

Terminology & Ontology in the Digital Age



Renmin University (China)
26-30 October 2020

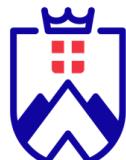
Monday 26 October	14:00-17:00 pm
Tuesday 27 October	14:00-17:00 pm
Wednesday 28 October	19:00-22:00 pm
Thursday 29 October	19:00-22:00 pm
Friday 30 October	19:00-22:00 pm

Beijing Time

Prof. Christophe Roche

Liaocheng University (China)
Savoie Mont-Blanc University (France)

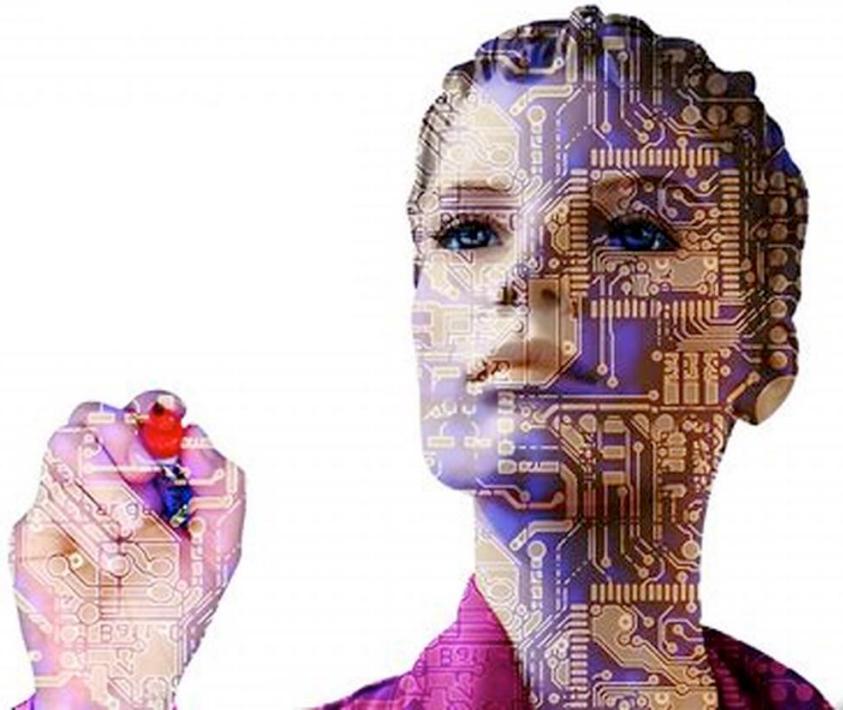
<http://christophe-roche.fr/>



Roche
1937

Contents

1. A Matter of Meaning (3h)
2. Terminology & ISO Standards (3h)
3. Ontology & W3C Standards (3h)
- 4. Software Environments (CmapTools, Protégé, Tedi) (3h)**
5. Applications (Smart City & Digital Humanities) (3h)

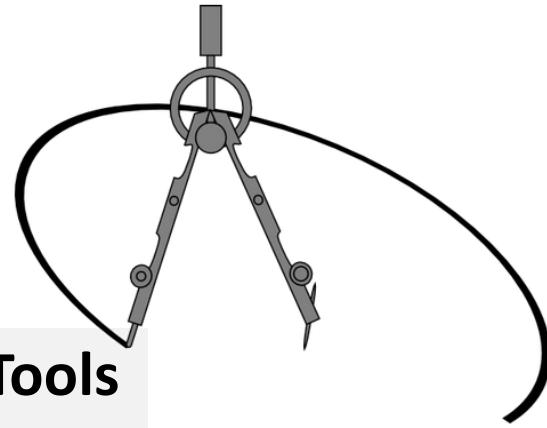


Tools & Environments

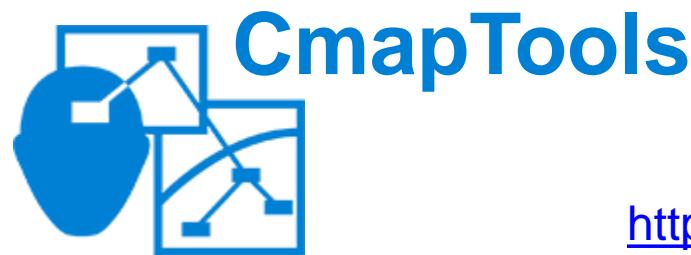


- **Graphical Tools:**
- **Formal Tool:**
- **Dedicated Tools:**

- CmapTools
- Protégé
- Tedi



Downloads



<http://cmap.ihmc.us/>



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

<http://cmap.ihmc.us/>



The Institute for Human & Machine Cognition (IHMC) pioneers technologies aimed at leveraging and extending human capabilities. Explore our research areas to see what we're doing.

FEATURED RESEARCH

- Language processing
- Computational & Philosophical Foundations
- Intentions, Beliefs & Trust



Concept maps are graphical tools for organizing and representing knowledge.

They include **concepts**, usually enclosed in circles or boxes of some type, and **relationships** between concepts indicated by a connecting line linking two concepts.



Epistemological Principles

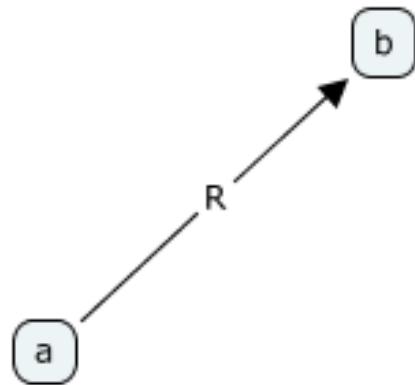
Concept: *Concept as a perceived regularity in events or objects, or records of events or objects, designated by a label.*

Proposition: *Propositions are statements about some object or event in the universe, either naturally occurring or constructed. Propositions contain two or more concepts connected using linking words or phrases to form a meaningful statement.*

Representation

Concept: *Node*

Proposition: *labeled link*





Epistemological Principles

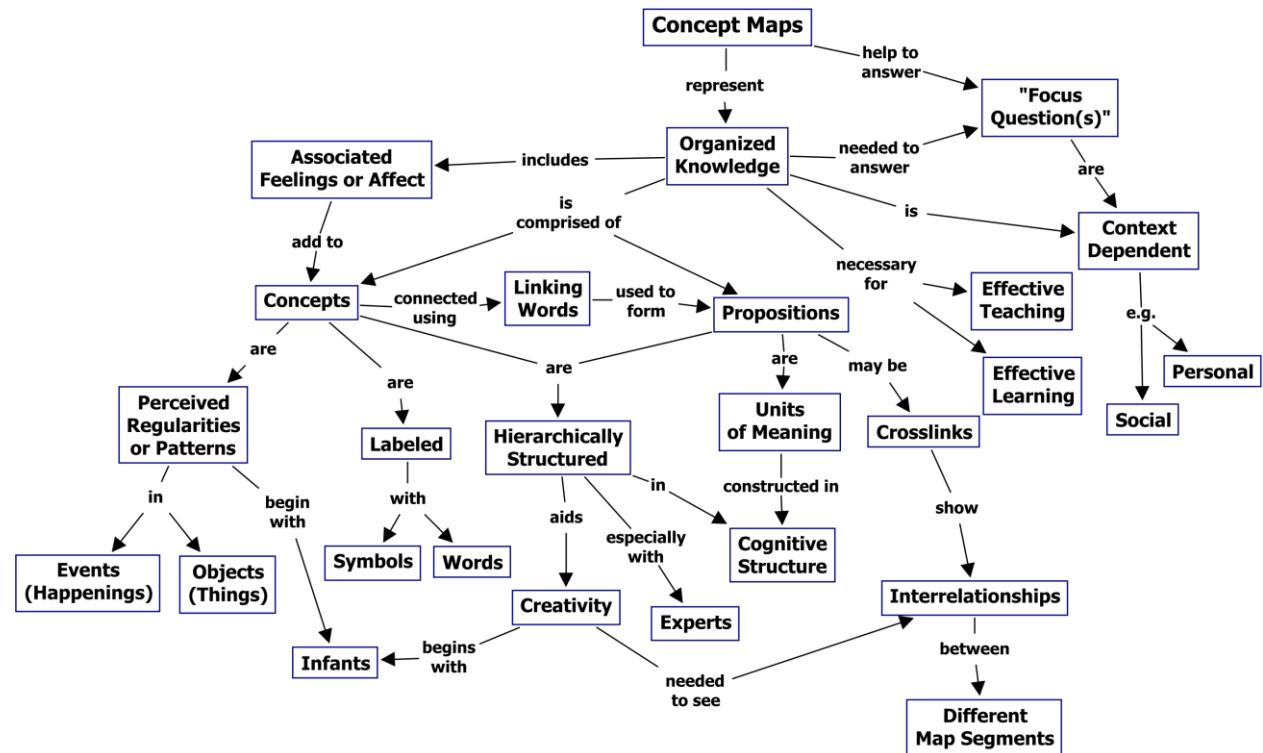
Concept: Concept as a perceived regularity in events or objects, or records of events or objects, designated by a label.

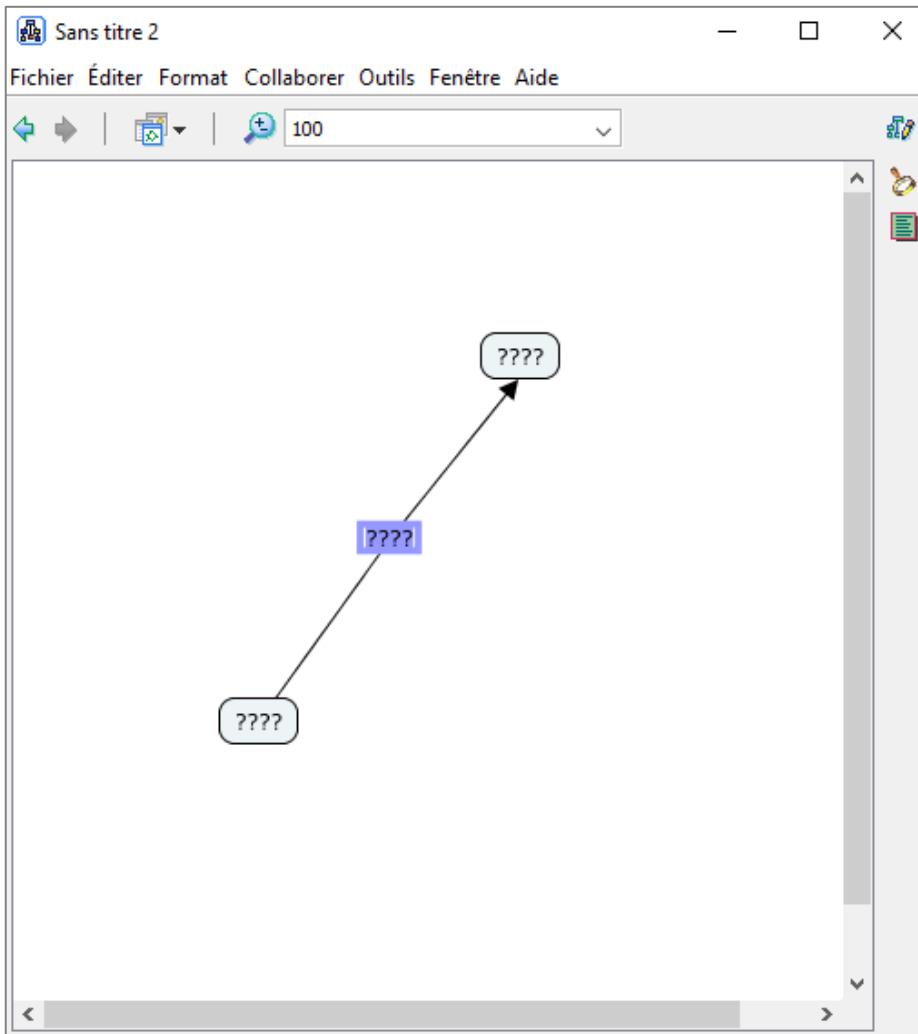
Proposition: Propositions are statements about some object or event in the universe, either naturally occurring or constructed. Propositions contain two or more concepts connected using linking words or phrases to form a meaningful statement.

Representation

Concept: Node

Proposition: labeled link





Do it





How to Represent the Categories of Thought?

Langue of Intellection

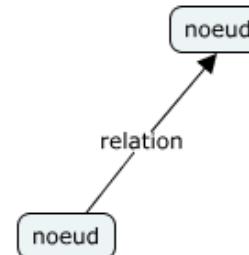
Categories of Thought :

- Object
- Characteristic :
 - essential
 - descriptive
- Concept
- Class
- Relation :
 - generic
 - partitive
 - functional
 - associative
 -

Representation Language

Language Categories of CmapTools

- Node
- Labelled link (binary relationships)



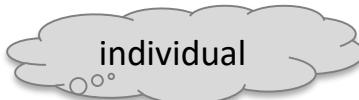


Semiotic code

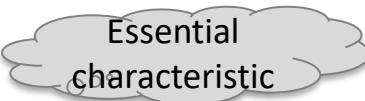
each category of thought will be represented differently



<Concept>



individual



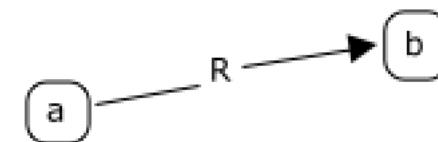
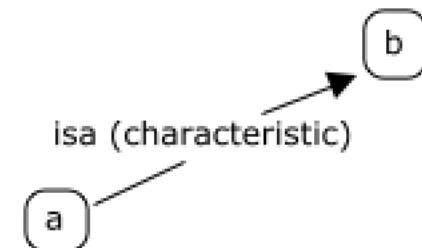
/characteristic/

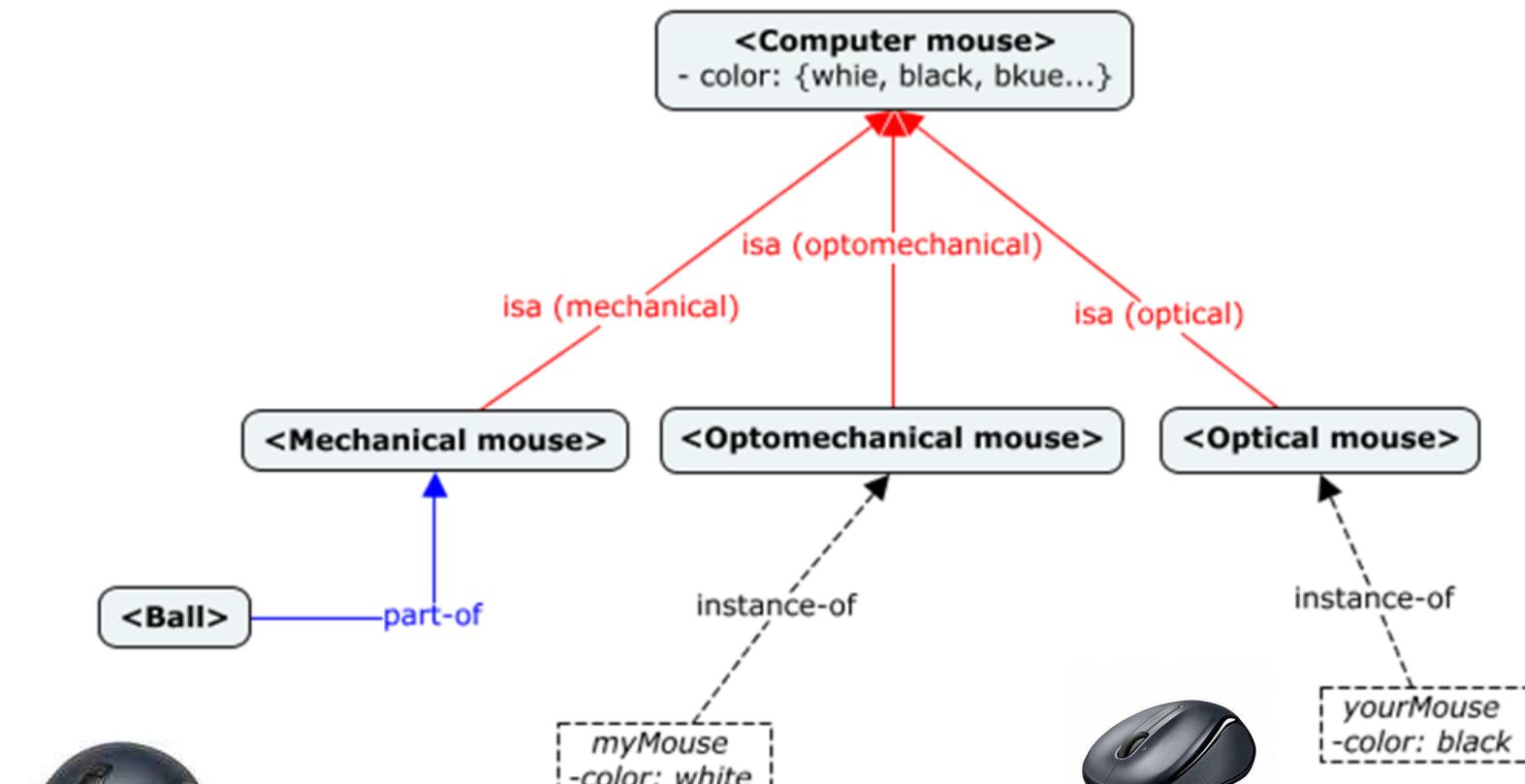


$R(a,b)$



$at(e,v)$





No verification

Exercise



stool

armchair

椅子



chaise

pouf

fauteuil

banc

chair

canapé

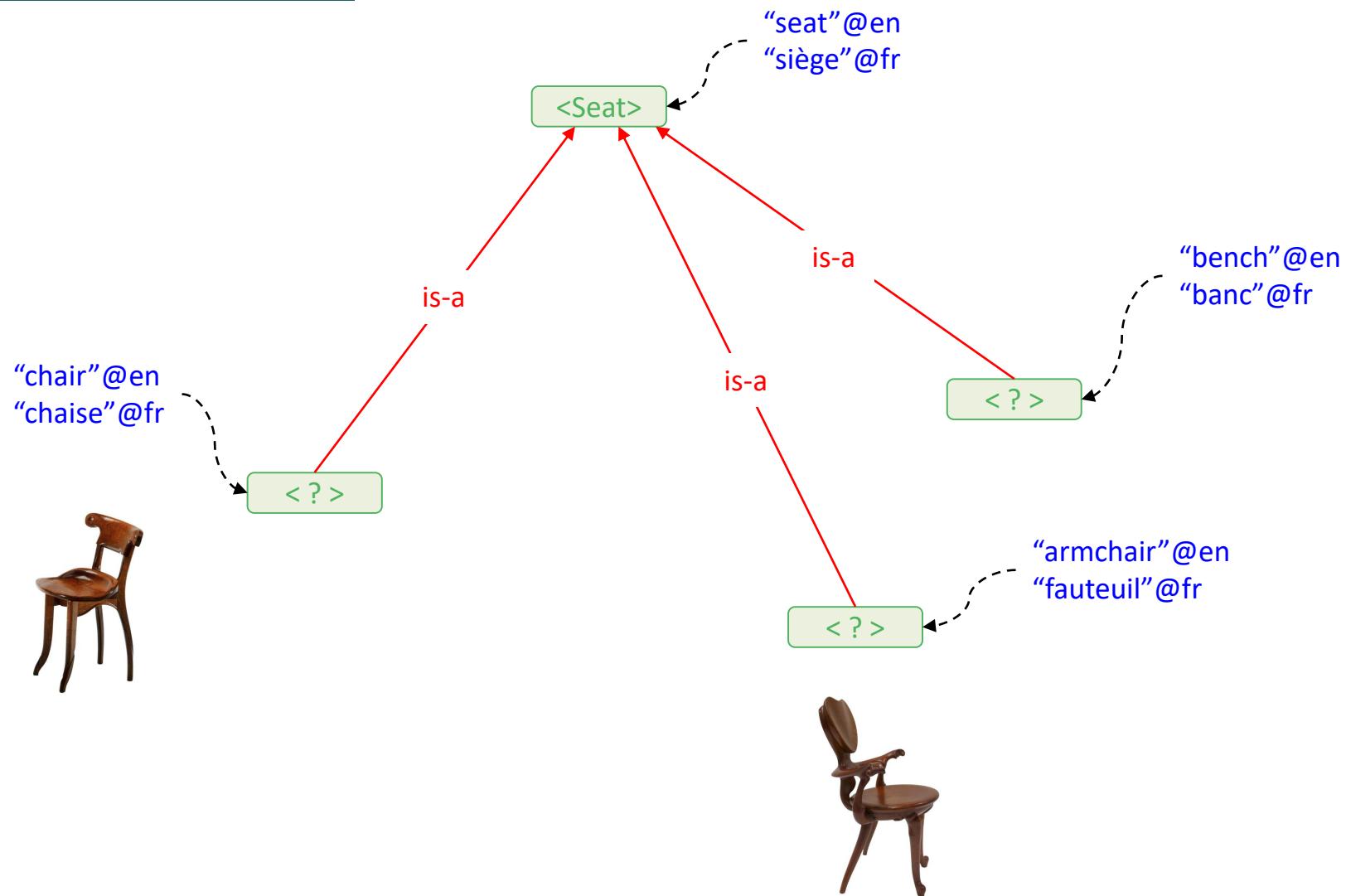
couch

长椅



tabouret

bench



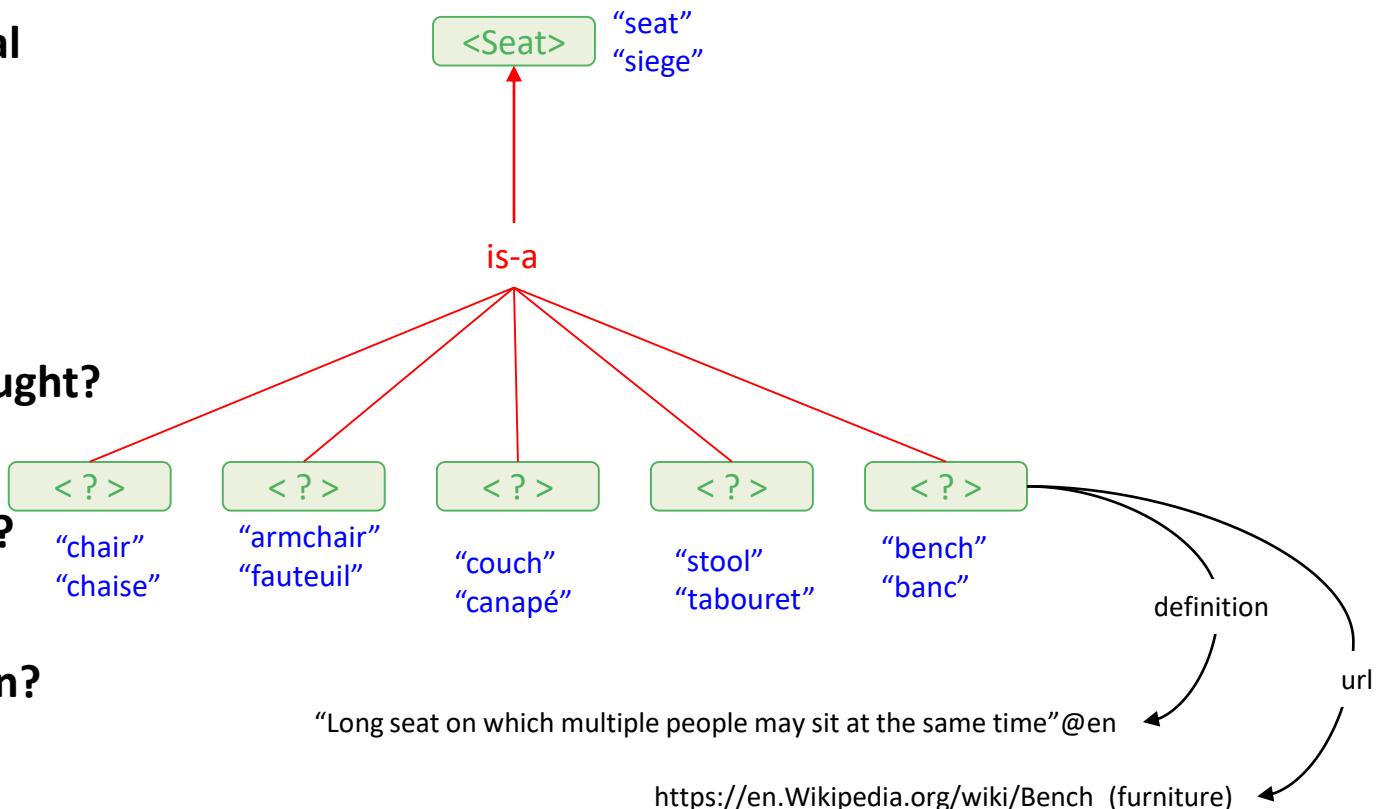
✓ **Easy to use**



✓ **Human Readable**

✓ **Semi-Formal**

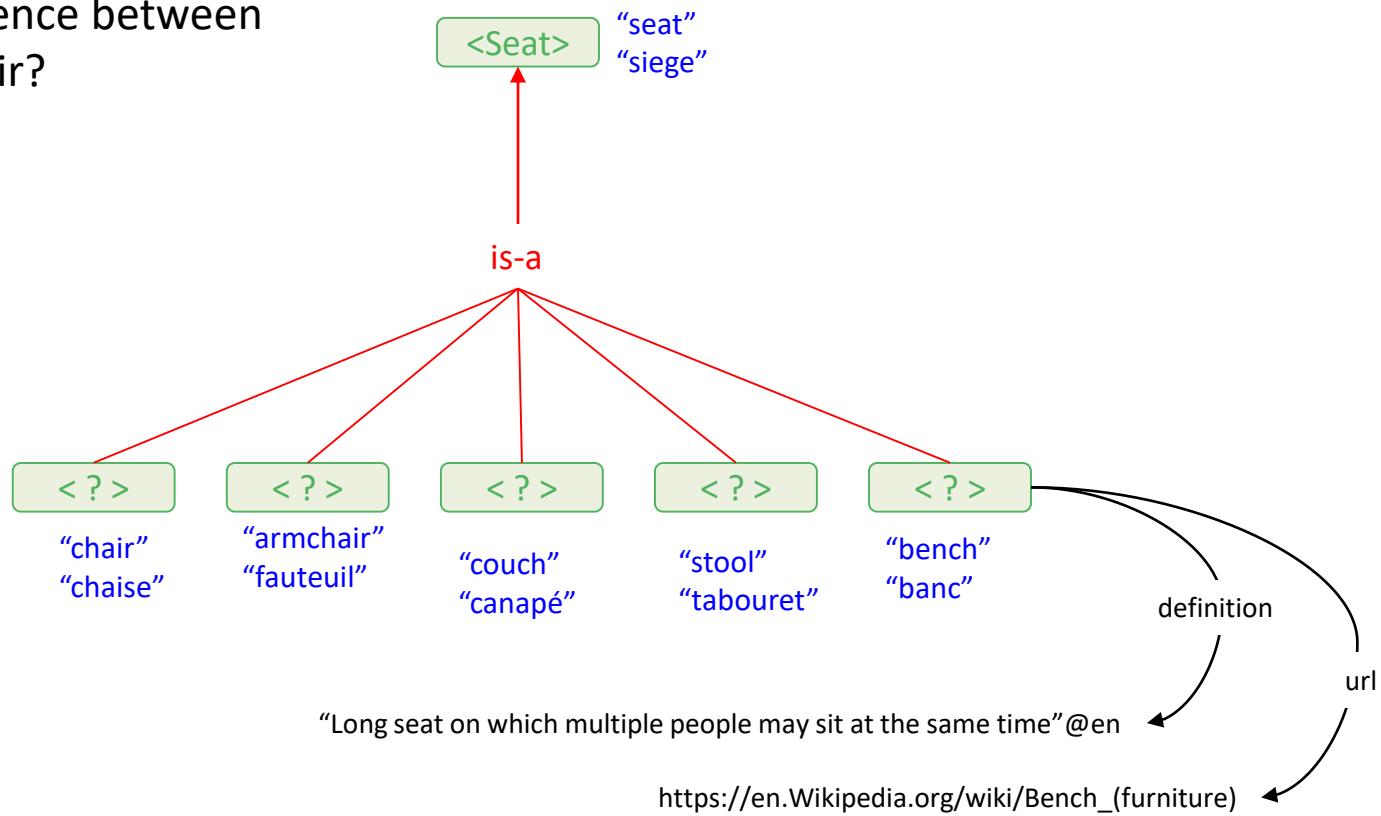
- **Categories of thought?**
- **Methodology?**
- **Formal definition?**
- **Coherency?**
- **Operationalization?**



(Formal) Definition?

