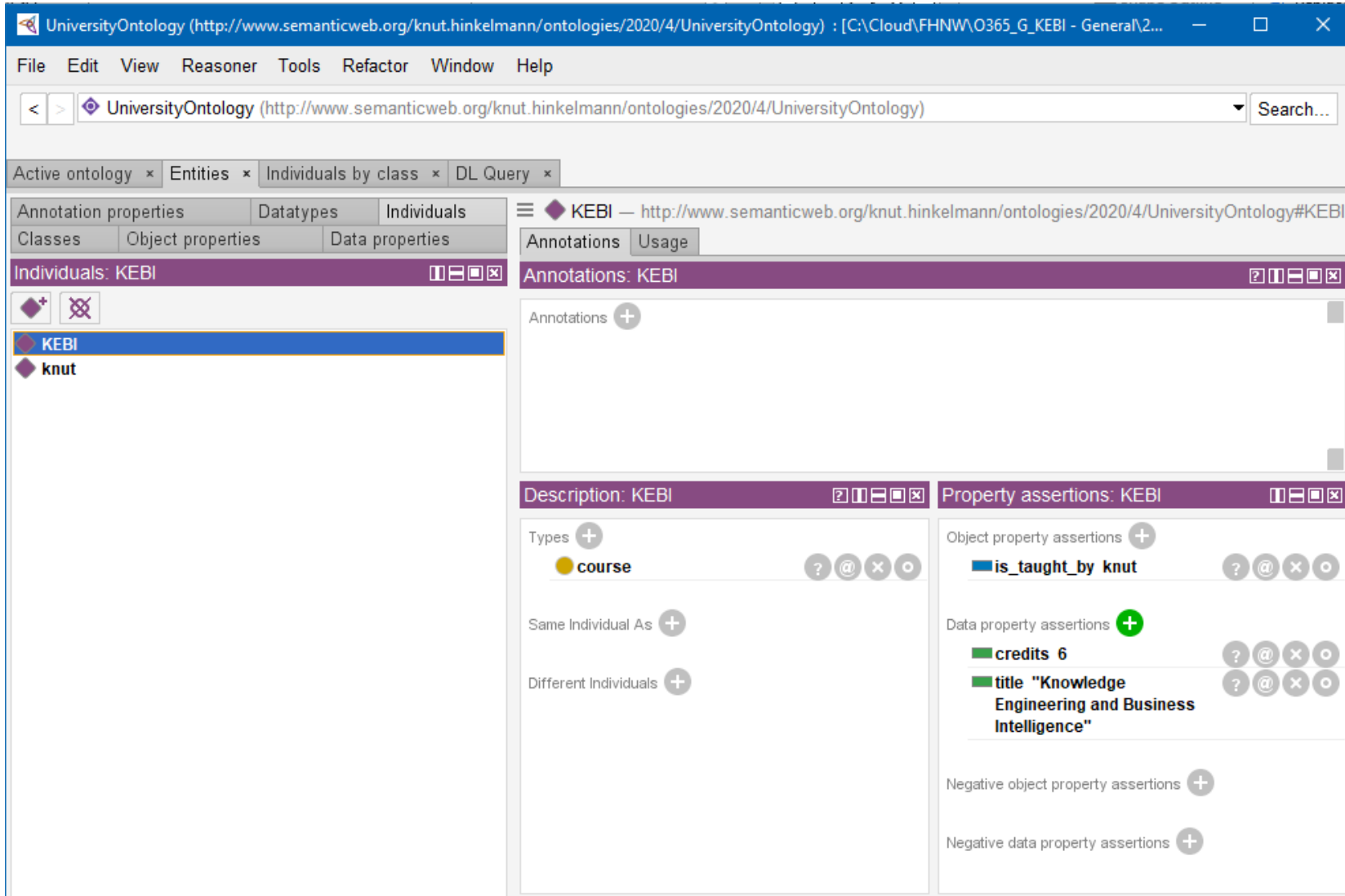
Data Property



The screenshot shows the UniversityOntology web interface. The browser address bar displays the URL: `http://www.semanticweb.org/knut.hinkelmann/ontologies/2020/4/UniversityOntology`. The interface includes a menu bar (File, Edit, View, Reasoner, Tools, Refactor, Window, Help) and a search bar. The main content area is divided into several panels:

- Data property hierarchy: credits:** A tree view showing the hierarchy of data properties. The root is `owl:topDataProperty`, which has three children: `credits` (highlighted), `title`, and `name`.
- Annotations: credits:** A panel for managing annotations for the `credits` property. It currently shows no annotations.
- Description: credits:** A panel for defining the properties of the `credits` data property. It includes a **Functional** checkbox (unchecked) and several configuration sections:
 - Equivalent To:** A plus sign (+) to add equivalent properties.
 - SubProperty Of:** A plus sign (+) to add subproperties. The current subproperty is `owl:topDataProperty`.
 - Domains (intersection):** A plus sign (+) to add domains. The current domain is `module`.
 - Ranges:** A plus sign (+) to add ranges. The current range is `xsd:integer`.
 - Disjoint With:** A plus sign (+) to add disjoint properties.

Instances



The screenshot shows a web browser window displaying the 'UniversityOntology' (http://www.semanticweb.org/knut.hinkelmann/ontologies/2020/4/UniversityOntology). The interface includes a menu bar (File, Edit, View, Reasoner, Tools, Refactor, Window, Help) and a search bar. The main content area is divided into several panels:

- Active ontology:** UniversityOntology
- Entities:** Individuals by class
- Annotation properties, Datatypes, Individuals, Classes, Object properties, Data properties:** Navigation tabs for the ontology structure.
- KEBI — http://www.semanticweb.org/knut.hinkelmann/ontologies/2020/4/UniversityOntology#KEBI:** The main ontology view, showing:
 - Annotations: KEBI:** An empty list of annotations.
 - Description: KEBI:**
 - Types:** A list containing 'course'.
 - Same Individual As:** A button to add relationships.
 - Different Individuals:** A button to add relationships.
 - Property assertions: KEBI:**
 - Object property assertions:** A list containing 'is_taught_by knut'.
 - Data property assertions:** A list containing 'credits 6' and 'title "Knowledge Engineering and Business Intelligence"'. A green plus sign indicates that more data property assertions can be added.
 - Negative object property assertions:** A button to add negative assertions.
 - Negative data property assertions:** A button to add negative assertions.