Is defining Chair as a subclass of Seat sufficient?

What is the difference between
Chair and Armchair?



Array of differences

Objects	Concepts	Axis of analysis		Axis of analysis		Axis of analysis		Axis of analysis		Terms	
Objects	Concepts	for one person	several persons	with feet	without feet	with back	without back	with arms	without arms	Designations (English)	Designations (French)
	<Seat 1 person with feet with back without arms>	X		X		X			X	"chair"	"chaise"
	<Seat 1 person with feet with back with arms>	X		X		X		X		"armchair"	"fauteuil"
	<Seat 1 person with feet without back without arms>	X		X			X		X	"stool"	"tabouret"
	<Seat several persons with feet with back with arms>		X	X		X		X		"couch"	"canapé"
	<Seat several persons with feet without back without arms>		X	X			X		X	"bench"	"banc"

"chair" : Seat for one person with feet and back without arms.

↳ <Seat for one person with feet with back without arms>

::= <Seat> + /for one person/ + /with feet/ + /with back/ + /without arms/



"armchair" : Seat for one person with feet and back with arms.

↳ <Seat for one person with feet with back without arms>

::= <Seat> + /for one person/ + /with feet/ + /with back/ + /with arms/



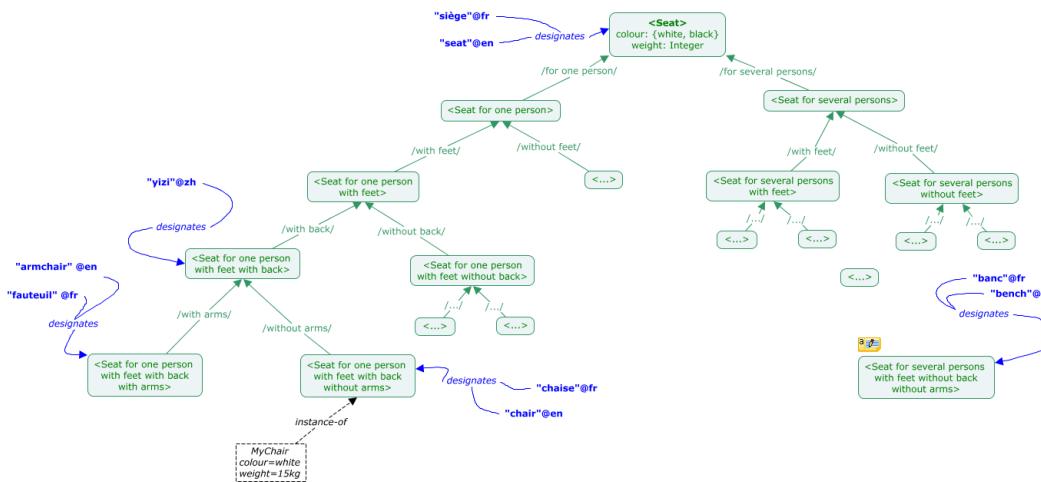
"bench" : Seat for several persons with feet, without back, and without arms.

↳ <Seat for one person with feet with back without arms>

::= <Seat> + /for one person/ + /with feet/ + /with back/ + /with arms/

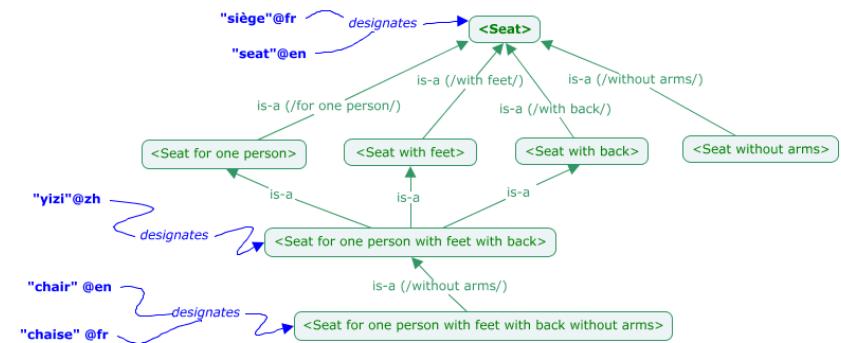
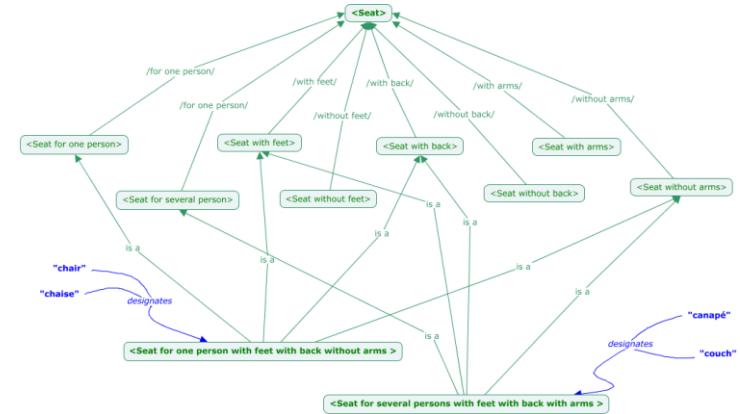


Graphical Notation



They are all equivalent.

A given graph is one of several possible representations of the same conceptual system



<Concept> = { essential characteristics}



protégé



protégé

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

Protégé is a free, open source ontology editor written in Java and developed at Stanford University.

More than 300,000 users are registered.

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.



Protégé provides:

- a graphic user interface to define ontologies;
- deductive classifiers to validate that models are consistent and to infer new information based on the analysis of ontology.

Class hierarchy 'Vessel_with_handle(s)_for_mixing_wine_with_water_with_Handle(s)'

- owl:Thing
- 'with_handle(s)'
- 'Vessel_for_mixing_wine_with_water_with_Handle(s)'
- 'Vessel_for_storing_and_transport_with_neck_with_Handle(s)_with_two_handles'
- hydrie
- column-like handle
- for_drinking
- for_mixing_wine_with_water
- for_perfume
- for_pouring
- for_ritual
- for_storing_and_transport
- liquid
- material
- oil
- part
- person
- place
- solid
- upward_curling_handle
- upward_curling_handles_placed_high_on_the_body
- upward_curling_handles_placed_low_on_the_body
- vessel
- Vessel
- 'Vessel_for_mixing_wine_with_water_with_Handle(s)_for_mixing_wine_with_water'
- 'Vessel_for_mixing_wine_with_water_with_Handle(s)_for_storing_and_trans...
- 'Vessel_for_mixing_wine_with_water_with_Handle(s)_for_storing_and_trans...
- 'Vessel_for_mixing_wine_with_water_with_Handle(s)_for_storing_and_trans...
- 'Vessel_for_mixing_wine_with_water_with_Handle(s)_for_storing_and_trans...
- 'Vessel_for_mixing_wine_with_water_with_Handle(s)_for_storing_and_trans...
- 'Vessel_for_storing_and_transport_with_neck_with_Handle(s)'
- hydrie
- Vessel_for_perfume_with_neck
- volute-like handle
- water
- with_column-like_handles

Annotations | Ontology metrics

Annotations +

- skos:prefLabel [language: fr] cratère à colonnettes
- skos:prefLabel [language: en] column krater
- skos:prefLabel [language: gr] κρατίρας κιονωτός
- skos:definition [language: fr] Cratère avec cou, avec des anses en forme de colonne.
- skos:definition [language: en]

The diagram illustrates the Protégé interface for defining ontologies. On the left, the 'Class hierarchy' tab shows a tree structure of classes and their subtypes. The 'Annotations' tab on the right displays semantic metadata for the selected class, including labels in multiple languages and definitions. The central area is a graph visualization showing the relationships between various ontology components, such as Place, owl:Thing, Handle, Neck, Foot, and different types of vessels like Vessel, Vessel_for_moving_wine_with_water, and hydrie. The graph uses colored nodes and arrows to represent different types of relationships, such as inheritance, part-whole, and specific-to-general.



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



DOWNLOAD NOW



Protégé Desktop is a feature rich ontology editing environment with full support for the OWL 2 Web Ontology Language, and direct in-memory connections to description logic reasoners like Hermit and Pellet.

Protégé Desktop supports creation and editing of one or more ontologies in a single workspace via a completely customizable user interface. Visualization tools allow for interactive navigation of ontology relationships. Advanced explanation support aids in tracking down inconsistencies. Refactor operations available including ontology merging, moving axioms between ontologies, rename of multiple entities, and more.



Screenshots



Documentation



Resources

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



Download for Windows
Protégé Desktop v.5.5.0

Download platform independent version
(requires a Java Runtime Environment)

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 and dlm@ksl.stanford.edu

1 Why develop an ontology?

In recent years the development of ontologies—explicit formal specifications of the terms in the domain and relations among them (Gruber 1993)—has been moving from the realm of Artificial-Intelligence laboratories to the desktops of domain experts. Ontologies have become common on the World-Wide Web. The ontologies on the Web range from large taxonomies categorizing Web sites (such as on Yahoo!) to categorizations of products for sale and their features (such as on Amazon.com). The WWW Consortium (W3C) is developing the Resource Description Framework (Brickley and Guha 1999), a language for encoding knowledge on Web pages to make it understandable to electronic agents searching for information. The Defense Advanced Research Projects Agency (DARPA), in conjunction with the W3C, is developing DARPA Agent Markup Language (DAML) by extending RDF with more semantic primitives to facilitate the reuse of knowledge on the Web (Hendler and McGuinness 2000). Many disciplines now develop standardized ontologies that domain experts can use to share and annotate information in their fields. Medicine, for example, has produced large, standardized, structured vocabularies such as SNOMED (Prise and Spackman 2000) and the semantic network of the Unified Medical Language System (Humphreys and Lindberg 1993). Broad general-purpose ontologies are emerging as well. For example, the United Nations Development Program and Dun & Bradstreet combined their efforts to develop the UNSPSC ontology which provides terminology for products and services (www.unpsc.org).

An ontology defines a common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and relations among them.

Why would someone want to develop an ontology? Some of the reasons are:

- To share common understanding of the structure of information among people or software agents
- To enable reuse of domain knowledge
- To make domain assumptions explicit
- To separate domain knowledge from the operational knowledge
- To analyze domain knowledge

Sharing common understanding of the structure of information among people or software agents is one of the more common goals in developing ontologies (Musen 1992; Gruber 1993). For example, suppose several different Web sites contain medical information or provide medical e-commerce services. If these Web sites share and publish the same underlying ontology of the terms they all use, then computer agents can extract and aggregate information from these different sites. The agents can use this aggregated information to answer user queries or as input data to other applications.

Enabling reuse of domain knowledge was one of the driving forces behind recent surge in ontology research. For example, models for many different domains need to represent the notion of time. This representation includes the notions of time intervals, points in time, relative measures of time, and so on. If one group of researchers develops such an ontology in detail, others can simply reuse it for their domains. Additionally, if we need to build a large

DOCUMENTATION

Learn about the Protégé toolset and developing ontologies using our product documentation.

USERS

DEVELOPERS

Look here first!

» [Protégé Frequently Asked Questions](#)

Find answers to commonly asked questions about the Protégé Project.

» [WebProtégé Overview](#)

Get an overview of WebProtégé features and the latest project activity.

» [WebProtégé User's Guide](#)

A short user's guide with screencasts and screenshots describing the main user interface elements in WebProtégé.

» [Protégé Desktop](#)

Top-level entry point for all documentation pertaining to the latest version of Protégé Desktop, including an overview of editor features, links to tutorials etc.

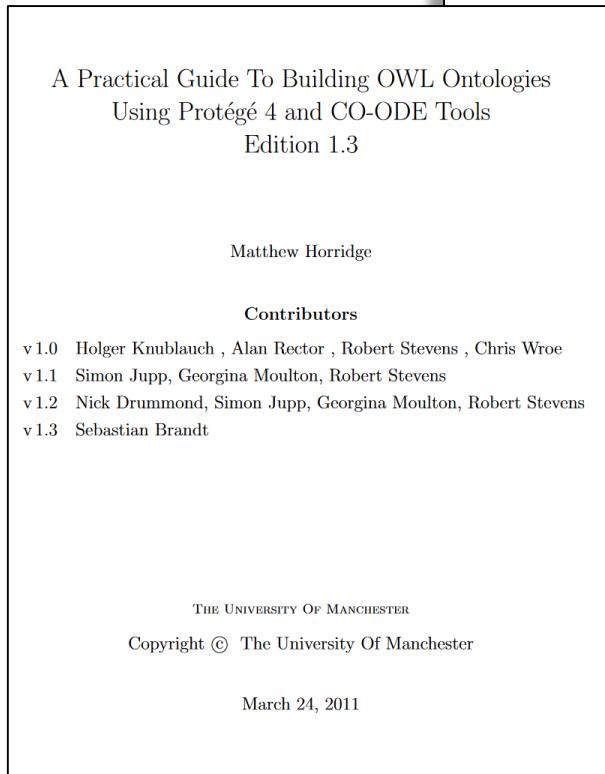
» [Ontology Development 101](#)

A guide to creating your first ontology.

» [Protégé OWL Tutorial](#)

A substantial guide to the Web Ontology Language (OWL) and ontology engineering.

22



DOCUMENTATION

Learn about the Protégé toolset and developing ontologies using our product documentation.

USERS

DEVELOPERS

Look here first!

» [Protégé Frequently Asked Questions](#)

Find answers to commonly asked questions about the Protégé Project.

» [WebProtégé Overview](#)

Get an overview of WebProtégé features and the latest project activity.

» [WebProtégé User's Guide](#)

A short user's guide with screencasts and screenshots describing the main user interface elements in WebProtégé.

» [Protégé Desktop](#)

Top-level entry point for all documentation pertaining to the latest version of Protégé Desktop, including an overview of editor features, links to tutorials etc.

» [Ontology Development 101](#)

A guide to creating your first ontology.

» [Protégé OWL Tutorial](#)

A substantial guide to the Web Ontology Language (OWL) and ontology engineering.

 Protégé 5 Documentation

Installation Getting Started Views Menus Class Expression Syntax

Protege Documentation

This is the official documentation for Protégé 5.5.0. You can find information about the Protégé user interface including descriptions of the various [views](#) and [menu](#) items.

Installation
Explains how to install Protégé on Windows, Mac OS X and Linux.

Getting Started
A quick start guide for Protege.

Views
Provides a list of all of the default views that are distributed with Protégé

Menus
Explains what each menu item in Protégé does

Class Expression Syntax
Provides a reference for the class expression syntax that is used throughout Protégé.

DOCUMENTATION

Learn about the Protégé toolset and developing ontologies using our product documentation.

USERS

DEVELOPERS

Look here first!

» [Protégé Frequently Asked Questions](#)

Find answers to commonly asked questions about the Protégé Project.

» [WebProtégé Overview](#)

Get an overview of WebProtégé features and the latest project activity.

» [WebProtégé User's Guide](#)

A short user's guide with screencasts and screenshots describing the main user interface elements in WebProtégé.

» [Protégé Desktop](#)

Top-level entry point for all documentation pertaining to the latest version of Protégé Desktop, including an overview of editor features, links to tutorials etc.

» [Ontology Development 101](#)

A guide to creating your first ontology.

» [Protégé OWL Tutorial](#)

A substantial guide to the Web Ontology Language (OWL) and ontology engineering.

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.

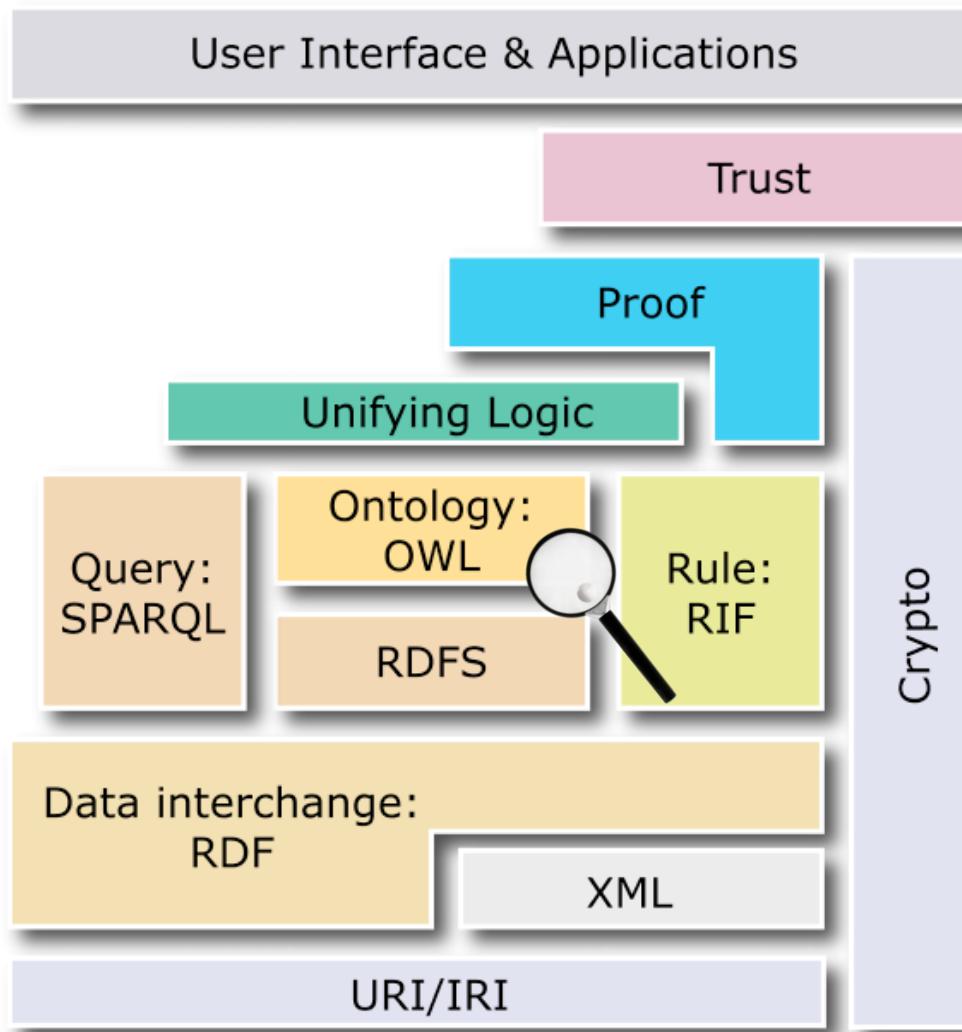
Open World Assumption

The Open World Assumption, used by OWL, says that "just because we don't know something to be true does not mean that we can assume it to be false"

OWL

OWL provides the theoretical basis for Protege 4 ontologies

OWL & the Semantic Web Architecture



What are OWL Ontologies?

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.

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

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.

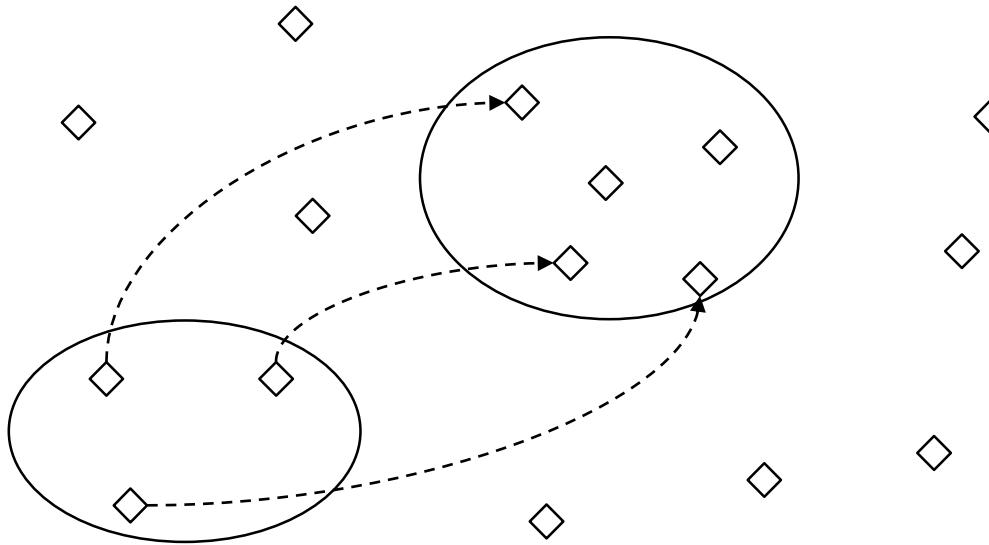
A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools - Edition 1.3

Main ideas of OWL (DL)?

Extensional Logic

1

Organising the objects with populate the world into classes according to the relationships that linked objects together



2

An object is not defined by its “nature”, but through its relationships with other objects



Components of OWL Ontologies: Individuals

1) Individuals

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

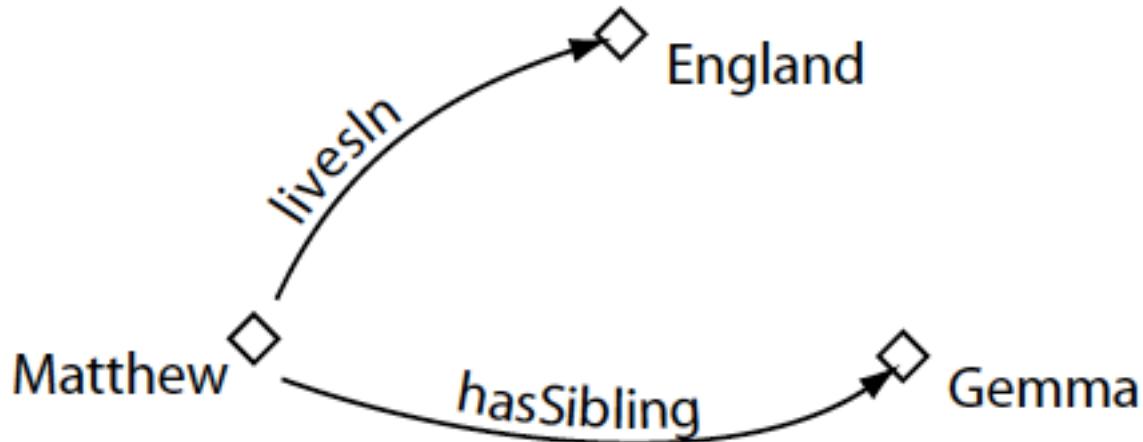


Terminology: « *individual* », « *instance* », « *object* »

Components of OWL Ontologies: Properties

2) Properties

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

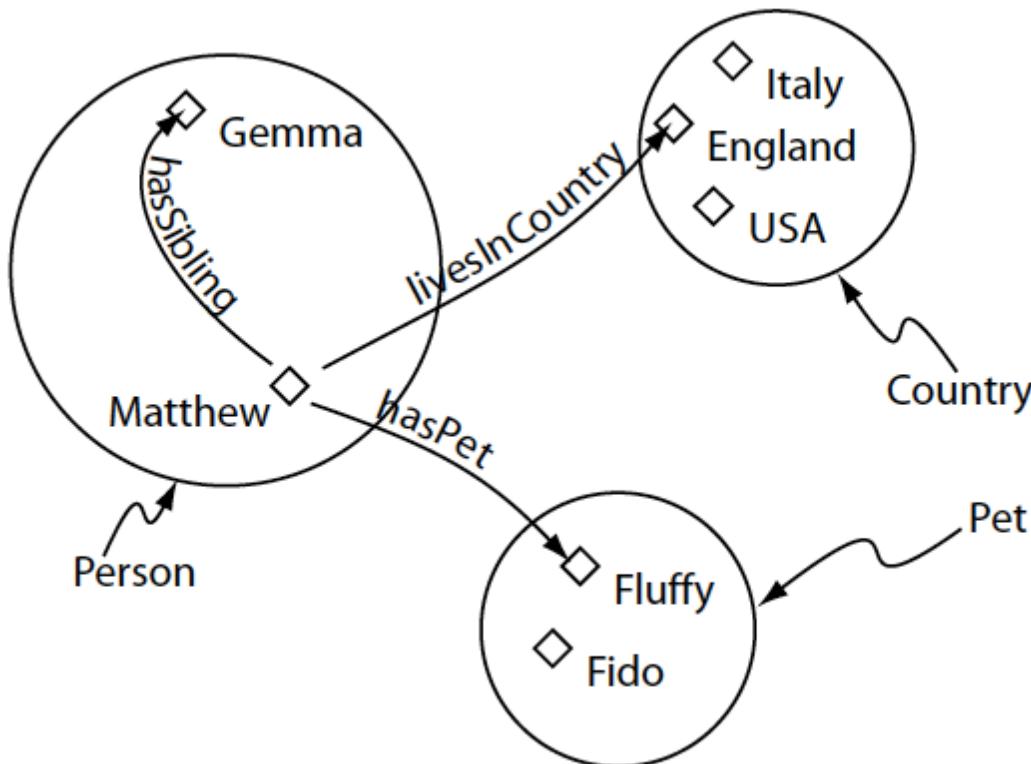


Terminology: « properties », « slots » (Protégé), « roles » (DL), « relations », « attributes »

Components of OWL Ontologies: Classes

3) Classes

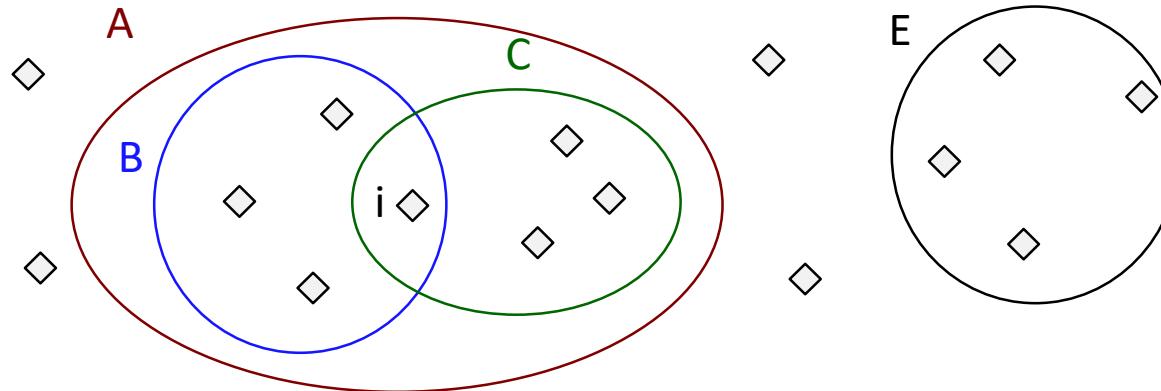
OWL classes are interpreted as sets that contain individuals.



Classes are a concrete representation of concepts.

Components of OWL Ontologies: Classes

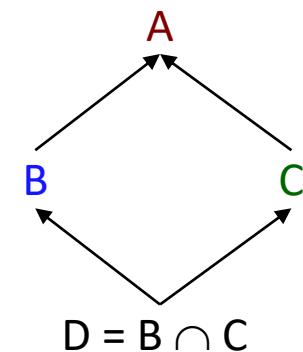
An individual can belong to different classes: $i \in B, i \in C, i \in B \cap C$



As sets, classes can be defined using set operators : \cup, \cap $A = B \cup C$

Classes may be organised into a superclass-subclass hierarchy corresponding to inclusion between sets:

$B \subseteq A$ All members of the class B
are members of the class A



Classes can be disjoint

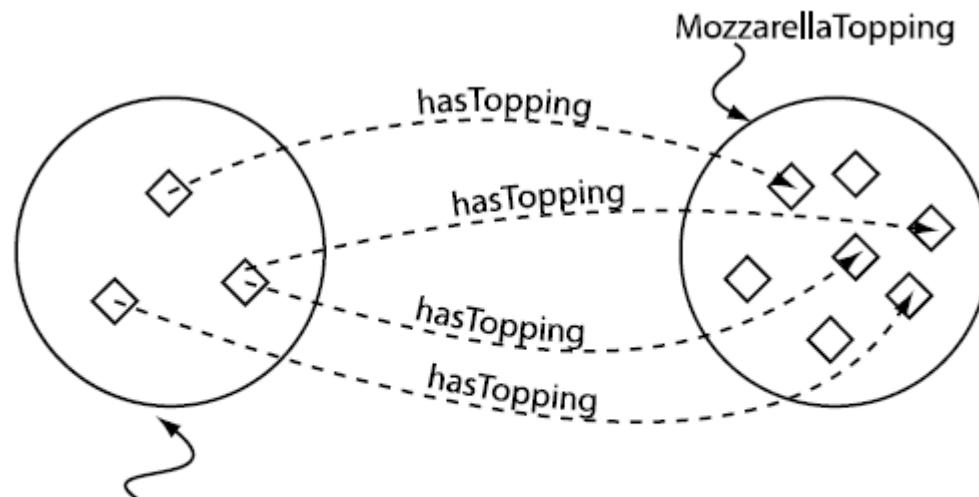
Components of OWL Ontologies: Properties Restriction

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

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.

(at least one value of the property must be of a certain type)



Things that have at least one
MozzarellaTopping
(hasTopping some MozzarellaTopping)

Components of OWL Ontologies: Properties restriction

- 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

Components of OWL Ontologies: Reasoner

5) Reasoner

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

My 1st K-Graph in Protégé



armchair

椅子



chaise

pouf

fauteuil

banc

stool

长凳

tabouret



bench

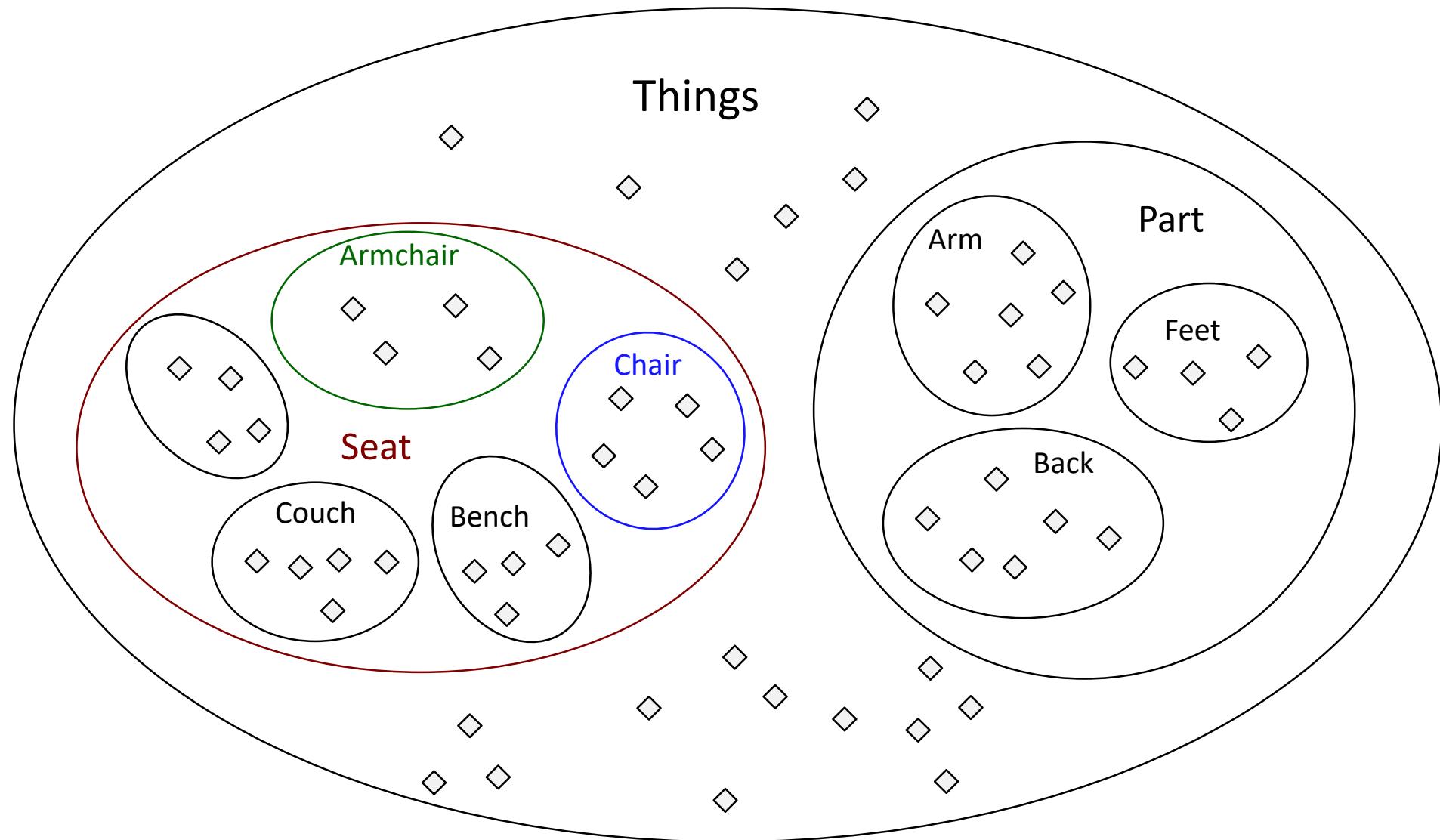
chair

canapé

couch

长椅

Building an OWL Ontology using Protégé: Named Classes

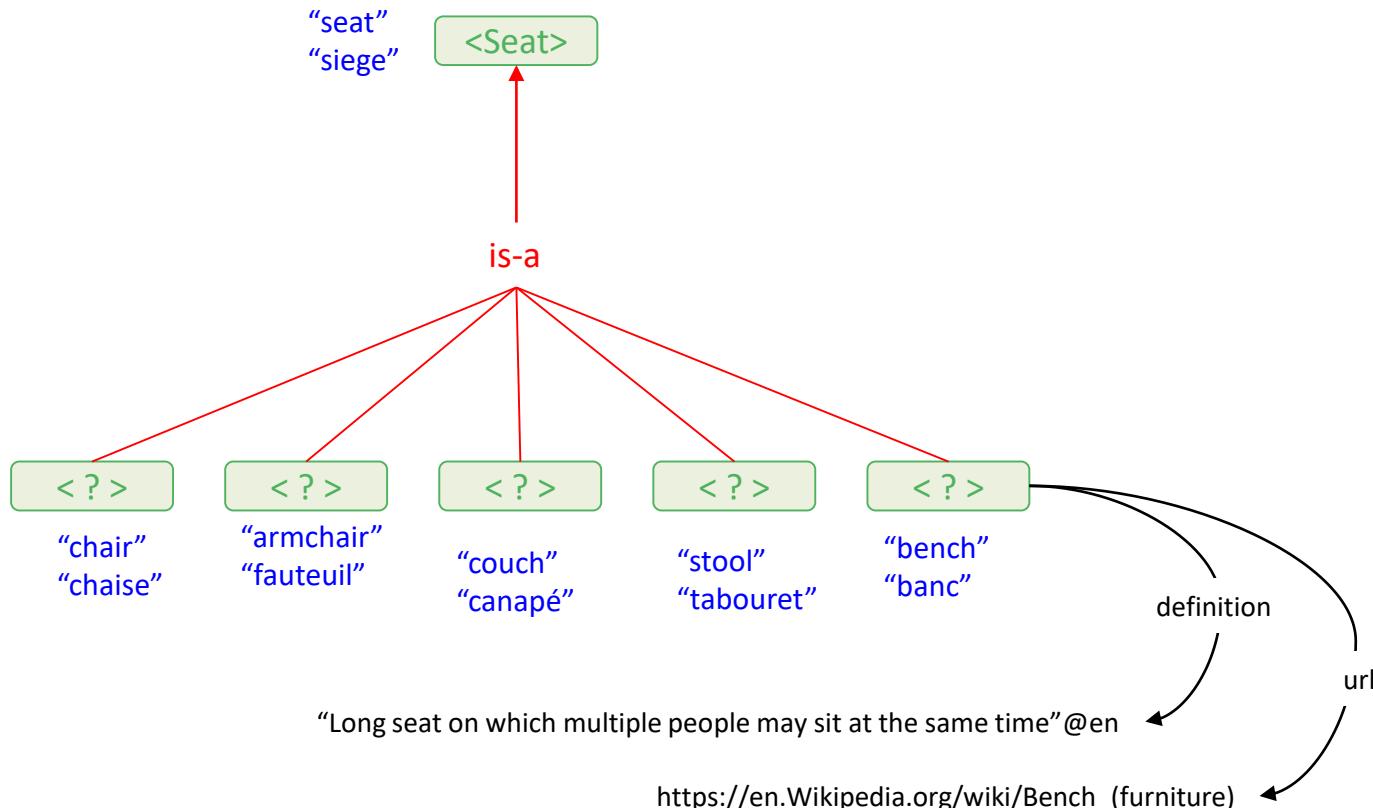


My 1st K-Graph in Protégé



1. Building my 1st K-Graph in Protégé

2. Querying my 1st K-Graph in Protégé



My 1st K-Graph in Protégé



protégé

1 Defining classes

2 Organizing them protégé

1 Class

3 Annotation

"seat"
"siege"

<Seat>

Class

1

3 Annotating them

2 Subclass
relationship

1 Class

1 Class

< ? >
"chair"
"chaise"

< ? >
"armchair"
"fauteuil"

< ? >
"couch"
"canapé"

< ? >
"stool"
"tabouret"

< ? >
"bench"
"banc"

is-a

3 Annotation

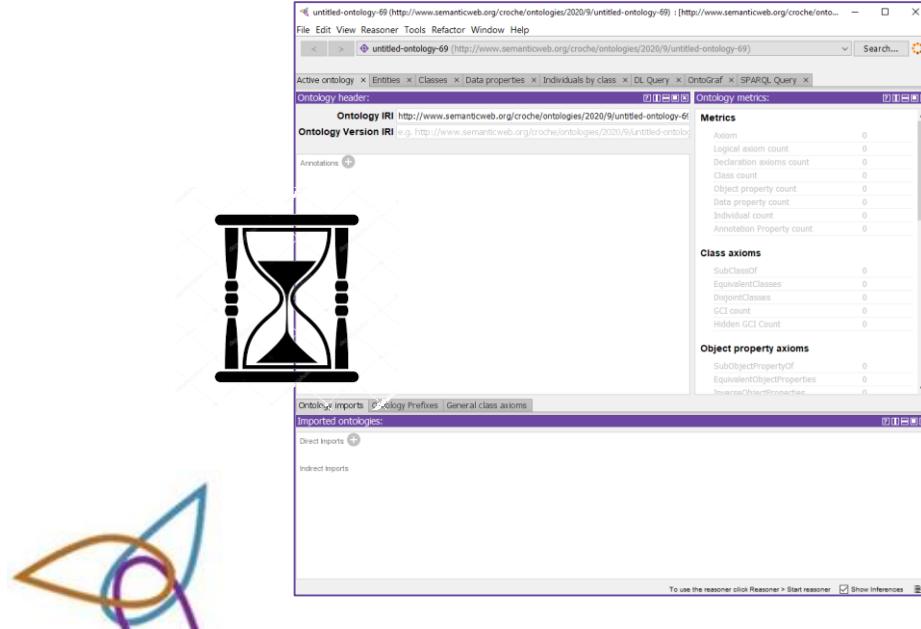
3

definition

url

"Long seat on which multiple people may sit at the same time"@en

[https://en.Wikipedia.org/wiki/Bench_\(furniture\)](https://en.Wikipedia.org/wiki/Bench_(furniture))



protégé



untitled-ontology-76 (http://www.semanticweb.org/croche/ontologies/2020/9/untitled-ontology-76) : [h...] — □ ×

File Edit View Reasoner Tools Refactor Window Help

< > ♦ untitled-ontology-76 Search...

Active ontology × Entities × Classes × Data properties × Individuals by class × DL Query × OntoGraf × SPARQL Query ×

Ontology header: 

Ontology IRI http://www.semanticweb.org/croche/ontologies/2020/9/untitled-ontology-76

Ontology Version IRI e.g. http://www.semanticweb.org/croche/ontologies/2020/9/untitled-ontology-76

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

Object property axioms

Ontology imports | Ontology Prefixes | General class axioms

Imported ontologies:

Direct Imports +

Indirect Imports

To use the reasoner click Reasoner > Start reasoner Show Inferences



untitled-ontology-76 (<http://www.semanticweb.org/croche/ontologies/2020/9/untitled-ontology-76>)

File Edit View Reasoner Tools Refactor Window Help

< > untitled-ontology-76

Data properties x Individuals by class

Active ontology

Ontology header:

Ontology IRI <http://www.semanticweb.org>

Ontology Version IRI e.g. <http://www.semanticweb.org>

Annotations +

Views > Search...

Tabs > ✓ Active ontology
✓ Entities
✓ Classes
Object properties
✓ Data properties
Annotation properties
✓ Individuals by class
OWLviz
✓ DL Query
SWRLTab
✓ OntoGraf
SQWRLTab
✓ SPARQL Query

Create new tab...
Delete custom tabs...
Import tab...
Export current tab...
Store current layout
Reset selected tab to default state
Capture view to clipboard...
Timestamp log / console
Show log...
Look & Feel >
Refresh user interface

F 0
Classes 0
DisjointClasses 0
GCI count 0
Hidden GCI Count 0

Ontology imports | Ontology Prefixes | General class axioms

Imported ontologies:

Direct Imports +

Indirect Imports

To use the reasoner click Reasoner > Start reasoner Show Inferences

1 Defining classes

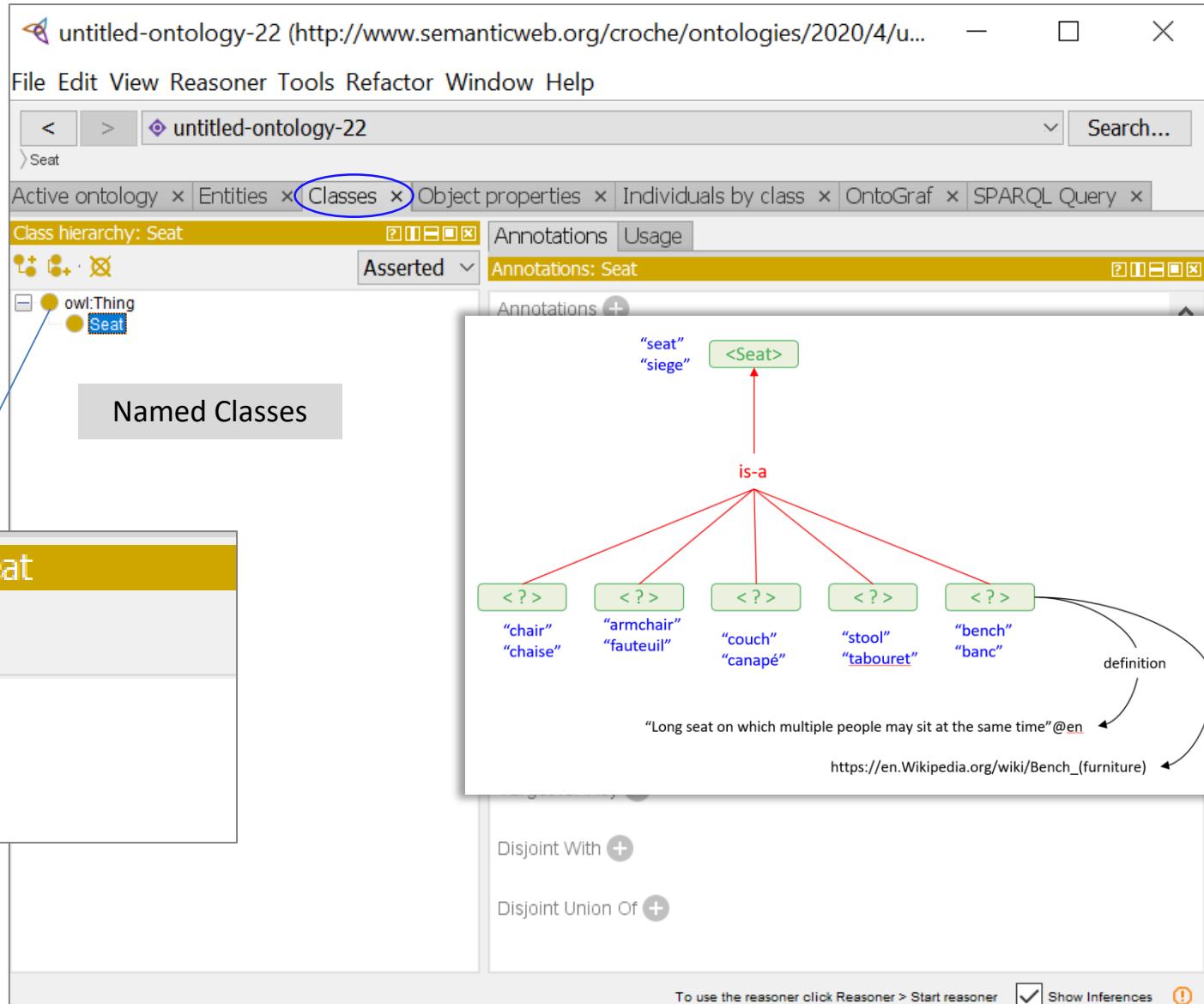
2 Organizing them



The class Thing represents the set containing all individuals

Class hierarchy: Seat

owl:Thing
Seat



1 Defining classes

2 Organizing them



untitled-ontology-6 (<http://www.semanticweb.org/croche/ontologies/2020/6/untitled-ontology-6>) : [http://www.semanticweb.org/croche...]

File Edit View Reasoner Tools Refactor Window Help

< > untitled-ontology-6 (<http://www.semanticweb.org/croche/ontologies/2020/6/untitled-ontology-6>) Search...

Annotations Usage

Class hierarchy: Seat

Add subclass

Annotations: Seat

Description: Seat

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

Annotations: Annotations: Seat

Description: Description: Seat

SubClass Of: SubClass Of

General class axioms: General class axioms

SubClass Of (Anonymous Ancestor): SubClass Of (Anonymous Ancestor)

Instances: Instances

Target for Key: Target for Key

Disjoint With: Disjoint With

Disjoint Union Of: Disjoint Union Of

OWL Viz: OWLViz

OntoGraf: OntoGraf

SPARQL Query: SPARQL Query

OWL Class Hierarchy Diagram:

```
graph TD; Seat --> chair["chair<br><u>chaise</u>"]; Seat --> armchair["armchair<br><u>fauteuil</u>"]; Seat --> couch["couch<br><u>canapé</u>"]; Seat --> stool["stool<br><u>tabouret</u>"]; Seat --> bench["bench<br><u>banc</u>"]
```

Annotations:

- "seat" "siege" [<Seat>](#)
- "is-a"
- "chair" "chaise" [<?>](#)
- "armchair" "fauteuil" [<?>](#)
- "couch" "canapé" [<?>](#)
- "stool" "tabouret" [<?>](#)
- "bench" "banc" [<?>](#)

Notes:

- "Long seat on which multiple people may sit at the same time"@en
- [https://en.Wikipedia.org/wiki/Bench_\(furniture\)](https://en.Wikipedia.org/wiki/Bench_(furniture))
- definition
- url

1 Defining classes

2 Organizing them



untitled-ontology-6 (<http://www.semanticweb.org/croche/ontologies/2020/6/untitled-ontology-6>) : [http://www.semanticweb.org/croche...]

File Edit View Reasoner Tools Refactor Window Help

< > untitled-ontology-6 (<http://www.semanticweb.org/croche/ontologies/2020/6/untitled-ontology-6>) Search...

› Seat › Chair

Active ontology Entities Classes Object properties Data properties Individuals by class OWLViz OntoGraf SPARQL Query

Class hierarchy: Chair Annotations Usage

owl:Thing
Seat
Chair

Annotations: Chair

Annotations +

Description: Chair

Equivalent To +
SubClass Of +
Seat

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 !

```

graph TD
    owlThing[owl:Thing] --> seat[Seat]
    seat --> chair[Chair]
    chair -- "is-a" --> seat
    chair -- "Annotations" --> annotations[Annotations]
    annotations -- "French terms" --> chairAnnotations["chair", "fauteuil", "canapé", "tabouret", "banc"]
    annotations -- "definition" --> chairDefinition["Long seat on which multiple people may sit at the same time" @ en]
    annotations -- "url" --> chairUrl["https://en.Wikipedia.org/wiki/Bench_(furniture)"]
  
```

1 Defining classes

2 Organizing them



Screenshot of the protégé ontology editor showing the Class hierarchy for the Chair class.