Querying

- Query Language: SPARQL
 - ◆ Variables: ?x
- Elements are denoted as URI
 - ◆ Prefixes for Abbreviations
 - Example: PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
- Sample query: Select all lecturers:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX uo: <http://www.semanticweb.org/knut.hinkelmann/ontologies/2020/4/UniversityOntology#>
SELECT ?subject
      WHERE { ?subject rdf:type uo:lecturer }
```

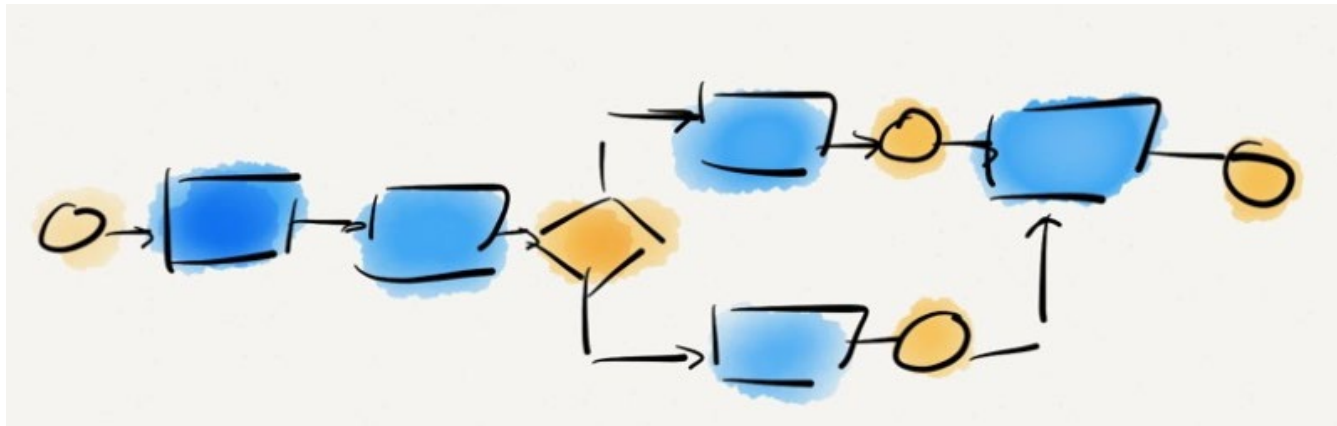
Exercise

- Add new classes: project
 - ◆ A project has a supervisor
 - ◆ Master Thesis is a project
 - ◆ Supervisor is a lecturer
 - ◆ A project is performed by a student
- Add new instances

Exercise: Modeling Process Knowledge in an Ontology

Exercise: Modeling Process Knowledge in an Ontology

- We create a knowledge base for process knowledge
 - ◆ Define the ontology
 - ◆ Represent knowledge of a process



Ontology Development 101

- 1 Determine the domain and scope of the ontology
- 2
 - Consider reusing existing ontologies
- 3
 - Enumerate important terms
- 4
 - Define classes and class hierarchy
- 5
 - Define the data and object properties of classes
- 6
 - Define the facets of properties
- 7
 - Create instances

Determine the domain and scope of the ontology

- What is the domain that the ontology will cover?
- For what we are going to use the ontology?
- For what types of questions the information in the ontology should provide answers? → Competency questions
- Who will use and maintain the ontology?

Competency Questions

- One of the ways to determine the scope of the ontology is to sketch a list of questions that a knowledge base based on the ontology should be able to answer (Gruninger and Fox 1995)
 - ◆ Does the ontology contain enough information to answer these types of questions?
 - ◆ Do the answers require a particular level of detail or representation of a particular area?

- Exercise: We want to represent knowledge about
 - ◆ the process flow
 - ◆ Responsibilities for tasks
- Competency Questions:
 - Who executes task X?
 - Which task is executed after task X?
 - When can task X start?
- Sample process:

The waiter serves the beverages. Then the waiter serves the food. When the guests are finished, the waiter presents the bill.

Consider reusing existing ontologies

- It is always worth considering what others have done, and check if their work can be refined and extended for our particular domain and task
- Mandatory if the system needs to interact with other applications that have already committed to particular ontologies or controlled vocabularies

Enumerate important terms in the ontology

- What are the terms we would like to talk about?
- What are their properties?
- What would we like to say about those terms?



<https://www.menti.com/dkpew59hq4>

Define Classes and Class Hierarchy

- Several possible approaches in developing a class hierarchy:
 - ◆ Top-down: General to specific concepts
 - ◆ Bottom-up: Specific to general concepts
 - ◆ Combination: Salient to general and specific concepts
- Classes for
 - ◆ Modeling Objects
 - ◆ Relations

Define the properties of classes

- Describe the internal structure of concepts
 - ◆ Data Properties: Attributes
 - Range are data types like String, Integer, ...
 - ◆ Object Properties: Relations to other concepts
 - Range are Classes
- Inheritance to Subclasses

Create Instances

- Model a business process in an ontology

The waiter serves the food. When the guest are finished, the waiter presents the bill.

Conclusion

Modeling Business Processes as ontologies is not adequate for business people

→ Graphical Models