The Class hierarchy pane shows:

- owl:Thing
- Seat
- Chair (selected)

The Annotations pane for Chair contains:

- Annotations: Chair
- Annotations Of: Chair

The Description pane for Chair contains:

- Equivalent To: +
- SubClass Of: +
Seat (circled in red)
- General class axioms: +
- SubClass Of (Anonymous Ancestor):
- Instances: +
- Target for Key: +
- Disjoint With: +
- Disjoint Union Of: +

A detailed diagram of the Chair class is displayed in a modal window:

- Annotations:** <Seat>, <siege>
- Description:** is-a
- Subclasses:**
 - <?> "chair" "chaise"
 - <?> "armchair" "fauteuil"
 - <?> "couch" "canapé"
 - <?> "stool" "tabouret"
 - <?> "bench" "banc"
- Notes:**
 - "Long seat on which multiple people may sit at the same time"@en
 - url: https://en.Wikipedia.org/wiki/Bench_(furniture)
- Relationships:**
 - A curved arrow labeled "definition" points from the "bench" subclass to the "Long seat on which multiple people may sit at the same time" note.
 - A curved arrow labeled "url" points from the "url" note back to the "bench" subclass.

1 Defining classes

2 Organizing them



Screenshot of the protégé ontology editor showing the class hierarchy for 'Chair'.

The left sidebar shows the class hierarchy under 'Chair':

- owl:Thing
- Add sibling class
- Chair

A red arrow points to the 'Add sibling class' button.

The main workspace displays the 'Annotations: Chair' tab, which includes:

- An 'Annotations' section with a '+' button.
- A large diagram titled 'is-a' showing inheritance relationships:
 - 'Chair' is a subclass of 'Seat'.
 - 'Seat' is a subclass of 'Siege'.
 - 'Siege' is a subclass of 'owl:Thing'.
- A list of subclasses:
 - <?> "chair" "chaise"
 - <?> "armchair" "fauteuil"
 - <?> "couch" "canapé"
 - <?> "stool" "tabouret"
 - <?> "bench" "banc"
- A note: "Long seat on which multiple people may sit at the same time"@en
- A URL: [https://en.Wikipedia.org/wiki/Bench_\(furniture\)](https://en.Wikipedia.org/wiki/Bench_(furniture))
- A 'definition' link pointing to the URL.

The bottom status bar includes:

- To use the reasoner click Reasoner > Start reasoner
- Show Inferences
- A warning icon

1 Defining classes

2 Organizing them



untitled-ontology-6 (<http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6>) : [http://www.semanticweb....] — X

File Edit View Reasoner Tools Refactor Window Help

< > untitled-ontology-6 (<http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6>) Search...

Active ontology x Entities x Classes x Object properties x Data properties x Individuals by class x DL Query x SPARQL Query x

Class hierarchy (inferred)

Annotations Usage

Annotations:

Annotations +

Class hierarchy: Asserted

owl:Thing

Seat

Armchair

Chair

Annotations:

Description:

Equivalent To +

SubClass Of +

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

Disjoint Union Of +

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

The diagram shows the 'Seat' class hierarchy. At the top is the 'Seat' class, which is asserted to be a subclass of 'owl:Thing'. Below it is the 'Armchair' class, also asserted to be a subclass of 'Seat'. Further down is the 'Chair' class, also asserted to be a subclass of 'Seat'. The 'Seat' class is also shown to be an inferred subclass of 'owl:Thing'.

Annotations:

The 'Annotations' tab is active. It displays a tree diagram under the heading 'is-a'. The root node is 'Seat' (with annotations 'seat' and 'siege'). It branches into five nodes: 'chair' ('chaise'), 'armchair' ('fauteuil'), 'couch' ('canapé'), 'stool' ('tabouret'), and 'bench' ('banc'). A note below the tree states: "Long seat on which multiple people may sit at the same time"@en. A URL link is provided: [https://en.Wikipedia.org/wiki/Bench_\(furniture\)](https://en.Wikipedia.org/wiki/Bench_(furniture)). A curved arrow labeled 'definition' points to the 'couch' node, and another curved arrow labeled 'url' points to the Wikipedia link.

1 Defining classes

2 Organizing them



Screenshot of the protégé ontology editor showing the organization of classes under the `Seat` class.

The left sidebar shows the class hierarchy (inferred) starting from `owl:Thing`, with `Seat` as the root class. Subclasses include `Armchair`, `Bench`, `Chair`, `Couch`, and `Stool`.

The main workspace displays the `Annotations: Seat` tab, which contains:

- Description: Seat**:
 - Equivalent To: `<?>`
 - SubClass Of: `<?>`
 - General class axioms: `<?>`
 - SubClass Of (Anonymous Ancestor)
 - Instances: `<?>`
 - Target for Key: `<?>`
 - Disjoint With: `<?>`
 - Disjoint Union Of: `<?>`
- Annotations: Seat**:
 - Annotations: `<?>`
 - Usage: `<?>`

A detailed diagram of the `is-a` inheritance hierarchy is shown, with `<Seat>` at the top, connected to `<?>` via red arrows. Below `<?>` are five boxes containing pairs of labels:

- `<?>`: "seat" "siège"
- `<?>`: "chair" "chaise"
- `<?>`: "armchair" "fauteuil"
- `<?>`: "couch" "canapé"
- `<?>`: "stool" "tabouret"
- `<?>`: "bench" "banc"

Annotations for the `Bench` class are shown in a callout box:

- "Long seat on which multiple people may sit at the same time"@en
- [https://en.Wikipedia.org/wiki/Bench_\(furniture\)](https://en.Wikipedia.org/wiki/Bench_(furniture))

Annotations for the `Seat` class are also present:

- definition
- url

At the bottom, a message states: "No Reasoner set. Select a reasoner from the Reasoner menu" with a checked checkbox for "Show Inferences".

1 Defining classes

2 Organizing them

2' Adding properties



Screenshot of the protégé ontology editor showing the organization of classes:

The left sidebar shows the class hierarchy (inferred) under owl:Thing, with Seat as the active class. Subclasses include Armchair, Bench, Chair, Couch, and Stool.

The main workspace displays the annotations for the class `<Seat>`. The annotations tab is selected, showing the annotation `Annotations: Seat`.

A detailed diagram of the `is-a` inheritance hierarchy is shown, with `<Seat>` at the top, followed by `chair`, `armchair`, `couch`, `stool`, and `bench` below it.

The description tab for `Seat` includes:

- Equivalent To: `<?>` "seat" "siège"
- SubClass Of: `<?>` "chair" "chaise", `<?>` "armchair" "fauteuil", `<?>` "couch" "canapé", `<?>` "stool" "tabouret", `<?>` "bench" "banc"
- General class axioms: `<?>`
- SubClass Of (Anonymous Ancestor): `<?>`
- Instances: `<?>`
- Target for Key: `<?>`
- Disjoint With: `<?>` (highlighted with a red arrow)
- Disjoint Union Of: `<?>`

The bottom right corner of the workspace contains a callout box with the following information:

- Definition: "Long seat on which multiple people may sit at the same time"@en
- URL: [https://en.Wikipedia.org/wiki/Bench_\(furniture\)](https://en.Wikipedia.org/wiki/Bench_(furniture))

At the bottom of the interface, there is a message: "No Reasoner set. Select a reasoner from the Reasoner menu" and a checkbox for "Show Inferences".

1 Defining classes

2 Organizing them

2' Adding properties



Screenshot of the Protégé ontology editor showing the 'Annotations' tab for the 'Armchair' class.

The 'Annotations' tab displays the following information:

- Annotations:** Armchair
- Description:** Armchair
- General class axioms:** +
- SubClass Of (Anonymous Ancestor):** +
- Instances:** +
- Target for Key:** +
- Disjoint With:** + (This section is circled in red.)
- Disjoint Union Of:** +

The 'Annotations' tab also shows a diagram of the class hierarchy:

```
graph TD; Seat --> Armchair; Armchair --> Chair; Armchair --> Bench; Armchair --> Couch; Armchair --> Stool;
```

A callout box highlights the 'Disjoint With' section, which lists:

- Couch
- Stool
- Chair
- Bench

An annotation for the 'Bench' class is shown in the main pane:

"Long seat on which multiple people may sit at the same time"@en
[https://en.Wikipedia.org/wiki/Bench_\(furniture\)](https://en.Wikipedia.org/wiki/Bench_(furniture))

Annotations for the 'Bench' class include:

- <?> "chair" "chaise"
- <?> "armchair" "fauteuil"
- <?> "couch" "canapé"
- <?> "stool" "tabouret"
- <?> "bench" "banc"

Annotations for the 'Seat' class include:

- <?> "seat" "siège"
- <?> "siege"

Annotations for the 'is-a' relationship include:

- <Seat>

Annotations for the 'definition' and 'url' of the 'Bench' class are also present.

1 Defining classes

2 Organizing them

2' Adding properties



Screenshot of the Protégé ontology editor showing the 'Annotations' tab for the 'Bench' class.

The 'Annotations' tab displays the following information:

- Annotations:** Bench
- Description:** Bench
- General class axioms:** +
- SubClass Of (Anonymous Ancestor):**
- Instances:** +
- Target for Key:** +
- Disjoint With:** + (This section is circled in red)
- Disjoint Union Of:** +

The main workspace shows the class hierarchy for 'Seat' and its subclasses: Armchair, Bench, Chair, Couch, and Stool. A detailed diagram for the 'Bench' class is shown, illustrating its definition and URL.

Definition Diagram:

- The 'Bench' class is defined as an 'is-a' subclass of 'Seat'.
- The 'Seat' class is annotated with 'seat' and 'siege'.
- The 'Bench' class has five disjoint subclasses: 'chair' ('chaise'), 'armchair' ('fauteuil'), 'couch' ('canapé'), 'stool' ('tabouret'), and 'bench' ('banc').
- A note states: "Long seat on which multiple people may sit at the same time"@en.
- The URL for the definition is: [https://en.Wikipedia.org/wiki/Bench_\(furniture\)](https://en.Wikipedia.org/wiki/Bench_(furniture))

Bottom Status Bar:

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

1 Defining classes

2 Organizing them

2' Adding properties



Object properties Data properties Individuals by class DL Query SPARQL Query **OntoGraf** Entities Classes

Class hierarchy: S □ □ □ □ □ OntoGraf:

Asserted ▾

owl:Thing
Seat
Armchair
Bench
Chair
Couch
Stool

Search: contains Search Clear

Diagram showing the class hierarchy:

```
graph TD; owlThing[owl:Thing] --> seat[Seat]; seat --> stool[Stool]; seat --> chair[Chair]; seat --> couch[Couch]; seat --> bench[Bench]; seat --> armchair[Armchair]
```

Details for the Seat class:

"seat"
"siege"
<Seat>

is-a

<?> "chair" "chaise"
<?> "armchair" "fauteuil"
<?> "couch" "canapé"
<?> "stool" "tabouret"
<?> "bench" "banc"

definition

url

"Long seat on which multiple people may sit at the same time"@en

[https://en.wikipedia.org/wiki/Bench_\(furniture\)](https://en.wikipedia.org/wiki/Bench_(furniture))

1 Defining classes

2 Organizing them

3 Annotating them



Annotations Usage

Annotations: Bench

Annotations +

Description: Bench

< Bench>

"bench"@en
"banc"@fr

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

Stool
Couch
Chair
Armchair

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

This screenshot of the protégé ontology editor illustrates the three steps of defining, organizing, and annotating classes. The left sidebar shows the 'Class hierarchy (inferred)' and 'Class hierarchy' panes, with 'Bench' selected. The main workspace shows the 'Annotations' tab for 'Bench', where annotations like "bench"@en and "banc"@fr are listed. A callout highlights the '**Bench**' class with these annotations. Below the annotations, the 'Description' tab lists 'Seat' as a general class axiom. Further down, 'SubClass Of (Anonymous Ancestor)', 'Instances', and 'Target for Key' sections are shown. At the bottom, the 'Disjoint With' section lists 'Stool', 'Couch', 'Chair', and 'Armchair' as disjoint classes. A note at the bottom indicates no reasoner is set, with options to select one or show inferences.

1 Defining classes

2 Organizing them

3 Annotating them



The screenshot shows the Protégé ontology editor interface. On the left, the 'Class hierarchy (inferred)' panel displays a tree of classes under 'owl:Thing'. A red arrow points to the '+' button in the 'Annotations' tab of the 'Annotations: Bench' panel. The 'Annotations' panel lists various OWL annotations, with 'rdfs:isDefinedBy' selected and its IRI value 'http://www.w3.org/2000/01/rdf-schema#isDefinedBy' highlighted. The 'Description: Bench' panel shows 'Seat' as a general class axiom. The 'Disjoint With' panel lists 'Stool', 'Couch', 'Chair', and 'Armchair' as disjoint classes. At the bottom, a message says 'No Reasoner set. Select a reasoner from the Reasoner menu' with a checked checkbox for 'Show Inferences'.

1 Defining classes

2 Organizing them

3 Annotating them

The screenshot illustrates the Protege ontology editor interface across three main sections: Class hierarchy, Annotations, and Disjoint With.

Class hierarchy (inferred): Shows the inferred class hierarchy. The 'Bench' class is highlighted in blue. Other classes listed include Armchair, Bench, Chair, Couch, and Stool.

Annotations: Bench: This panel shows annotations for the 'Bench' class. A red arrow points to the 'Annotations' tab. Another red arrow points to the 'rdfs:label' entry in the dropdown menu.

Description: Bench: This panel shows the description of the 'Bench' class. It includes a link to 'Seat' and a section for 'General class axioms'.

Annotations: Bench (Details): A detailed view of the 'Annotations' tab for 'Bench'. It lists various owl properties and rdfs:label. A red box highlights the 'rdfs:label' entry. A callout bubble contains the labels "bench" @ en and "banc" @ fr.

Disjoint With: This panel lists classes that are disjoint with 'Bench': Stool, Couch, Chair, and Armchair.

Annotations Dialog (Bottom Right): A modal dialog for adding annotations. It has fields for 'Type' (set to 'xsd:string'), 'Lang' (set to 'en'), and buttons for 'OK' and 'Annuler'. Red arrows point to the 'Type' and 'Lang' fields.

Footer: The footer indicates "No Reasoner set. Select a reasoner from the Reasoner menu" and a checked "Show Inferences" checkbox.

1 Defining classes

2 Organizing them

3 Annotating them



Annotations Usage

Annotations: Bench

Annotations +

rdfs:label [language: en]
bench

Description: Bench

Seat

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

Stool
Couch
Chair
Armchair

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

< Bench >

"bench"@en
"banc"@fr

The screenshot shows the Protégé ontology editor interface. On the left, the 'Class hierarchy (inferred)' panel displays a tree structure under 'owl:Thing' with nodes for 'Seat', 'Bench', 'Chair', 'Couch', and 'Stool'. The 'Annotations' tab is selected in the top navigation bar. In the main workspace, the 'Annotations: Bench' panel shows an 'rdfs:label' assertion with the value 'bench' and language 'en'. Below it, the 'Description: Bench' panel shows the class 'Seat' and various general class axioms like 'SubClass Of (Anonymous Ancestor)'. A callout box highlights the '< Bench >' node, with a red arrow pointing from the 'bench' label in the annotations to this node, and a dashed arrow pointing from the 'banc' label back to the same node, illustrating how multiple labels can be associated with a single class instance.

1 Defining classes

2 Organizing them

3 Annotating them



The screenshot shows the Protégé ontology editor interface. On the left, the Class hierarchy (inferred) panel displays a tree structure under the root 'owl:Thing'. The 'Seat' class is expanded, showing its subclasses: 'Armchair', 'Bench' (which is selected and highlighted in blue), 'Chair', 'Couch', and 'Stool'. The 'Annotations' tab is active in the main workspace, showing annotations for the 'Bench' class. Two 'rdfs:label' statements are listed: one in French ('banc') and one in English ('bench'). A callout box points to the English label 'bench' with the text "'bench'@en". The 'Description' tab is also visible, showing the 'Seat' class definition, which includes a general class axiom and a list of disjoint classes: 'Stool', 'Couch', 'Chair', and 'Armchair'. A callout box points to the French label 'banc' with the text "'banc'@fr".

Annotations Usage

Annotations: Bench

Annotations +

rdfs:label [language: fr]
banc

rdfs:label [language: en]
bench

Description: Bench

Seat

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

Stool
Couch
Chair
Armchair

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

1 Defining classes

2 Organizing them

3 Annotating them



The screenshot shows the Protégé ontology editor interface with three main panels:

- Class hierarchy (inferred)**: Shows the class hierarchy under "owl:Thing". The class **Bench** is selected and highlighted in blue. Other classes shown include **Seat**, **Armchair**, **Chair**, **Couch**, and **Stool**.
- Annotations: Bench**: Displays annotations for the **Bench** class:
 - rdfs:label [language: fr] : banc
 - rdfs:label [language: en] : bench
- Description: Bench**: Displays the description of the **Bench** class:
 - Seat** (highlighted in blue)
 - General class axioms
 - SubClass Of (Anonymous Ancestor)
 - Instances
 - Target for Key
 - Disjoint With
 - Stool**
 - Couch**
 - Chair**
 - Armchair**

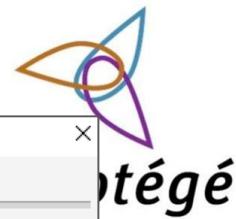
A callout box points from the "definition" label to the description text: "long seat on which multiple people may sit at the same time"@en.

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

1 Defining classes

2 Organizing them

3 Annotating them



The screenshot shows the ontotèque interface for defining and annotating classes. On the left, the 'Class hierarchy (inferred)' panel displays a tree structure under 'owl:Thing' with nodes for 'Seat', 'Armchair', 'Bench', 'Chair', 'Couch', and 'Stool'. The 'Annotations' panel on the right lists various annotations like owl:backwardCompatibleWith, owl:deprecated, etc., with 'rdfs:comment' selected. A modal dialog for 'Annotations' is open, showing the comment 'long seat on which multiple people may sit at the same time' with type 'xsd:string' and language 'en'. Below the annotations, a diagram shows a 'Bench' node connected to a text box containing the definition: 'long seat on which multiple people may sit at the same time'@en. A curved arrow labeled 'definition' points from the text box back to the 'Bench' node.

1 Defining classes

2 Organizing them

3 Annotating them



Annotations Usage

Annotations: Bench

Annotations +

rdfs:label [language: fr]
 banc

rdfs:label [language: en]
 bench

rdfs:comment [language: en]
 long seat on which multiple people may sit at the same time

Description: Bench

Seat

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

- Stool
- Couch
- Chair
- Armchair

Bench

"long seat on which multiple people may sit at the same time"@en

definition

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

1 Defining classes

2 Organizing them

3 Annotating them



Annotations Usage

Annotations: Bench

Annotations +

rdfs:label [language: fr]
 banc

rdfs:label [language: en]
 bench

rdfs:comment [language: en]
 long seat on which multiple people may sit at the same time

Description: Bench

Seat

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Bench

"long seat on which multiple people may sit at the same time"@en

definition url

[https://en.wikipedia.org/wiki/Bench_\(furniture\)](https://en.wikipedia.org/wiki/Bench_(furniture))

The screenshot shows the Protégé ontology editor interface. On the left, the 'Class hierarchy (inferred)' panel displays a tree structure under 'owl:Thing' with nodes for 'Seat', 'Armchair', 'Bench', 'Chair', 'Couch', and 'Stool'. The 'Bench' node is selected and highlighted in blue. In the center, the 'Annotations' tab is active, showing annotations for the 'Bench' class. It lists three rdfs:label entries: one in French ('banc') and two in English ('bench'). Below these is an rdfs:comment entry: 'long seat on which multiple people may sit at the same time'. At the bottom, the 'Description' tab is open, showing the 'Seat' class and its properties: 'General class axioms', 'SubClass Of (Anonymous Ancestor)', 'Instances', and 'Target for Key'. A diagram at the bottom right illustrates the relationships between the annotations and the original source URL.

1 Defining classes

2 Organizing them

3 Annotating them



The screenshot shows the Protégé ontology editor interface, illustrating the process of defining, organizing, and annotating classes.

Class hierarchy (inferred): Shows the inferred class hierarchy. The **Bench** class is highlighted in blue, indicating it is currently selected. Other classes shown include **owl:Thing**, **Seat**, **Armchair**, **Bench**, **Chair**, **Couch**, and **Stool**.

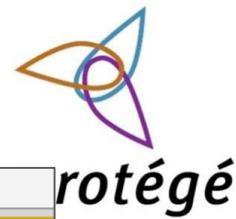
Annotations: The tab for annotations is selected. A dialog box titled "Annotations: Bench" is open, showing the properties for the **Bench** class. The **rdfs:seeAlso** property is selected, and its value is set to the URL [https://en.wikipedia.org/wiki/Bench_\(furniture\)](https://en.wikipedia.org/wiki/Bench_(furniture)). The "Type" dropdown indicates the value is of type **xsd:anyURI**.

Target for Key: A tooltip or callout box points to the **Bench** class in the class hierarchy. It contains the definition: "**"long seat on which multiple people may sit at the same time"**@en". Below this, the URL [https://en.wikipedia.org/wiki/Bench_\(furniture\)](https://en.wikipedia.org/wiki/Bench_(furniture)) is shown, with arrows indicating a relationship between the definition and the URL.

1 Defining classes

2 Organizing them

3 Annotating them



Annotations Usage

Annotations: Bench

Annotations +

- rdfs:label [language: fr]
 banc
- rdfs:label [language: en]
 bench
- rdfs:comment [language: en]
 long seat on which multiple people may sit at the same time
- rdfs:seeAlso [type: xsd:anyURI]
[https://en.wikipedia.org/wiki/Bench_\(furniture\)](https://en.wikipedia.org/wiki/Bench_(furniture))

Description: Bench

Instances +

Target for Key +

Disjoint With +

- Stool
- Couch
- Chair
- Armchair

Disjoint Union Of +

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

2. Querying my 1st K-Graph in Protégé



untitled-ontology-6 (<http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6>) : [C:\Users\roche\Nextcloud\Maria-...]

File Edit View Reasoner Tools Refactor Window Help

New... Ctrl-N

Open... Ctrl-O

Open from URL... Ctrl+Maj-O

Open recent

Save Ctrl-S

Save as... Ctrl+Maj-S

Gather ontology Save as Ctrl+Maj-G

Export inferred axioms as ontology...

Reload Ctrl+Maj-R

Edit ontology catalog file...

Loaded ontology sources...

Check for plugins...

Close window Ctrl-W

Preferences...

Exit

Properties Data properties Individuals by class DL Query SPARQL Query

Search

Bench

[language: fr]

[language: en]

[language: en]

which multiple people may sit at the same time

[type: xsd:anyURI]

[media.org/wiki/Bench_\(furniture\)](#)

Description: Bench

Equivalent To +

SubClass Of +

Seat

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

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

This screenshot shows the Protégé ontology editor interface. The main window displays a list of inferred axioms for the 'Bench' class, including its language variants and a reference to its Wikipedia page. Below this, the 'Description' tab for the 'Bench' class is open, showing various semantic relations and annotations. The 'File' menu is open, highlighting the 'Save as...' option. The top status bar indicates that no reasoner is currently selected.

2. Querying my 1st K-Graph in Protégé



Screenshot of the Protégé ontology editor showing the 'Annotations' tab for the 'Seat' class. The 'Annotations' tab is highlighted in yellow. The 'Annotations' section shows two rdfs:label entries: 'siège' (language: fr) and 'seat' (language: en). A modal dialog box is open in the center, titled 'Select an ontology format', asking to choose a format for saving the ontology. The 'RDF/XML Syntax' option is selected. At the bottom of the dialog are 'OK' and 'Annuler' buttons.

Annotations: Seat

Annotations: +

rdfs:label [language: fr]
siège

rdfs:label [language: en]

Select an ontology format

Choose a format to use when saving the 'untitled-ontology-6' ontology.

(If you are unsure as to what format to choose, we recommend that you use the standard RDF/XML format, or a widely supported format such as Turtle)

RDF/XML Syntax

OK Annuler

SubClass Of +

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

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

2. Querying my 1st K-Graph in Protégé



```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6#"
    xml:base="http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
<owl:Ontology rdf:about="http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6"/>

<!--
///////////////////////////////
// Classes
//
/////////////////////////////
-->

<!-- http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6#Armchair -->
<owl:Class rdf:about="http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6#Armchair">
    <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6#Seat"/>
    <owl:disjointWith rdf:resource="http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6#Bench"/>
    <owl:disjointWith rdf:resource="http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6#Chair"/>
    <owl:disjointWith rdf:resource="http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6#Couch"/>
    <owl:disjointWith rdf:resource="http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6#Stool"/>
    <rdfs:comment xml:lang="en">long seat on which multiple people may sit at the same time</rdfs:comment>
    <rdfs:label xml:lang="fr">banc</rdfs:label>
    <rdfs:label xml:lang="en">bench</rdfs:label>
    <rdfs:seeAlso rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">https://en.wikipedia.org/wiki/Bench\_\(furniture\)</rdfs:seeAlso>
</owl:Class>

<!-- http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6#Bench -->
<owl:Class rdf:about="http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6#Bench">
    <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6#Seat"/>
    <owl:disjointWith rdf:resource="http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6#Chair"/>
    <owl:disjointWith rdf:resource="http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6#Couch"/>
    <owl:disjointWith rdf:resource="http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6#Stool"/>
    <rdfs:comment xml:lang="en">long seat on which multiple people may sit at the same time</rdfs:comment>
    <rdfs:label xml:lang="fr">banc</rdfs:label>
    <rdfs:label xml:lang="en">bench</rdfs:label>
    <rdfs:seeAlso rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">https://en.wikipedia.org/wiki/Bench\_\(furniture\)</rdfs:seeAlso>
</owl:Class>
```

2. Querying my 1st K-Graph in Protégé



The screenshot shows the Protégé interface with several tabs at the top: Object properties, Data properties, Individuals by class, DL Query, SPARQL Query (which is circled in red), Entities, and Classes.

The main workspace displays the annotations for the class **Seat**. The left sidebar shows the class hierarchy under **owl:Thing**, with **Seat** selected. Other subclasses listed are **Armchair**, **Bench**, **Chair**, **Couch**, and **Stool**.

Annotations:

- rdfs:label** [language: fr] : siège
- rdfs:label** [language: en] : seat
- rdfs:comment** [language: en] : A thing made or used for sitting on, such as a chair or stool.

Description: Seat

General class axioms

SubClass Of (Anonymous Ancestor)

Instances

Target for Key

Disjoint With

Disjoint Union Of

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

2. Querying my 1st K-Graph in Protégé



Screenshot of the Protégé ontology editor interface showing the "untitled-ontology-76" tab selected. The "Window" menu is open, specifically the "Views" submenu under "Tabs". The "SPARQL Query" option is highlighted with a blue selection bar. Other tabs listed include Active ontology, Entities, Classes, Object properties, Data properties, Annotation properties, Individuals by class, OWLViz, DL Query, OntoGraf, SWRLTab, SQWRLTab, and SPARQL Query.

The main workspace shows the "Active ontology" tab selected, displaying the class hierarchy with "owl:Thing" as the root node. Other tabs like Entities, Classes, and Data are also visible in the tabs bar.

Below the tabs, there are several buttons for interacting with the ontology:

- Instances +
- Target for Key +
- Disjoint With +
- Disjoint Union Of +

At the bottom right, there is a note: "To use the reasoner click Reasoner > Start reasoner" and a checked checkbox labeled "Show Inferences".

2. Querying my 1st K-Graph in Protégé



untitled-ontology-6 (<http://www.semanticweb.org/roche/ontologies/2020/7/untitled-ontology-6>)

File Edit View Reasoner Tools Refactor Window Help

< > ⌂ untitled-ontology-6 Search...

Data properties x Individuals by class x DL Query x SPARQL Query x
Active ontology x Entities x Classes x Object properties x

SPARQL query:

```
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#>
SELECT ?subject ?object
      WHERE { ?subject rdfs:subClassOf ?object }
```

Execute

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

Class hierarchy: Armchair

Asserted

- owl:Thing
- Seat
 - Armchair
 - Bench
 - Chair
 - Couch
 - Stool

Annotations: Armchair

Annotations: Armchair

Description: Armchair

Equivalent To +

SubClass Of +

Seat

General class axioms +

A large blue question mark icon with a small white 3D character standing next to it, looking up at the question mark.

2. Querying my 1st K-Graph in Protégé



untitled-ontology-6 (<http://www.semanticweb.org/roche/ontologies/2020/7/untitled-6>)

File Edit View Reasoner Tools Refactor Window Help

< > untitled-ontology-6 Search...

Data properties Individuals by class DL Query SPARQL Query

Active ontology Entities Classes Object properties

SPARQL query:

```
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#>
SELECT ?subject ?object
      WHERE { ?subject rdfs:subClassOf ?object }
```

subject	object
Bench	Seat
Chair	Seat
Armchair	Seat
Stool	Seat
Couch	Seat

Execute

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

2. Querying my 1st K-Graph in Protégé



```
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#>
```

```
SELECT ?class ?name
```

```
    WHERE {      ?root rdfs:label "seat"@en.  
                 ?class rdfs:subClassOf ?root.  
                 ?class rdfs:label ?name.  
                 FILTER (lang(?name)="en")  
             }
```

```
ORDER BY ?name
```

SPARQL query:

```
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#>  
SELECT ?class ?name  
    WHERE {      ?root rdfs:label "seat"@en.  
                 ?class rdfs:subClassOf ?root.  
                 ?class rdfs:label ?name.  
                 FILTER (lang(?name)="en")  
             }
```

```
ORDER BY ?name
```

class

Annotations Usage

Annotations: Bench

- Annotations +
 - rdfs:label [[language: fr]] banc
 - rdfs:label [[language: en]] bench
 - rdfs:comment [[language: en]] long seat on which multiple people may sit at the same time

Description: Bench

- Seat

General class axioms +

Class hierarchy (inferred) Class hierarchy

Class hierarchy: Ben

- owl:Thing
- Seat
 - Armchair
 - Bench
 - Chair
 - Couch
 - Stool



No Reasoner set. Select

2. Querying my 1st K-Graph in Protégé



```
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#>
```

```
SELECT ?class ?name
```

```
    WHERE {      ?root rdfs:label "seat"@en.  
                 ?class rdfs:subClassOf ?root.  
                 ?class rdfs:label ?name.  
                 FILTER (lang(?name)="en")  
             }
```

```
ORDER BY ?name
```

SPARQL query:

```
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#>
```

```
SELECT ?class ?name
```

```
    WHERE {      ?root rdfs:label "seat"@en.  
                 ?class rdfs:subClassOf ?root.  
                 ?class rdfs:label ?name.  
                 FILTER (lang(?name)="en")  
             }
```

```
ORDER BY ?name
```

class	name
Armchair	"armchair"@en
Bench	"bench"@en
Chair	"chair"@en
Couch	"couch"@en
Stool	"stool"@en

Execute

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



2. Querying my 1st K-Graph in Protégé



SPARQL query:

```
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#>
SELECT ?name_en ?name_fr
    WHERE { ?subject rdfs:label ?name_en.
              FILTER (lang(?name_en)="en").
              ?subject rdfs:label ?name_fr.
              FILTER (lang(?name_fr)="fr") }
ORDER BY ?name_en
```

name_en	name_fr
---------	---------

Annotations Usage
Annotations: Bench

Annotations +

- rdfs:label [language: fr]
banc
- rdfs:label [language: en]
bench
- rdfs:comment [language: en]
long seat on which multiple people may sit at the same time

Description: Bench

Seat

General class axioms +

Class hierarchy (inferred)
Class hierarchy

Class hierarchy: Ben

- owl:Thing
- Seat
 - Armchair
 - Bench
 - Chair
 - Couch
 - Stool

No Reasoner set. Select



2. Querying my 1st K-Graph in Protégé



SPARQL query:

```
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#>
SELECT ?name_en ?name_fr
    WHERE { ?subject rdfs:label ?name_en.
              FILTER (lang(?name_en)="en").
              ?subject rdfs:label ?name_fr.
              FILTER (lang(?name_fr)="fr") }
ORDER BY ?name_en
```

name_en	name_fr
"armchair"@en	"fauteuil"@fr
"bench"@en	"banc"@fr
"chair"@en	"chaise"@fr
"couch"@en	"canapé"@fr
"seat"@en	"siège"@fr
"stool"@en	"tabouret"@fr

Execute

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

2. Querying my 1st K-Graph in Protégé



Display the English terms of seat with their definition in English

Draw the query graph



2. Querying my 1st K-Graph in Protégé



Display the English terms of seat with their definition in English

```
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#>
SELECT ?name ?definition
  WHERE {    ?root rdfs:label "seat"@en.
              ?class rdfs:subClassOf ?root.
              ?class rdfs:label ?name.
              ?class rdfs:comment ?definition.
              FILTER (lang(?name)="en")
}
ORDER BY ?name
```

SPARQL query:

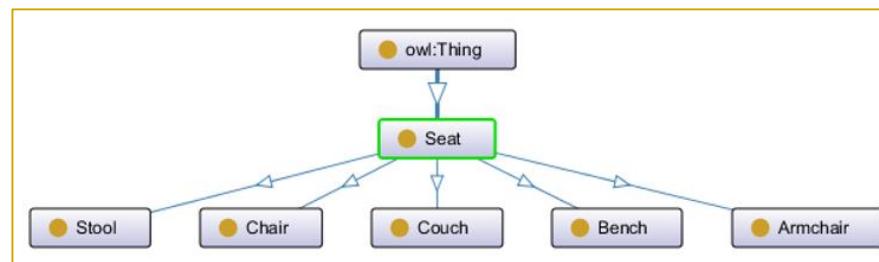
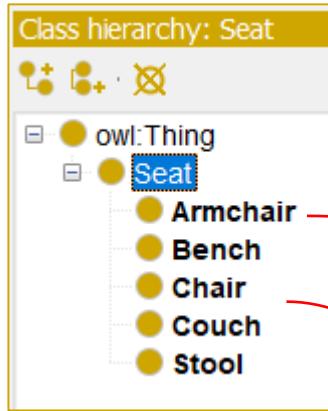
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?name ?definition
  WHERE {    ?root rdfs:label "seat"@en.
              ?class rdfs:subClassOf ?root.
              ?class rdfs:label ?name.
              ?class rdfs:comment ?definition.
              FILTER (lang(?name)="en")
}
ORDER BY ?name
```

name	definition
"armchair"@en	"Seat for one person wiht feet and back with arms"@en
"bench"@en	"Seat for several persons wiht feet without back and without arms"@en
"chair"@en	"chaise"@fr
"chair"@en	"Seat for one person wiht feet and back without arms"@en
"couch"@en	"Seat for several persons wiht feet, back and arms"@en
"stool"@en	"Seat for one person wiht feet without back without arms"@en



We need go a step further

A Step Further



Class hierarchy: Chair

Annotations:

- rdfs:label [language: en] chair

Description:

Equivalent To:

SubClass Of:

General class axioms:

SubClass Of (Anonymous Ancestor):

Instances:

Target for Key:

Disjoint With:

- Couch
- Bench
- Armchair
- Stool

Class hierarchy: Armchair

Annotations:

- rdfs:label [language: en] armchair

Description:

Equivalent To:

SubClass Of:

General class axioms:

SubClass Of (Anonymous Ancestor):

Instances:

Target for Key:

Disjoint With:

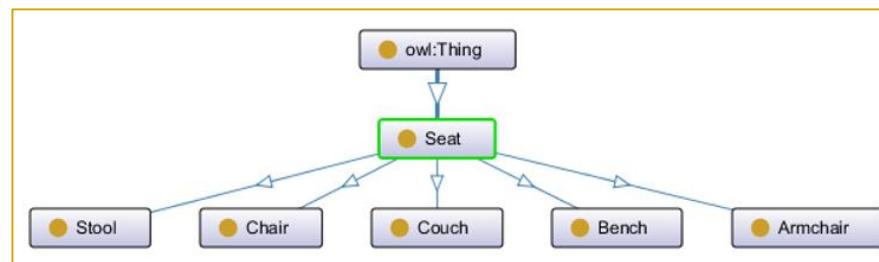
- Stool
- Bench
- Couch
- Chair

A Step Further

Class hierarchy: Seat

```

graph TD
    owlThing[owl:Thing] --> seat[Seat]
    seat --> armchair[Armchair]
    seat --> bench[Bench]
    seat --> chair[Chair]
    seat --> couch[Couch]
    seat --> stool[Stool]
  
```



Class hierarchy: Chair

Annotations:

- rdfs:label [language: en] chair

Description:

Equivalent To:

SubClass Of:

- Seat

General class axioms:

SubClass Of (Anonymous Ancestor):

Instances:

Target for Key:

Disjoint With:

- Couch
- Bench
- Armchair
- Stool

Class hierarchy: Armchair

Annotations:

- rdfs:label [language: en] armchair

Description:

Equivalent To:

SubClass Of:

- Seat

General class axioms:

SubClass Of (Anonymous Ancestor):

Instances:

Target for Key:

Disjoint With:

- Stool
- Bench
- Couch
- Chair



What is the difference between Chair and Armchair, Bench, Couch, Stool?

What is their definition?

A Step Further

Classes

```
<owl:Class rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Chair">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Seat"/>
  <rdfs:comment xml:lang="en">Seat for one person wiht feet and back without arms</rdfs:comment>
  <rdfs:comment xml:lang="fr">chaise</rdfs:comment>
  <rdfs:label xml:lang="en">chair</rdfs:label>
</owl:Class>
```

```
<owl:Class rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Armchair">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Seat"/>
  <rdfs:comment xml:lang="en">Seat for one person wiht feet and back with arms</rdfs:comment>
  <rdfs:label xml:lang="en">armchair</rdfs:label>
  <rdfs:label xml:lang="fr">fauteuil</rdfs:label>
</owl:Class>
```

General axioms

```
<rdf:Description>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AllDisjointClasses"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Armchair"/>
    <rdf:Description rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Bench"/>
    <rdf:Description rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Chair"/>
    <rdf:Description rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Couch"/>
    <rdf:Description rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Stool"/>
  </owl:members>
</rdf:Description>
```

A Step Further

Classes

```
<owl:Class rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Chair">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Seat"/>
  <rdfs:comment xml:lang="en">Seat for one person wiht feet and back without arms</rdfs:comment>
  <rdfs:comment xml:lang="fr">chaise</rdfs:comment>
  <rdfs:label xml:lang="en">chair</rdfs:label>
</owl:Class>
```

```
<owl:Class rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Armchair">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Seat"/>
  <rdfs:comment xml:lang="en">Seat for one person wiht feet and back with arms</rdfs:comment>
  <rdfs:label xml:lang="en">armchair</rdfs:label>
  <rdfs:label xml:lang="fr">fauteuil</rdfs:label>
</owl:Class>
```

General axioms

```
<rdf:Description>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AllDisjointClasses"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Armchair"/>
    <rdf:Description rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Bench"/>
    <rdf:Description rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Chair"/>
    <rdf:Description rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Couch"/>
    <rdf:Description rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Stool"/>
  </owl:members>
</rdf:Description>
```

Class hierarchy: Bag

- owl:Thing
- 311ROThing
- 311Type
- Action
- Agency
- LocationType
- MessageCategory
- ServiceRequest
- SPSPoint
- Subject
- GarbageContainer
- Bag
- Basket
- Bin
- Box
- Cardboard
- Cart
- Pile
- Waste
- Pest
- Plants
- RoadSymbol
- Sign
- Vehicle
- Feature
- foaf:Person

Annotations: Bag

Annotations: Bag

rdfs:label [language: en]
Bag

Description: Bag

Equivalent To +

SubClass Of +

GarbageContainer

General class axioms +

Instances +

Target for Key +

To use the reasoner click Reasoner > Start reasoner Show Inferences

Description: Bag

Equivalent To +

SubClass Of +

GarbageContainer

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

GarbageContainer

Bag

Basket

Bin

Box

Cardboard

Cart

Pile

Waste

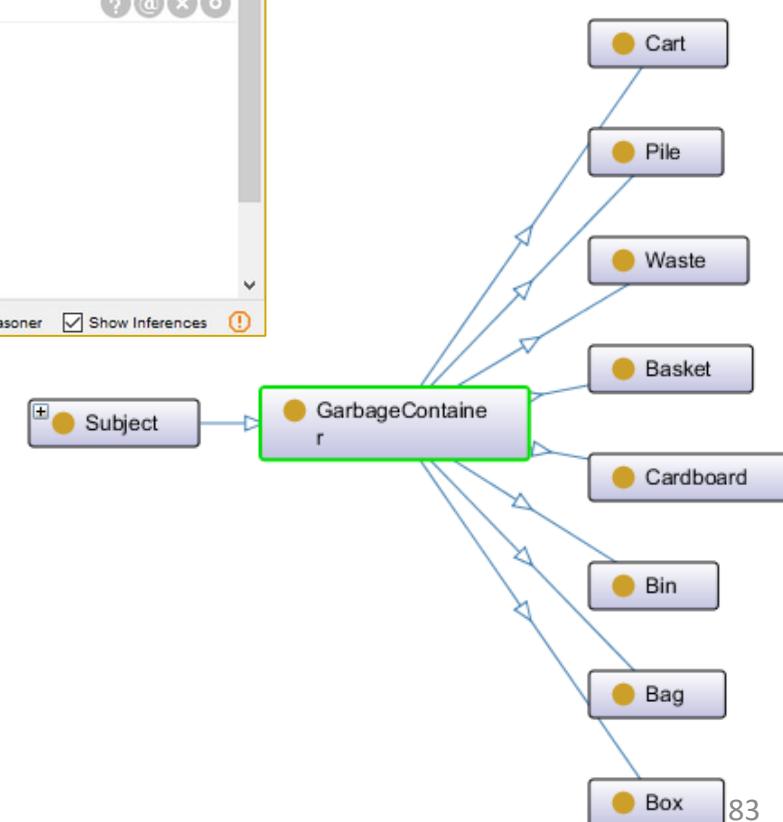
Description: Box

Equivalent To +

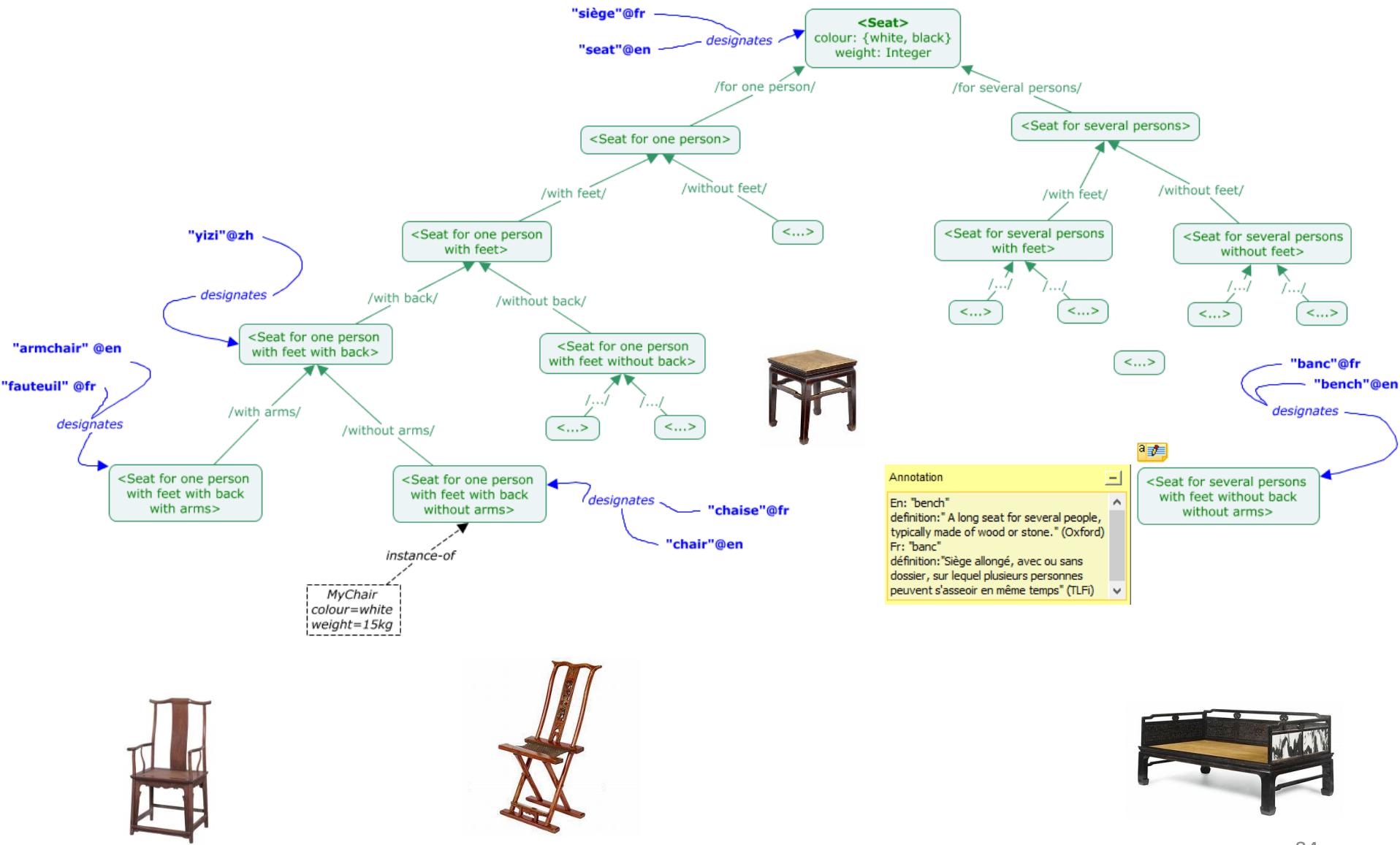
SubClass Of +

GarbageContainer

General class axioms +



Ontoterminology of Seats



Array of differences

Objects	Concepts	Axis of analysis		Axis of analysis		Axis of analysis		Axis of analysis		Terms	
Objects	Concepts	for one person	several persons	with feet	without feet	with back	without back	with arms	without arms	Designations (English)	Designations (French)
	<Seat 1 person with feet with back without arms>	X		X		X			X	"chair"	"chaise"
	<Seat 1 person with feet with back with arms>	X		X		X		X		"armchair"	"fauteuil"
	<Seat 1 person with feet without back without arms>	X		X			X		X	"stool"	"tabouret"
	<Seat several persons with feet with back with arms>		X	X		X		X		"couch"	"canapé"
	<Seat several persons with feet without back without arms>		X	X			X		X	"bench"	"banc"

"chair" : Seat for one person with feet and back without arms.



↳ <Seat for one person with feet with back without arms>

::= <Seat> + /for one person/ + /with feet/ + /with back/ + /without arms/

"armchair" : Seat for one person with feet and back with arms.



↳ <Seat for one person with feet with back with arms>

::= <Seat> + /for one person/ + /with feet/ + /with back/ + /with arms/

"bench" : Seat for several persons with feet, without back, and without arms.

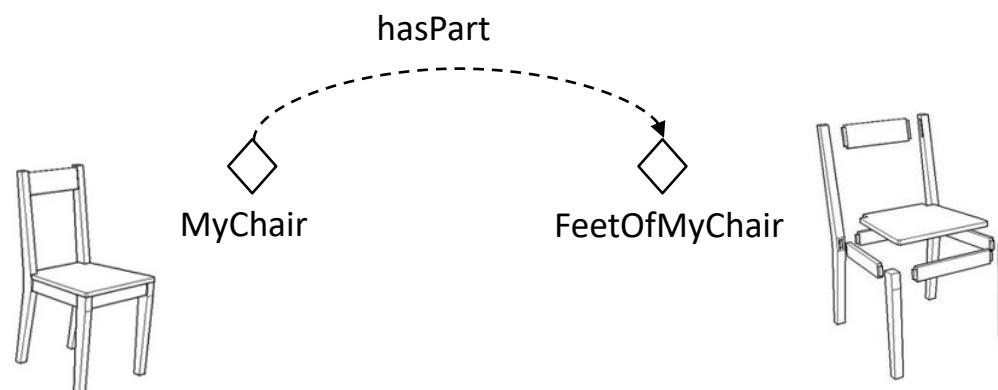
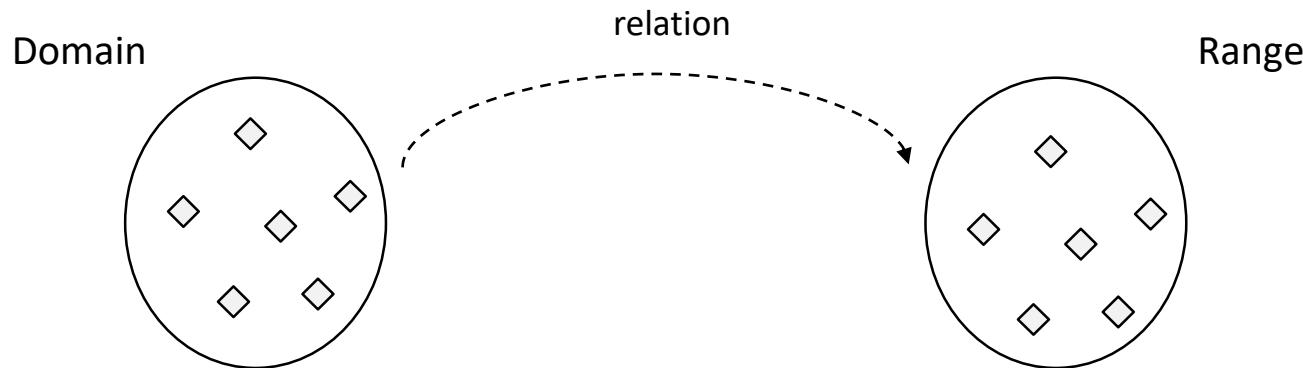


↳ <Seat for one person with feet with back without arms>

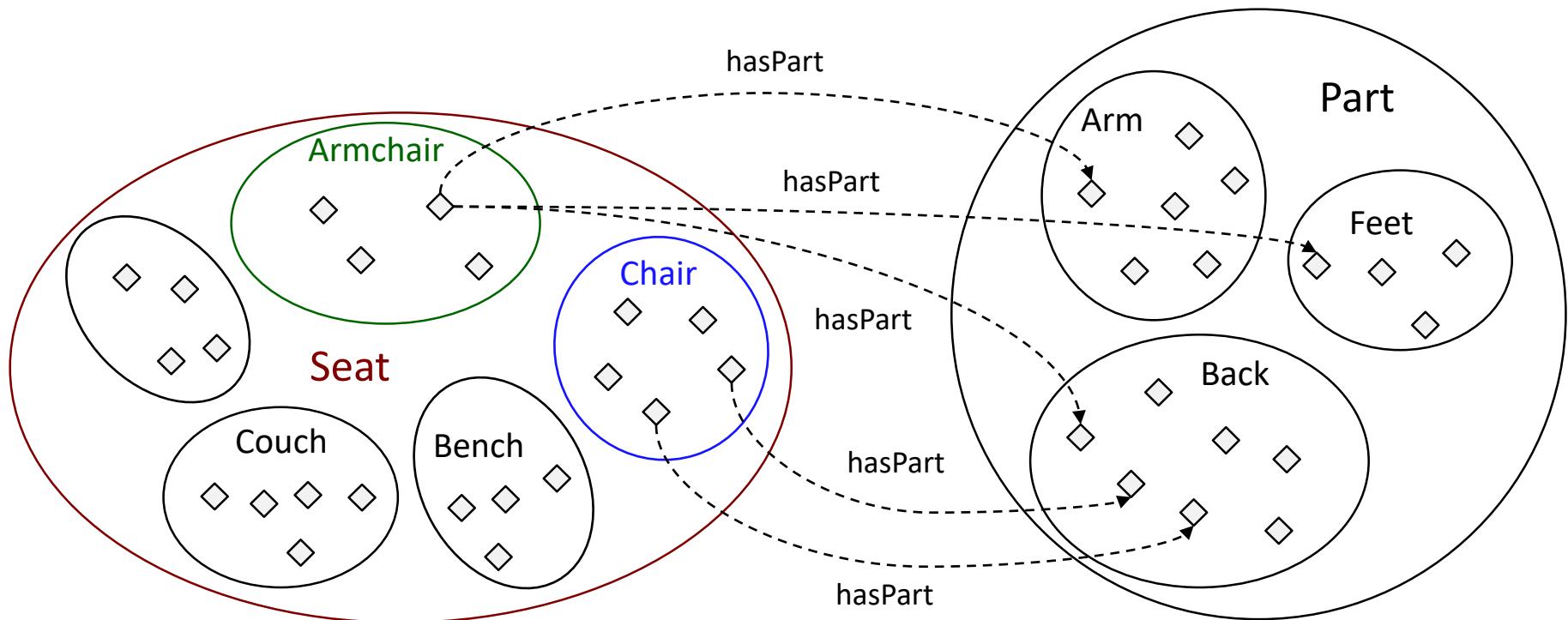
::= <Seat> + /for one person/ + /with feet/ + /with back/ + /without arms/

Building an OWL Ontology using Protégé: Object Properties

Relationships between individuals



Building an OWL Ontology using Protégé: Object Properties



Building an OWL Ontology using Protégé: Defining Classes

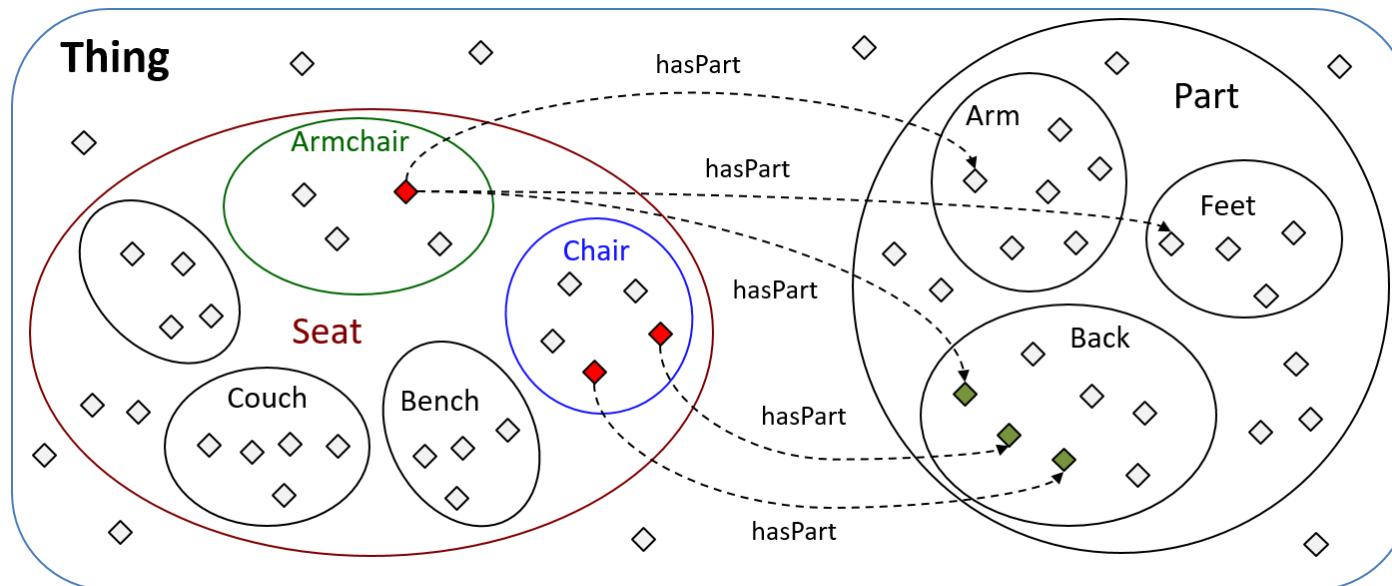
(Object or Data) **Properties Restrictions** can be used to define classes

The key idea is that a class of individuals is described or defined by the relationships that these individuals participate in

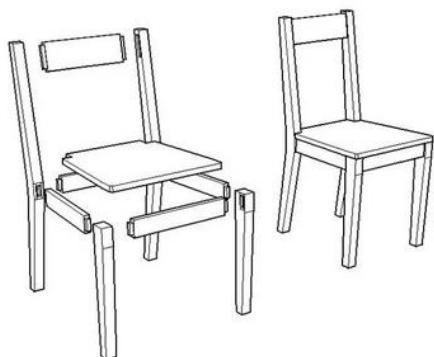
A **restriction** describes an **anonymous class** (an unnamed class).

The anonymous class contains all of the individuals that satisfy the restriction (i.e. all of the individuals that have the relationships required to be a member of the class).

$$\text{SeatWithBack} = \{ x / \exists y \text{ Back}(y) \wedge \text{hasPart}(x,y) \}$$



Building an OWL Ontology using Protégé: Object Properties



hasPart



MyChair

FeetOfMyChair

Object property hierarchy: hasPart

owl:topObjectProperty hasPart

untitled-ontology-22 (<http://www.semanticweb.org/croche/ontologies/2020/4/u...>) — X

File Edit View Reasoner Tools Refactor Window Help

Object properties x Individuals by class x OntoGraf x SPARQL Query x Data properties x Active ontology x Entities x Classes x

Object property hierarchy: hasPart Asserted Annotations Usage
Annotations: hasPart
Annotations +

Characteri Description: hasPart
 Functional Equivalent To +
 Inverse functio SubProperty Of +
 Transitive Inverse Of +
 Symmetric Domains (intersection) +
 Asymmetric Seat
 Reflexive Ranges (intersection) +
 Irreflexive Part
 Disjoint with +
 SuperProperty Of (Chain) +

To use the reasoner click Reasoner > Start reasoner Show Inferences

A screenshot of the Protégé ontology editor interface. The title bar shows the ontology name "untitled-ontology-22". The menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The toolbar has standard icons for back, forward, search, and other functions. The main workspace is divided into several tabs: "Object properties" (highlighted with a blue oval), "Individuals by class", "OntoGraf", "SPARQL Query", "Data properties", and "Active ontology" (highlighted with a blue oval). The "Object properties" tab shows the hierarchy: "owl:topObjectProperty" is the root, followed by "hasPart". The "Annotations" tab for "hasPart" shows it is transitive. The "Characteri" tab shows "hasPart" is functional and has domains like "Seat" and "Part". A small inset image in the bottom left corner shows a person working at a computer.



Building an OWL Ontology using Protégé: Data Properties

relationships between an individual and data values.



MyChair color: "brown"

untitled-ontology-22 (<http://www.semanticweb.org/croche/ontologie...>) — X

File Edit View Reasoner Tools Refactor Window Help

< > untitled-ontology-22 Search...

color

Individuals by class x OntoGraf x SPARQL Query x Data properties x

Active ontology x Entities x Classes x Object properties x

Data property hierarchy: color Annotations Usage

owl:topDataProperty color

Annotations: color Annotations +

Characteristics x x x Description: color x x x

Functional

Equivalent To +

SubProperty Of +

Domains (intersection) +

Seat

Ranges +

xsd:string

Disjoint With +

To use the reasoner click Reasoner > Start reasoner Show Inferences 90

The screenshot shows the Protégé ontology editor interface. The main window displays a navigation bar with tabs for 'Data properties' (circled in blue), 'Annotations', and 'Usage'. Below the navigation bar, there are tabs for 'Annotations' and 'Annotations: color'. The left sidebar shows a 'Data property hierarchy' for the 'color' property, which is a subproperty of 'owl:topDataProperty'. The right sidebar shows the 'Description' tab for the 'color' property, which is marked as 'Functional'. It also lists 'Equivalent To', 'SubProperty Of', 'Domains (intersection)' (with 'Seat' listed), 'Ranges' (with 'xsd:string' listed), and 'Disjoint With'. A green circle highlights the 'Functional' checkbox in the characteristics section, and a large green circle highlights the 'Ranges' section where 'xsd:string' is listed.



Building an OWL Ontology using Protégé: Properties Restrictions

Existential Restrictions

An **existential restriction** describes a **class of individuals** that have **at least one (some) relationship** along a **specified property** to an **individual** that is a member of a **specified class**.

$$\text{SeatWithBack} = \{ x / \exists y \text{ Back}(y) \wedge \text{hasPart}(x,y) \}$$

The screenshot shows the Protégé interface with the following components:

- Left Panel (Class hierarchy):** Shows the class hierarchy under "Active ontology". It includes owl:Thing, Part, Arms, Back, Feet, and Seat. Under Seat, there are sub-classes: SeatWithBack, Armchair, Bench, Chair, Couch, and Stool. The "SeatWithBack" node is highlighted.
- Middle Panel (Annotations):** The "Annotations" tab is selected, showing the annotation "Annotations: SeatWithBack".
- Bottom Panel (SubClass Of):** The "SubClass Of" section shows the asserted restriction: "hasPart some Back". A blue circle highlights the "+" button used to add this restriction.
- Right Panel (SeatWithBack Class Editor):** An open dialog for the "SeatWithBack" class. It contains:
 - Class expression editor:** Shows the class definition.
 - Object restriction creator:** Shows the asserted restriction "hasPart some Back".
 - Restricted property:** Shows "owl:topObjectProperty hasPart".
 - Restriction filler:** Shows the class hierarchy under "owl:Thing" including Part, Arms, Back, Feet, and Seat.

Building an OWL Ontology using Protégé: Class Expressions

Class expressions are used to describe individuals that share common characteristics.

Class Expression Syntax

<http://protegeproject.github.io/protege/class-expression-syntax/>

Keyword	Example	Intuitive Meaning
some	hasPet some Dog	<p>Things that have a pet that is a Dog</p> <p>This is the most common type of class expression. Also known as, an "SomeValueFrom restriction" or an "Existential Restriction". This kind of class expression consists of a property (in this case hasPet) and a class expression that is known as a filler (in this case the filler is Dog).</p> <p>Individuals that are instances of this class expression have a relationship along the hasPet property to an individual that is an instance of class Dog.</p>
value	hasPet value Tibbs	<p>Things that have a pet that is Tibbs.</p> <p>Here, Tibbs is a specific individual. Intuitively this would describe Tibb's owners. Note that this is a short cut for, and logically equivalent to, (hasPet some {Tibbs}), where the curly brackets describe a class of specific individuals - in this case, a class of one individual that is Tibbs. Also known as a "HasValue restriction"</p>
only	hasPet only Cat	<p>Things that have pets that are only Cats.</p> <p>Note that this does not mean that these things must have a Cat, but if they do have a pet then it will be a Cat. Also known as an "AllValuesFrom restriction" or a "Universal restriction"</p>
min	hasPet min 3 Cat	<p>Things that have at least three pets that are Cats.</p> <p>Things that have at least three pets that are Cats. Also known as a "Min cardinality restriction"</p>

Essential Characteristics: Property Restriction



Class hierarchy: Armchair

Annotations: Usage

Annotations: Armchair

Annotations +

- rdfs:label [language: fr] fauteuil
- rdfs:label [language: en] armchair

Description: Armchair

Equivalent To +

SubClass Of +

- hasPart some Arms
- hasPart some Back
- hasPart some Feet
- Seat

SubClass Of +

- hasPart exactly 1 Back
- hasPart exactly 2 Arm
- hasPart some Feet
- seat



Without arms?



Class hierarchy: Chair

Annotations: Usage

Annotations: Chair

Annotations +

- rdfs:label [language: en] chair
- rdfs:comment [language: fr] chaise

Description: Chair

Equivalent To +

SubClass Of +

- hasPart some Back
- hasPart some Feet
- not (hasPart some Arms)
- Seat

Essential Characteristics: Class

Seat for several persons?

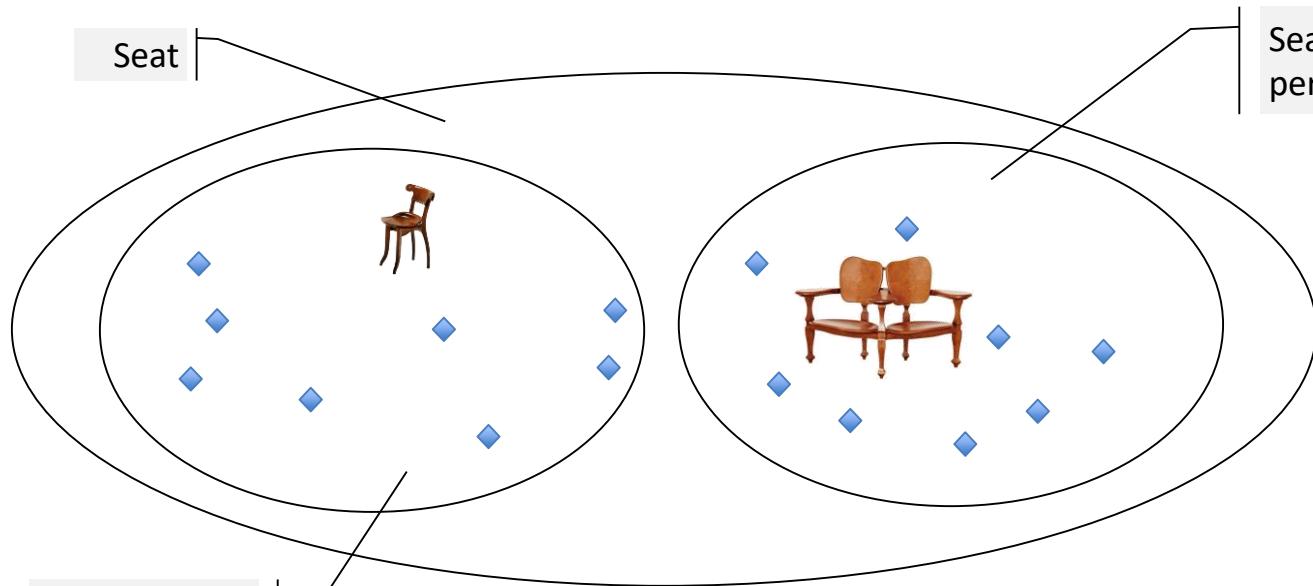


Seat

Seat for several person



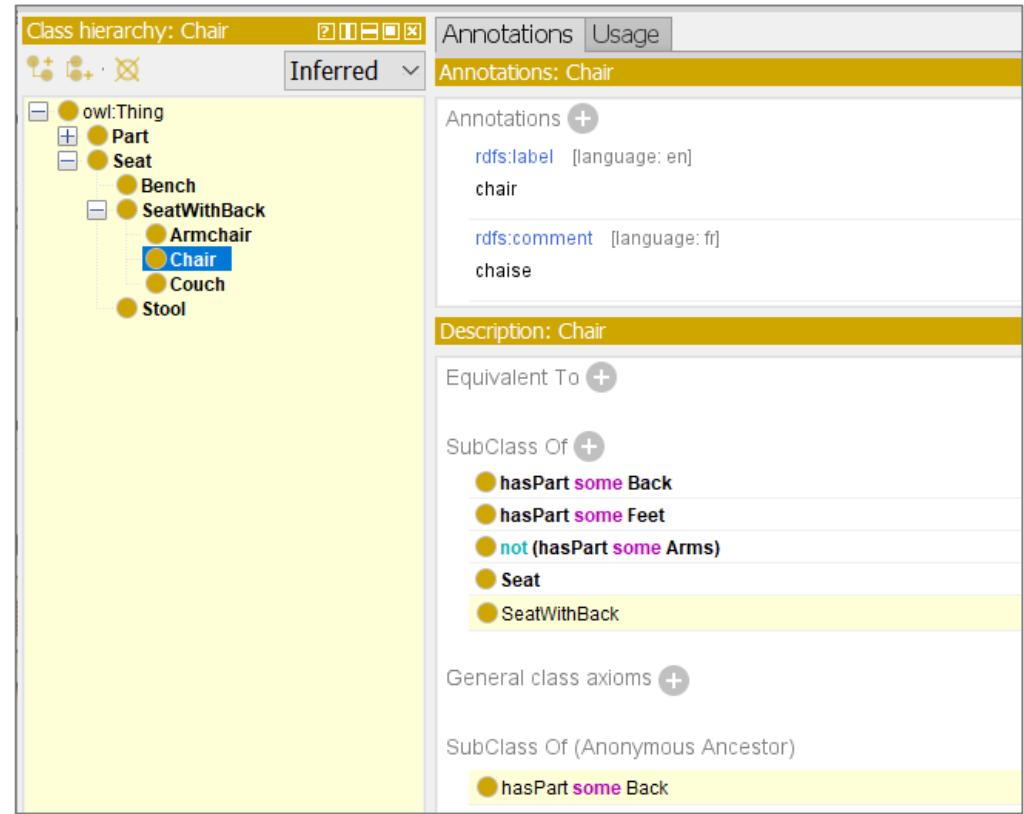
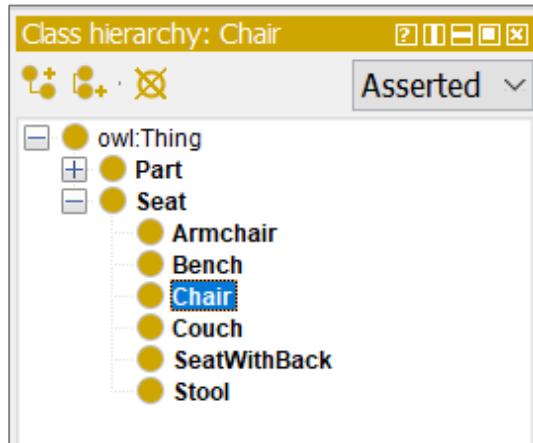
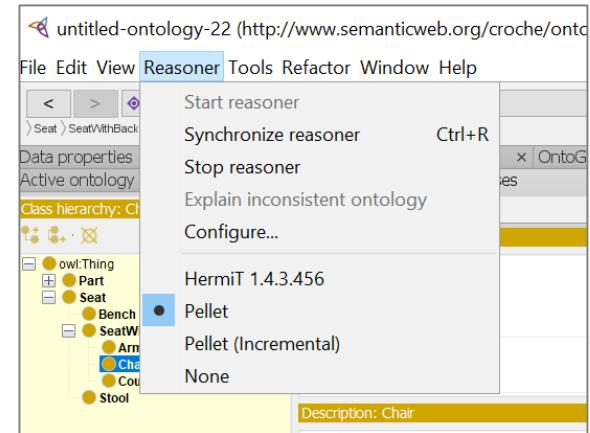
Seat for one person



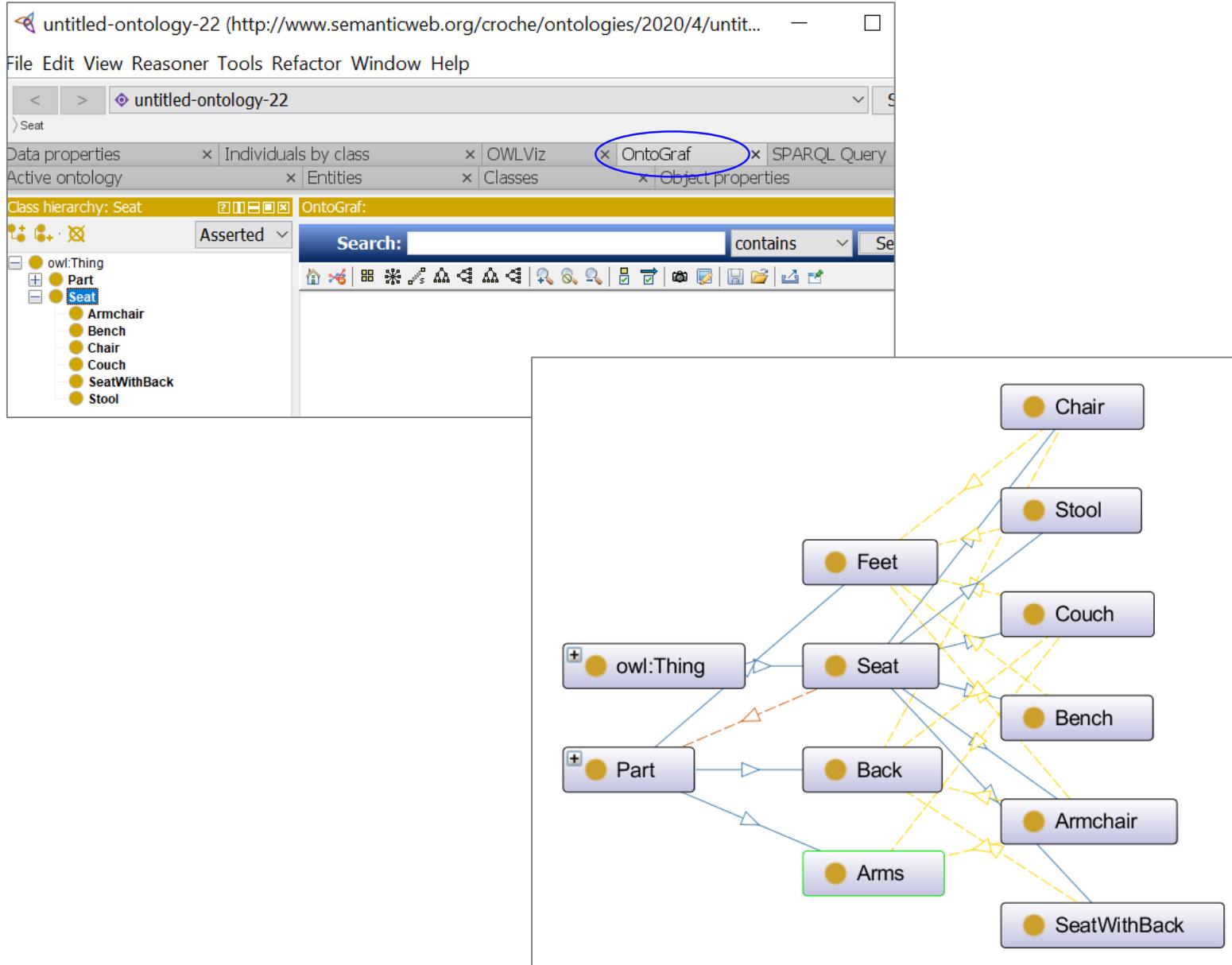
Building an OWL Ontology using Protégé: Reasoner

Protégé 4 allows different OWL reasoners to be plugged in

The class hierarchy that is automatically computed by the reasoner is called the inferred hierarchy.



Building an OWL Ontology using Protégé: Visualisation



Building an OWL Ontology using Protégé: SPARQL Query

```
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#>
SELECT ?specificClass ?genericClass
    WHERE { ?specificClass rdfs:subClassOf ?genericClass }
ORDER BY ?specificClass
```

The screenshot shows the Protégé interface with a SPARQL query results table. The query is:

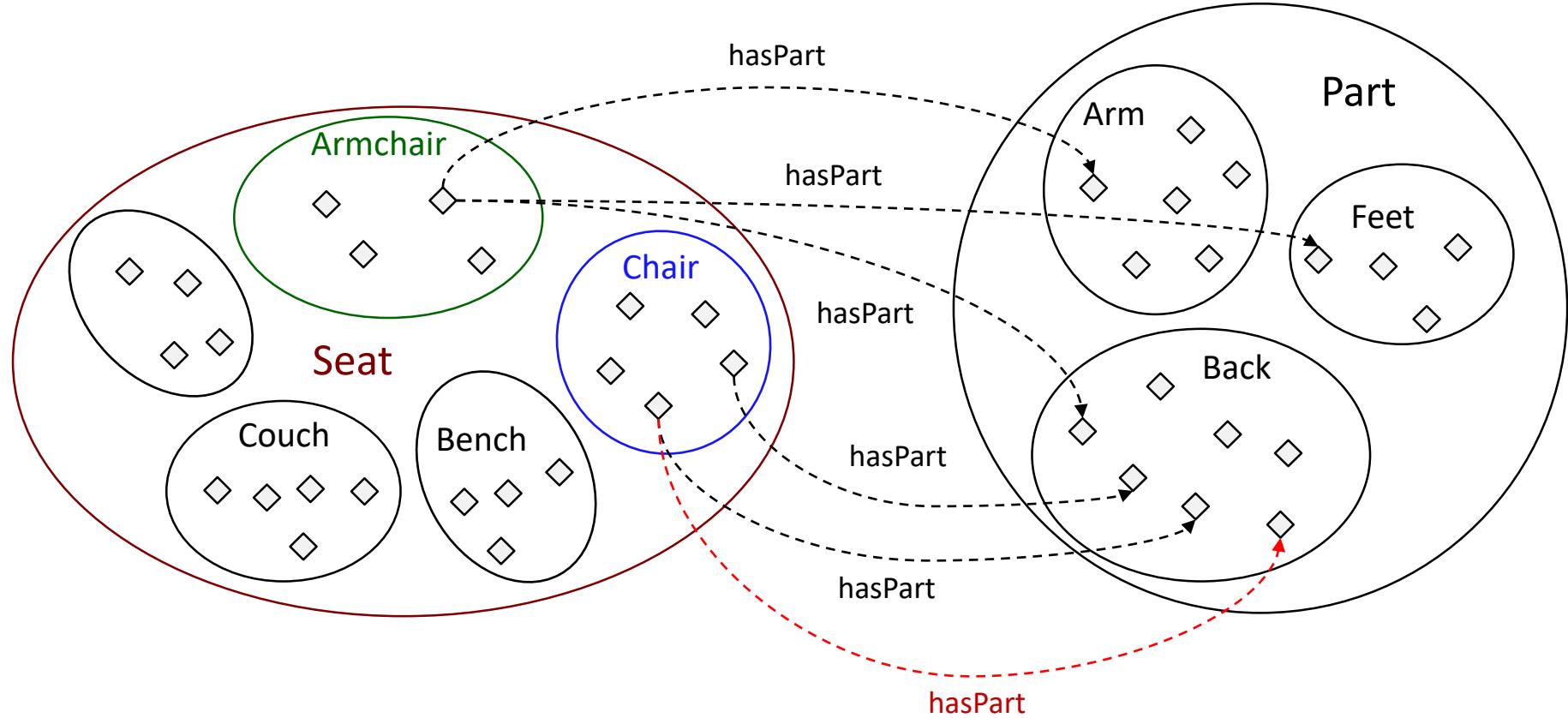
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?specificClass ?genericClass
    WHERE { ?specificClass rdfs:subClassOf ?genericClass }
ORDER BY ?specificClass
```

The results table has two columns:

specificClass	genericClass
Armchair	hasPart some Back
Armchair	hasPart some Feet
Armchair	hasPart some Arms
Armchair	Seat
Armchair	hasPart2 value with-back
Arms	Part
Back	Part

At the bottom right of the table is a "Execute" button.

Building an OWL Ontology using Protégé: Essential Characteristics



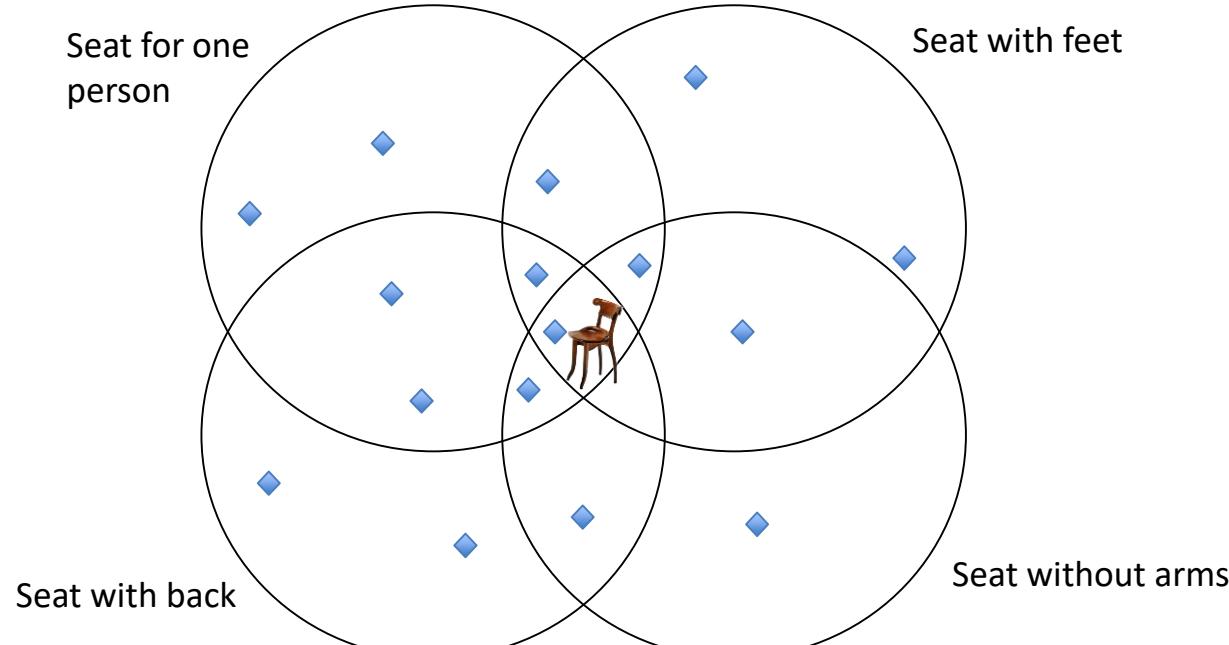
Seat for one person?

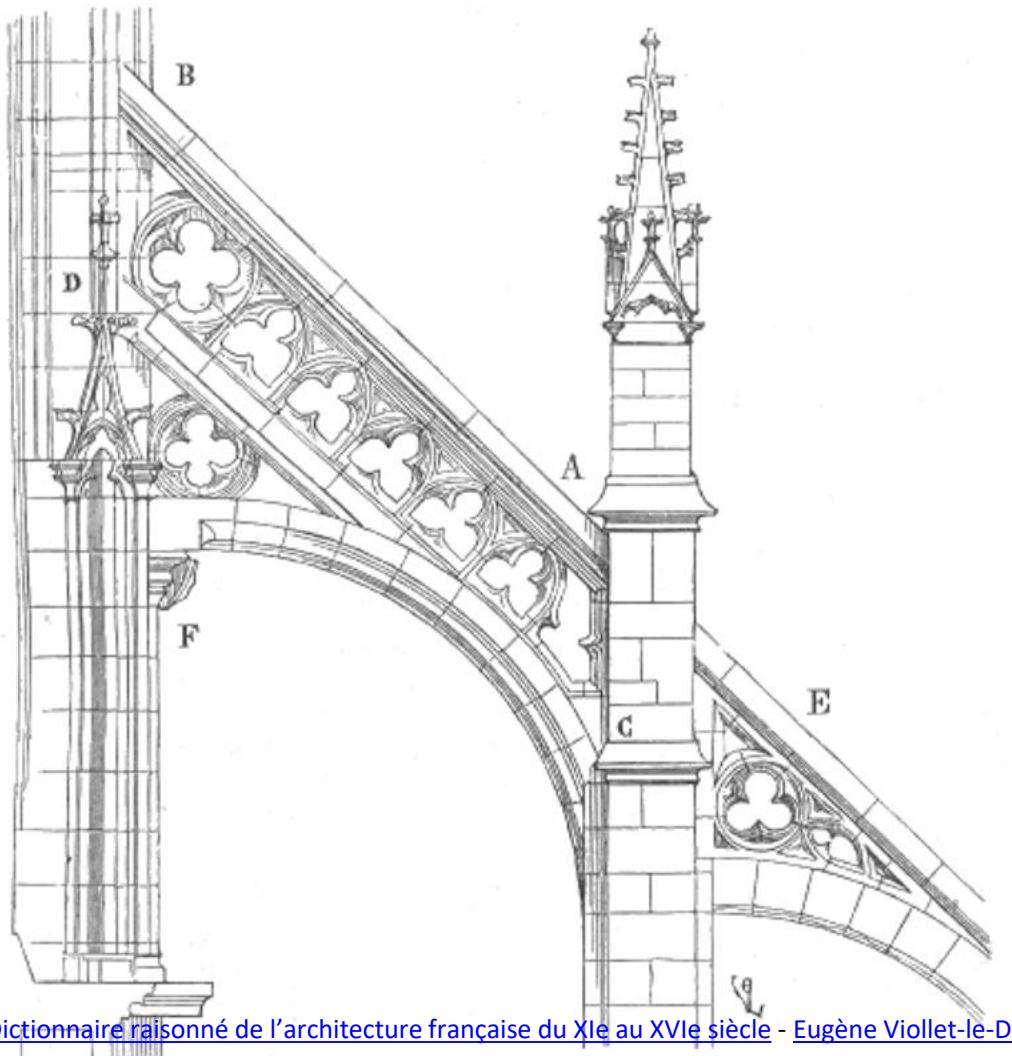


Building an OWL Ontology using Protégé: Essential Characteristics



- Essential characteristic == Class
- Essential characteristic == Role restriction
- Essential characteristic == individual





Prof. Christophe Roche

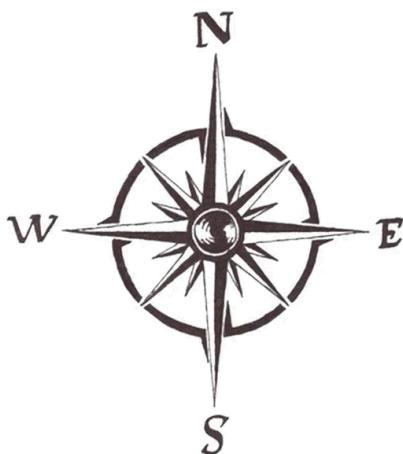
[Condillac Research Group – LISTIC Lab.](#)
University Savoie Mont-Blanc (France)

[KETRC Research Centre](#)
University of Liaocheng (China)

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

Contents

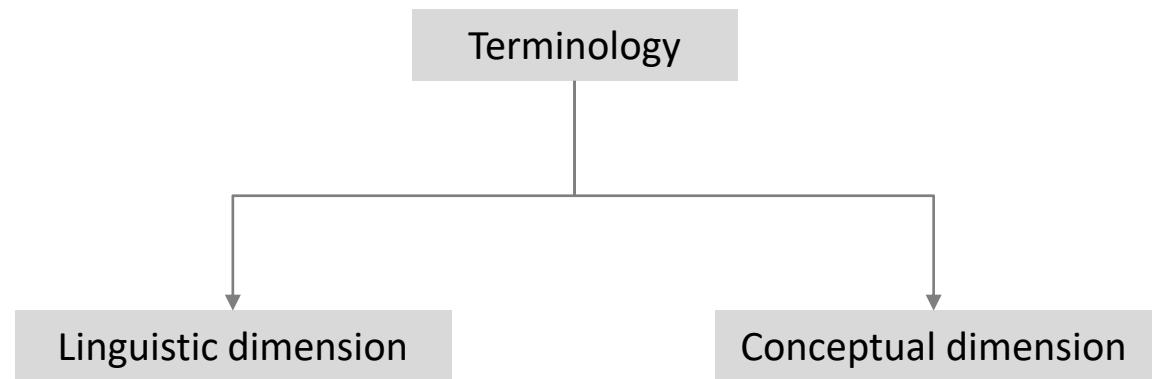
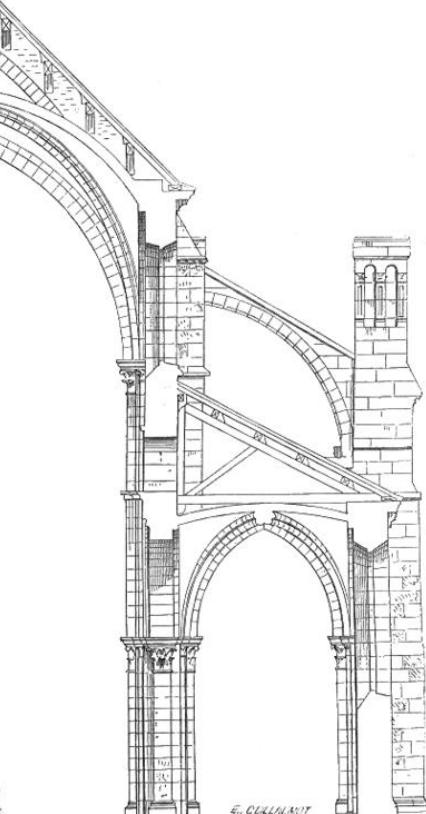
- 1) Theoretical Foundations
 - 1) Conceptual Dimension
 - 2) Linguistic Dimension
- 2) Environment
- 3) Term-guided Methodology
- 4) Export
- 5) Ontoterminology & W3C



(1) Theoretical Foundations

- 1) Theoretical Foundations
- 2) Environment
- 3) Methodology
- 4) Export

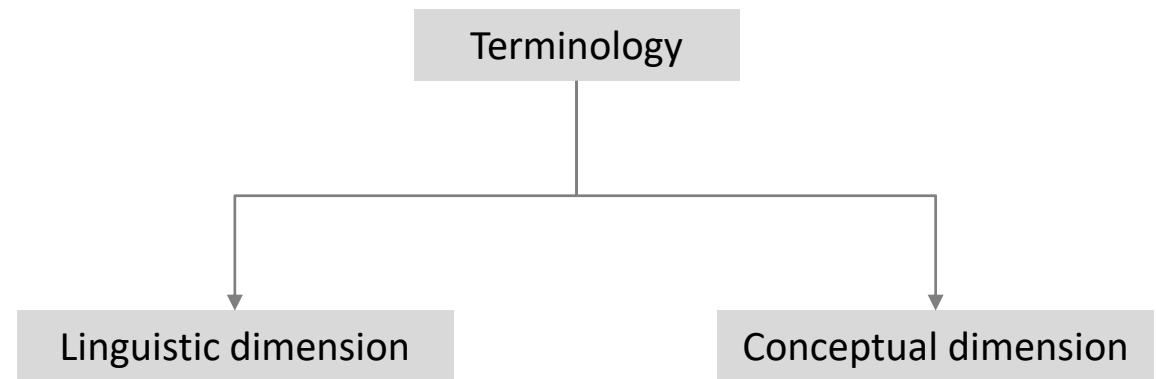
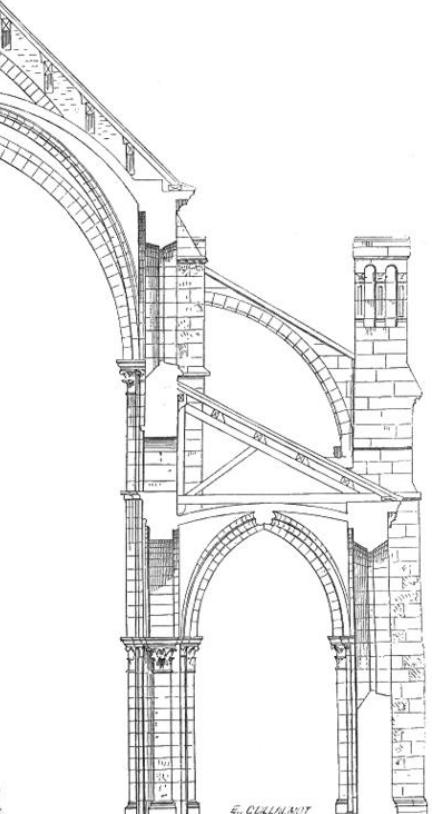
- An ontoterminology is a terminology whose conceptual system is a formal ontology



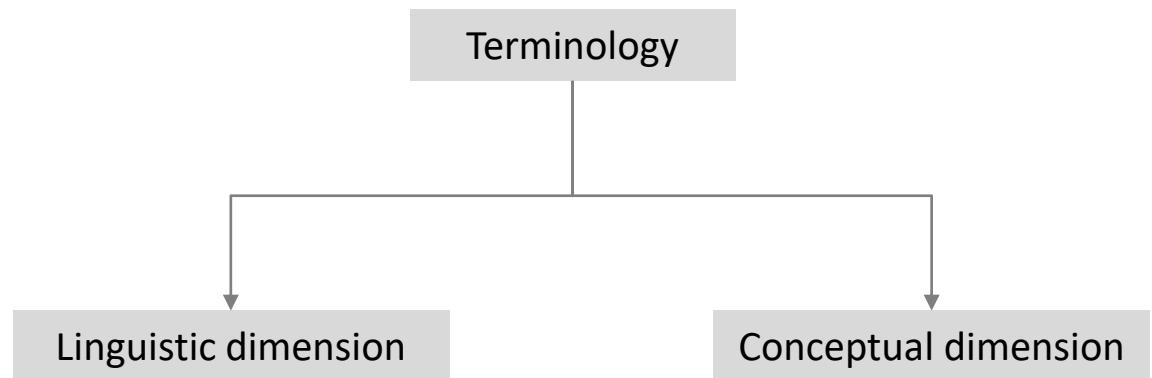
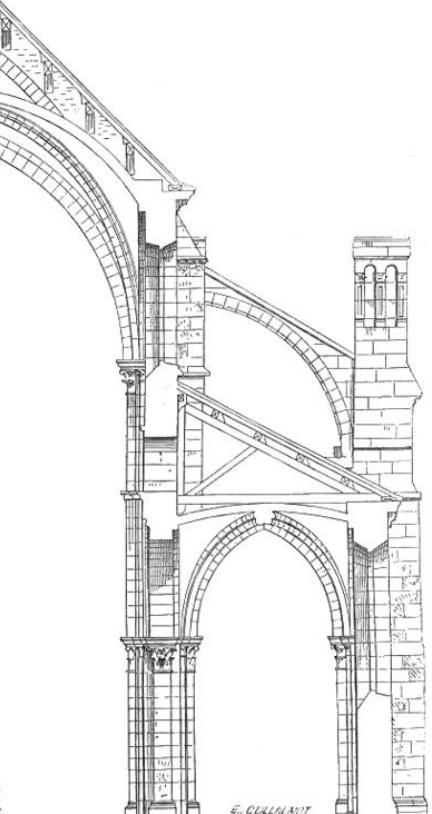
(1) Theoretical Foundations

- 1) Theoretical Foundations
- 2) Environment
- 3) Methodology
- 4) Export

- An ontoterminology is a terminology whose conceptual system is a formal ontology
- A term is a verbal designation of a concept



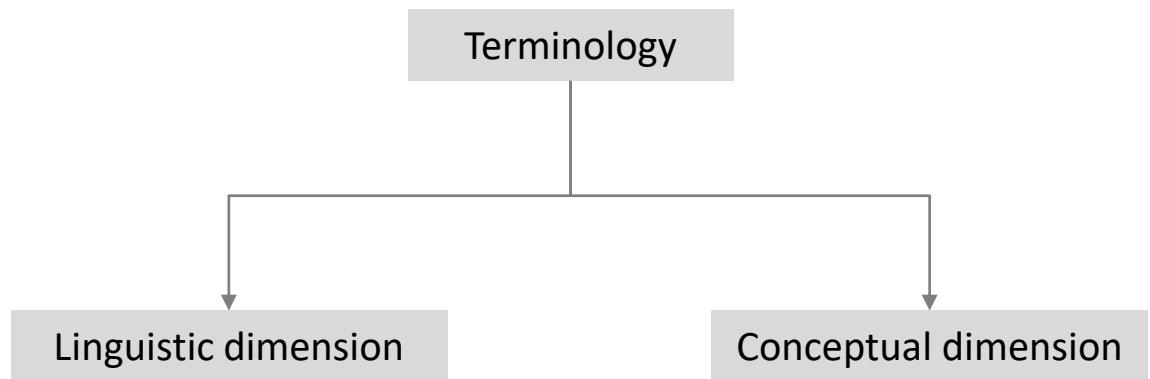
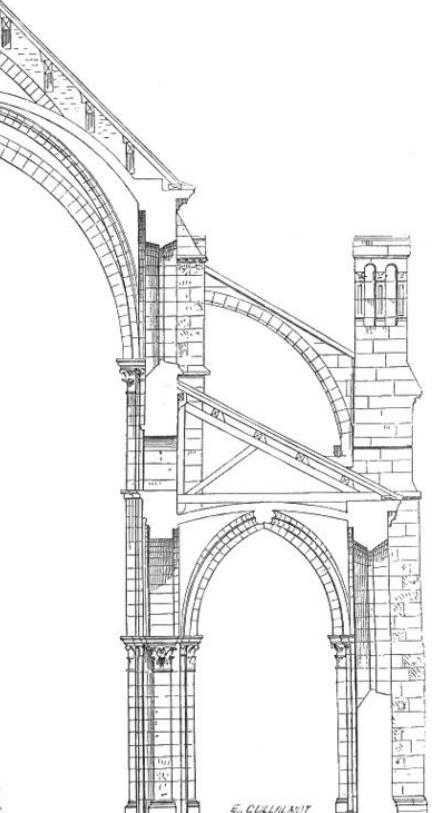
- An ontoterminology is a terminology whose conceptual system is a formal ontology
- A term is a verbal designation of a concept
- A concept is a unique combination of essential characteristics



(1) Theoretical Foundations

- 1) Theoretical Foundations
- 2) Environment
- 3) Methodology
- 4) Export

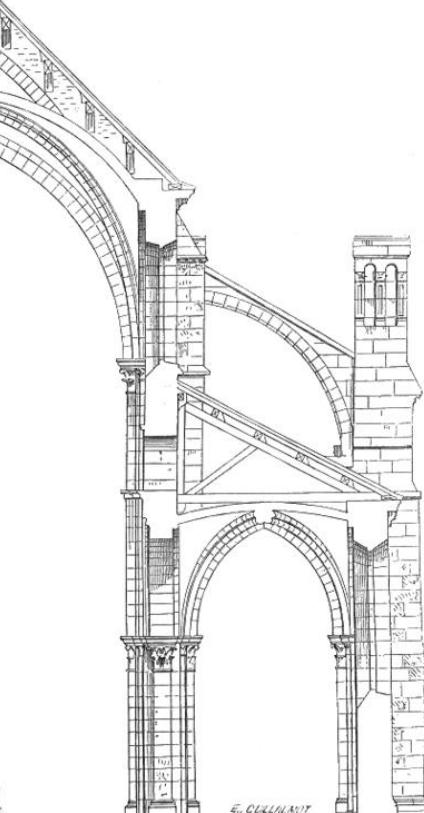
- An ontoterminology is a terminology whose conceptual system is a formal ontology
- A term is a verbal designation of a concept
- A concept is a unique combination of essential characteristics
- A concept is a set of essential characteristics enough stable to be named in a given natural language



Conceptual dimension

- A concept is defined as a unique combination of essential characteristics

<Seat for one person> ::= <Seat> + /one person/
<Seat with feet> ::= <Seat> + /with feet/
<Seat with back> ::= <Seat> + /with back/
etc.



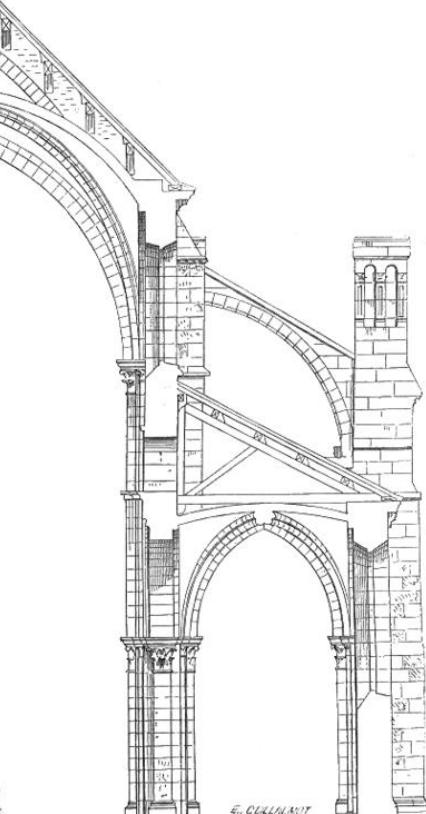
Conceptual dimension

- A concept is defined as a unique combination of essential characteristics

<Seat for one person> ::= <Seat> + /one person/
<Seat with feet> ::= <Seat> + /with feet/
<Seat with back> ::= <Seat> + /with back/
etc.

- Essential characteristics are structured into axes of analysis and are exclusive each other

Back = { /with back/ /without back/ }
Feet = { /with feet/ /without feet/ }
etc.



Conceptual dimension

- A concept is defined as a unique combination of essential characteristics

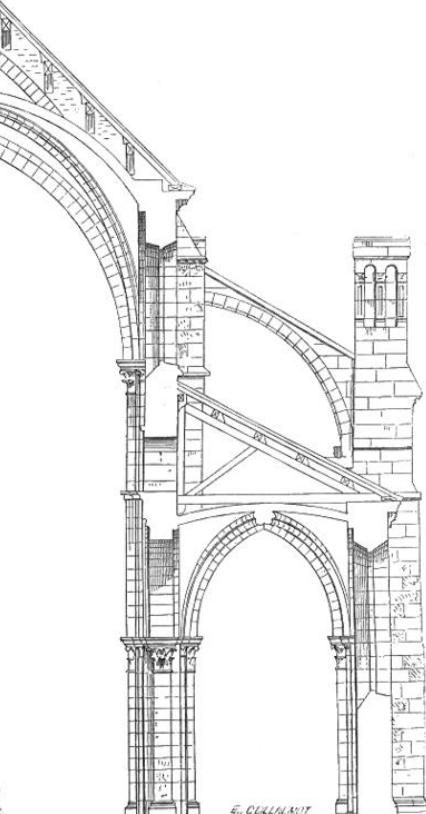
<Seat for one person> ::= <Seat> + /one person/
<Seat with feet> ::= <Seat> + /with feet/
<Seat with back> ::= <Seat> + /with back/
etc.

- Essential characteristics are structured into axes of analysis and are exclusive each other

Back = { /with back/ /without back/ }
Feet = { /with feet/ /without feet/ }
etc.

- Multiple hierarchy => multiple inheritance

<Seat for one person with feet with back without arm>
is-a <Seat for one person>
is-a <Seat with feet>
is-a <Seat with back>
etc.

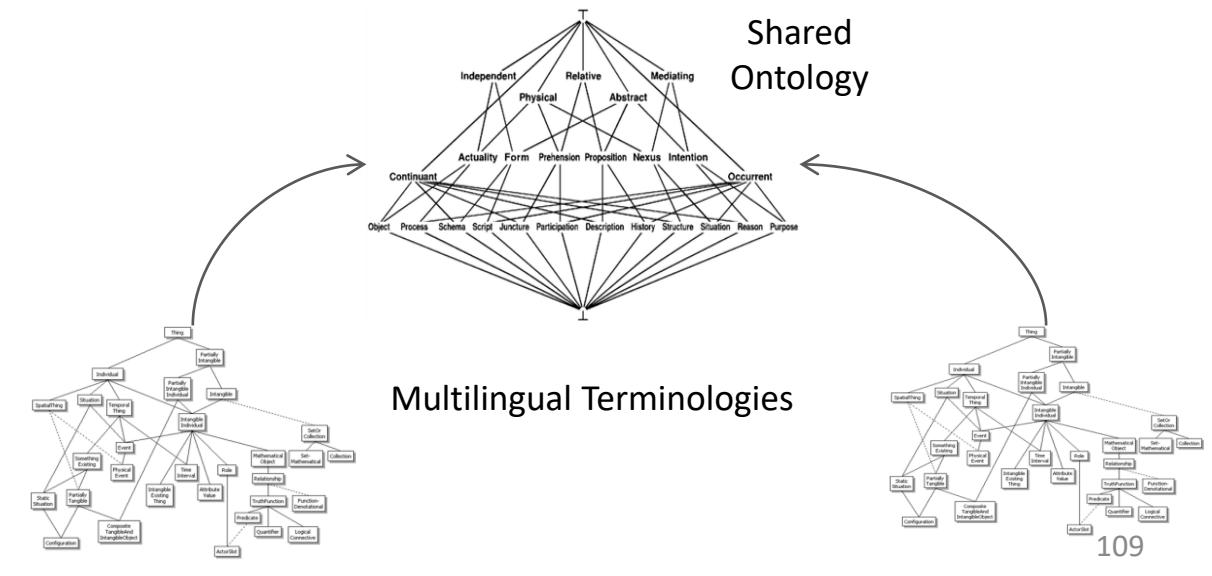
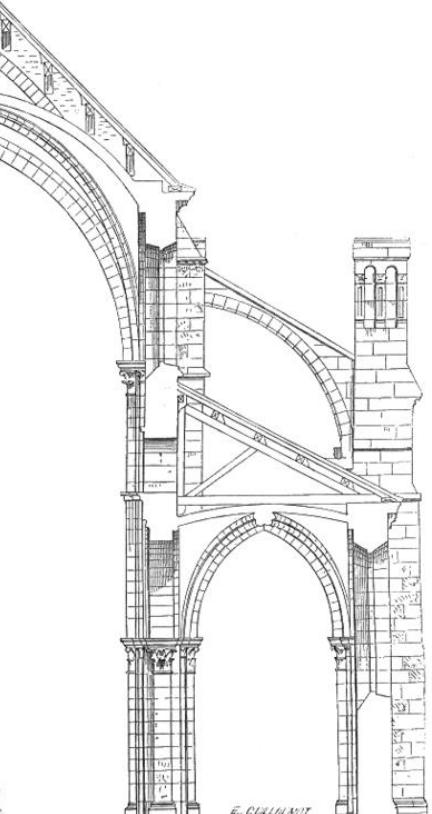


(1) Theoretical Foundations

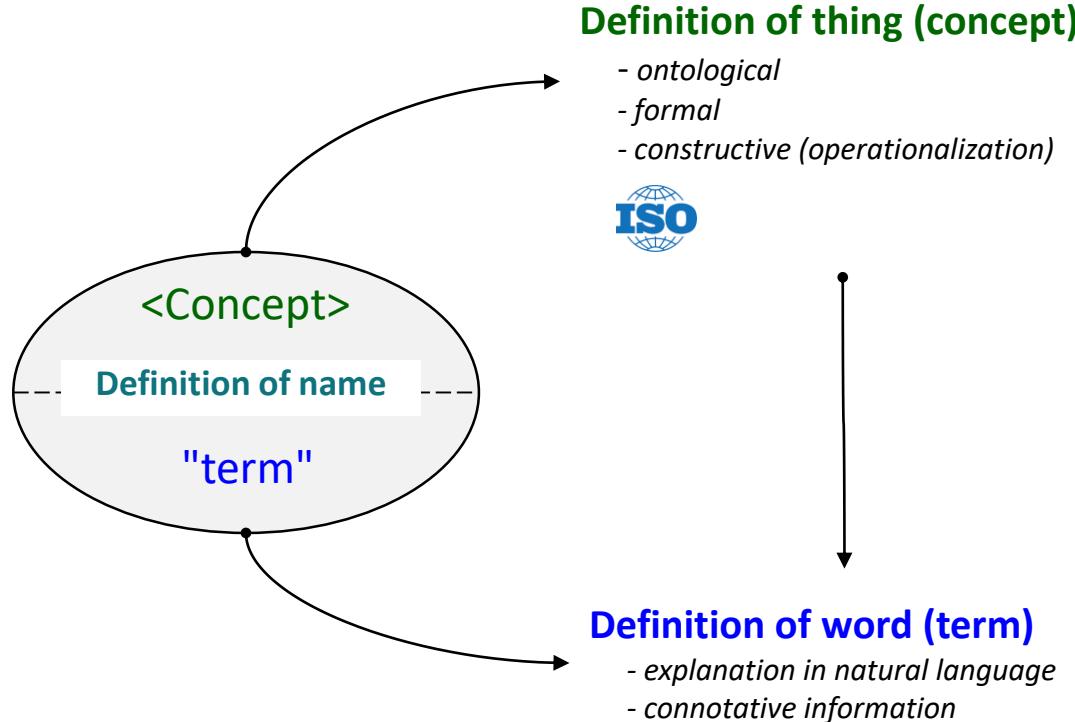
- 1) Theoretical Foundations
- 2) Environment
- 3) Methodology
- 4) Export

Linguistic dimension

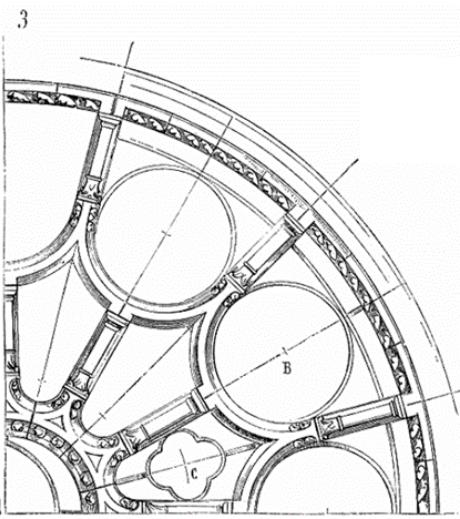
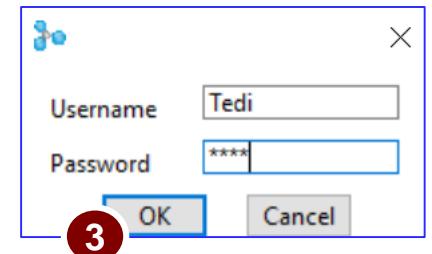
- Multi-lingual terminology
- Linguistic diversity
- Ontology shared by different terminologies
- Aristotelian definition in natural language (genus and specific difference) based on the formal definition



Ontoterm



■ Opening a session



■ Launcher

Tedi - Multilingual Ontoterminology editor

update
Version 2.1 edit parameters

Multilingual Ontoterminology Editor © C. Roche user Tedi edit www.ontoterminology.com

Onto Terminology Base ? OTB folder d:\Documents\NextCloud\Maria-Christophe\Tedi\OTB
 new OTB load OTB save OTB OTB file Seat.otb edit OTB

Ontoterminology List ?
 Ontoterminology of seats

author Christophe
 creation date 29 juin 2017 last update 25 mai 2020
 domain Furniture
 sub domain Seat
 other domains
 default language en begin period 0
 number of concepts 24 end period 0 reset
 number of terms 25 update metadata trace
 comment
 Ontoterminology of seats considered as things made or used for sitting on, such as a chair or stool.

nbr of cpts 24 nbr of terms 25
 nbr of objs 9 nbr of PNs 1

new rename delete
 upgrade analyze edit ontoterm

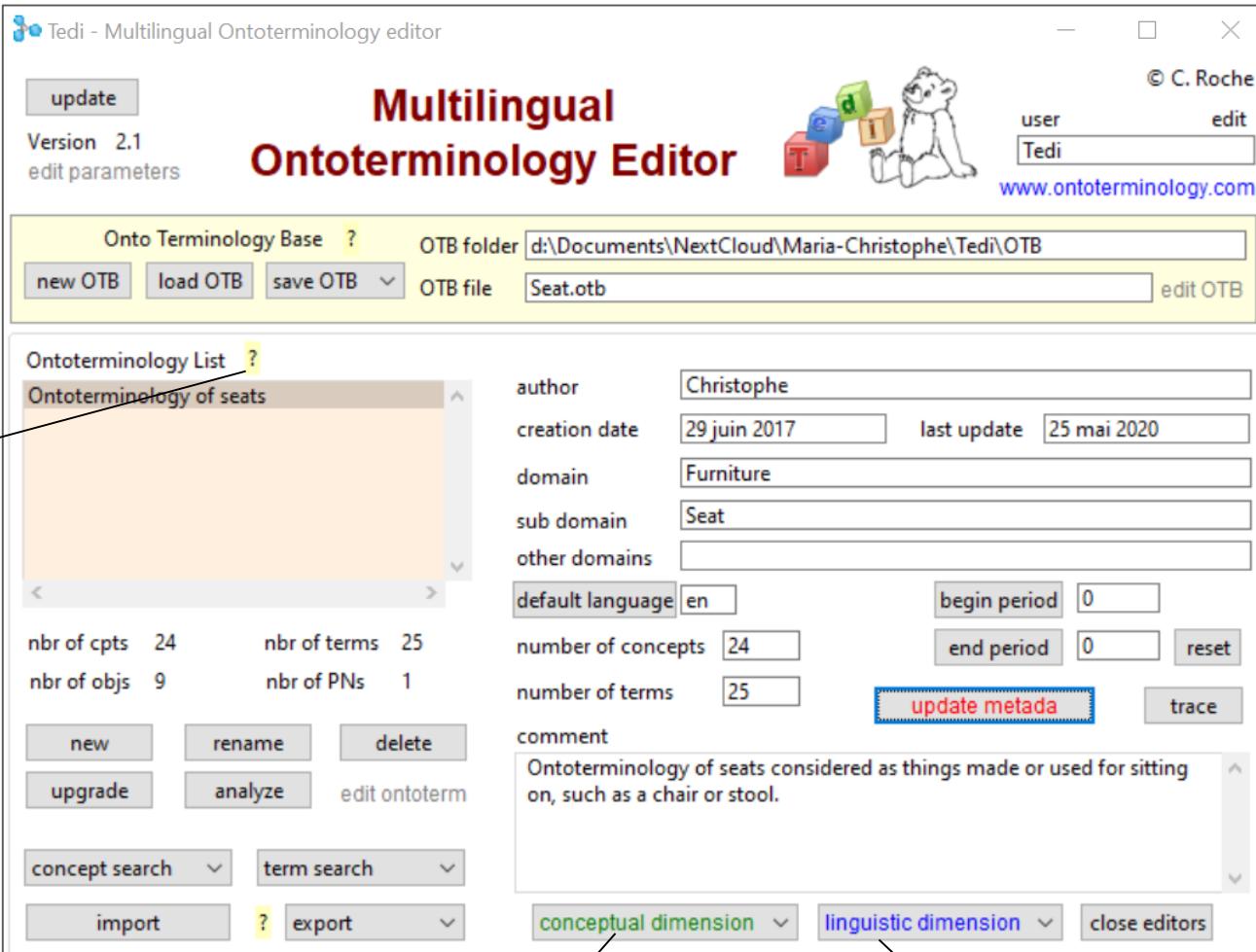
concept search term search
 import ? export conceptual dimension linguistic dimension close editors

Help functions

Access to the editors of the conceptual dimension

Access to the editors of the linguistic dimension

112

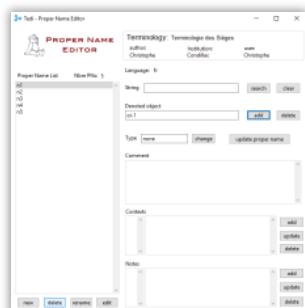


(2) Environment

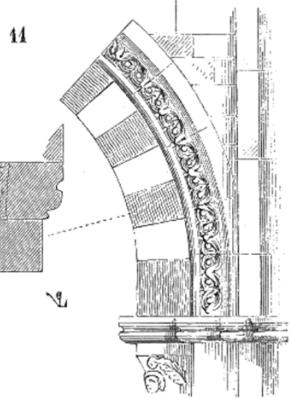
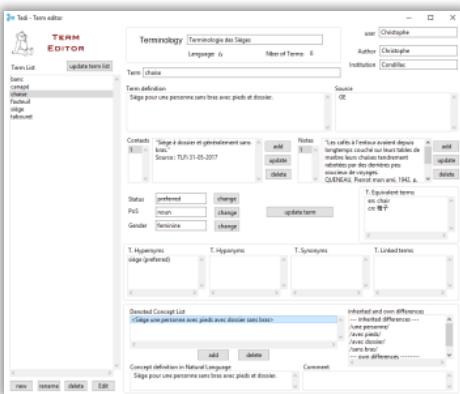
- 1) Theoretical Foundations
- 2) Environment
- 3) Methodology
- 4) Export

- A set of dedicated editors accessible from the Tedi Launcher

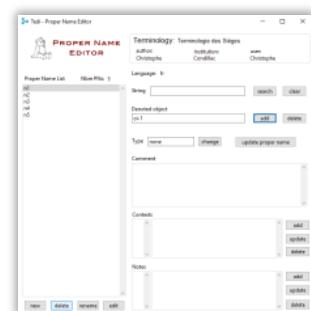
Proper names editor



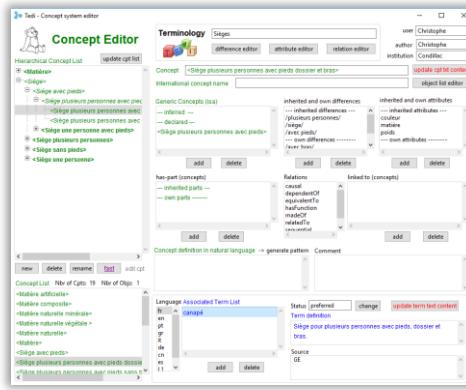
Term editor



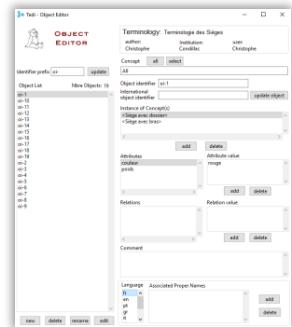
Proper names editor



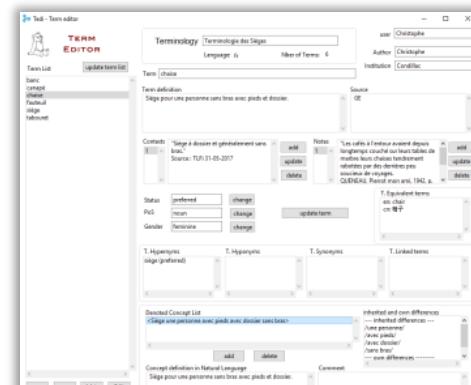
Concept editor



Object editor

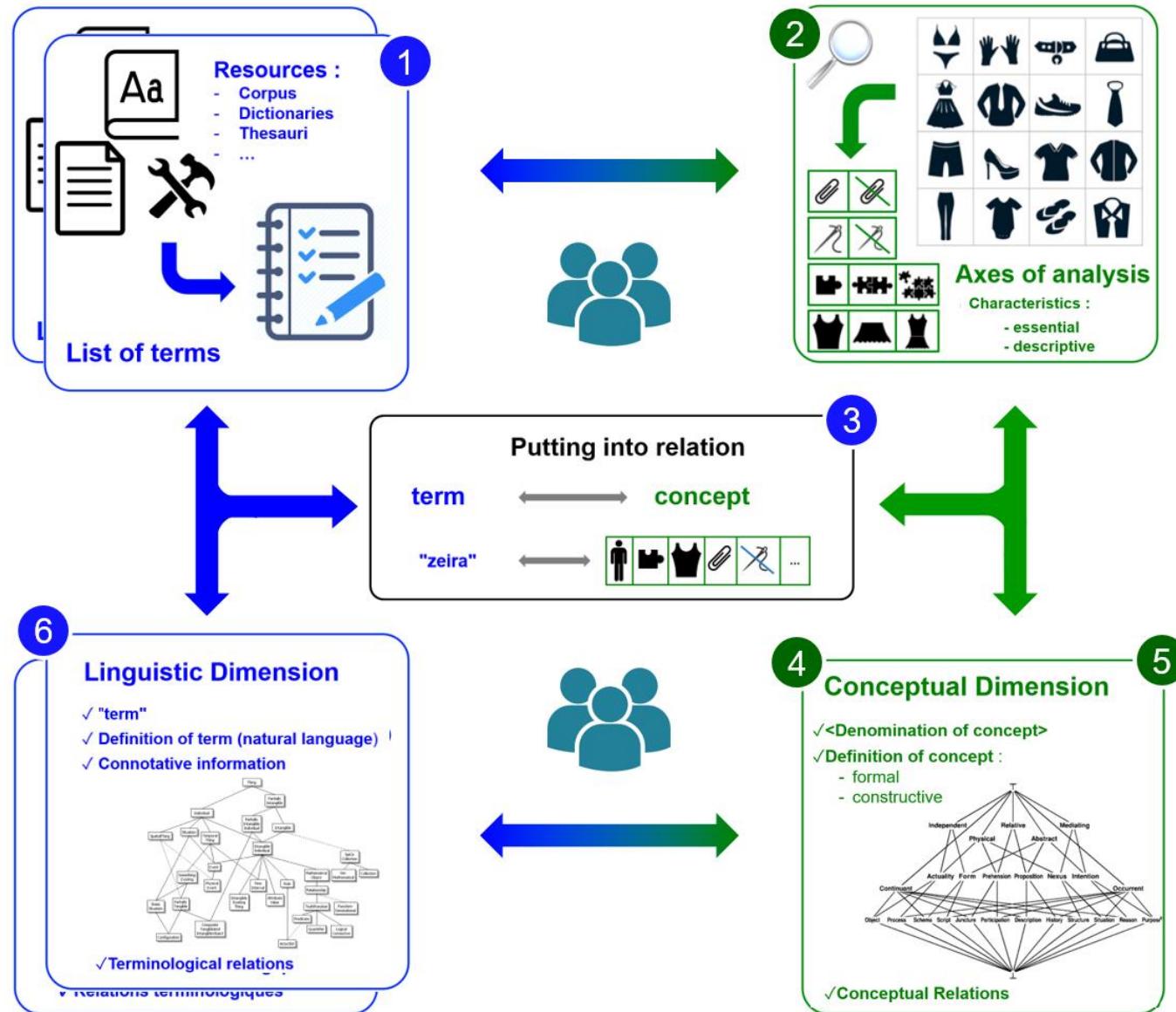


Term Editor



(3) A Term-Guided Methodology

- 1) Theoretical Foundations
- 2) Environment
- 3) Methodology
- 4) Export

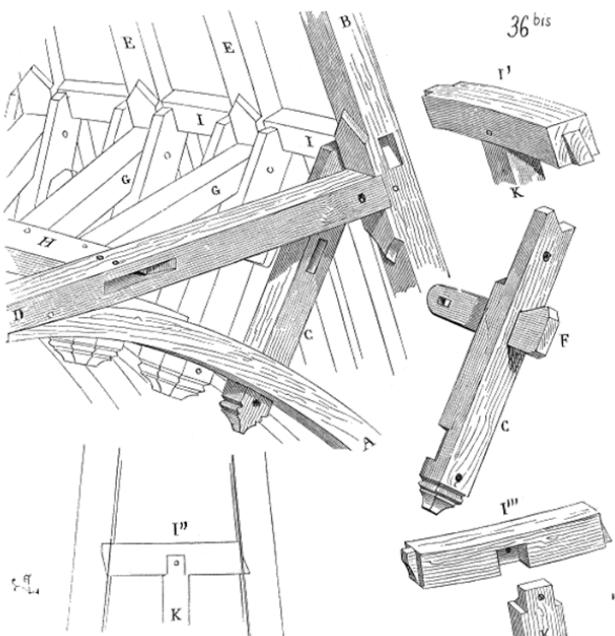


- Experts are guided by Tedi which proposes only valid information at each step



1: Enter the terms to be defined

2: Identify the axes of analysis and their essential characteristics



3: Select the set of characteristics denoted by the term

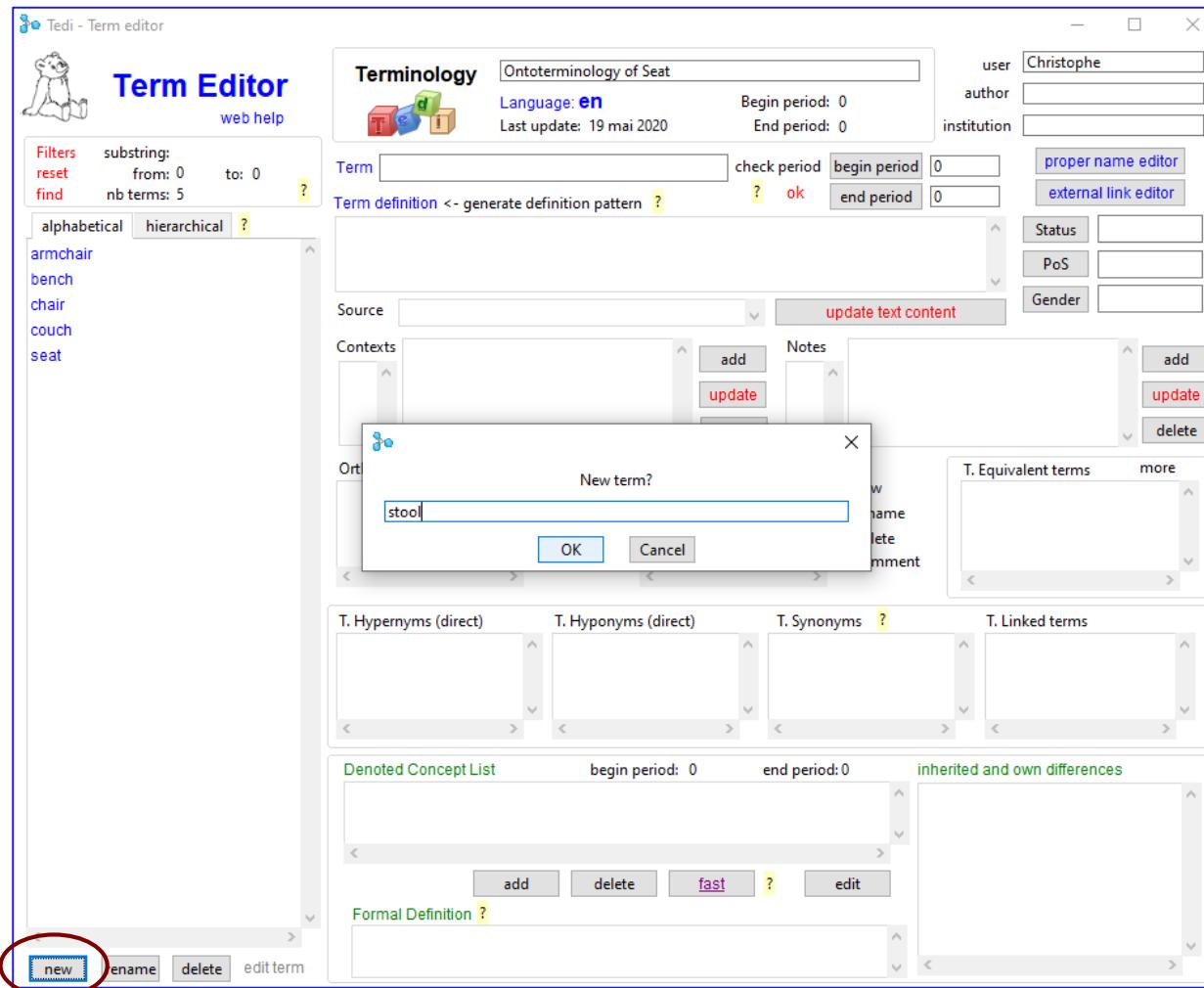
4: Create the concept if it does not exist

5: Update the concept system if necessary

6: Complete the linguistic dimension

1

Enter the terms to be defined



2

Identify the Axes of Analysis and their essential characteristics



Looking for differences between objects

/without arms/



/with arms/



Tedi - Concept system editor

Concept Editor

Terminology Ontoterminalogy of Seat
begin period: 0 end period: 0 **axis of analysis editor** attribute editor relation editor

Nbr of Cpts: 0 Nbr of Obs: 0 web help

hierarchical alphabetical labeled ?

international concept name

Comment

Tedi - Axis of analysis editor

Axis of Analysis Editor

Axis of analysis List ?

Arm
Back
Feet
Number of persons

International axis name
Arm
Begin period: 0 End period: 0 ?
Comment

Terminology begin period: 0 end period: 0
Ontoterminalogy of Seat

Difference (essential characteristic) List ?

/with arms/
/without arms/

International difference name /without arms/
Comment

new delete rename edit axis update comment

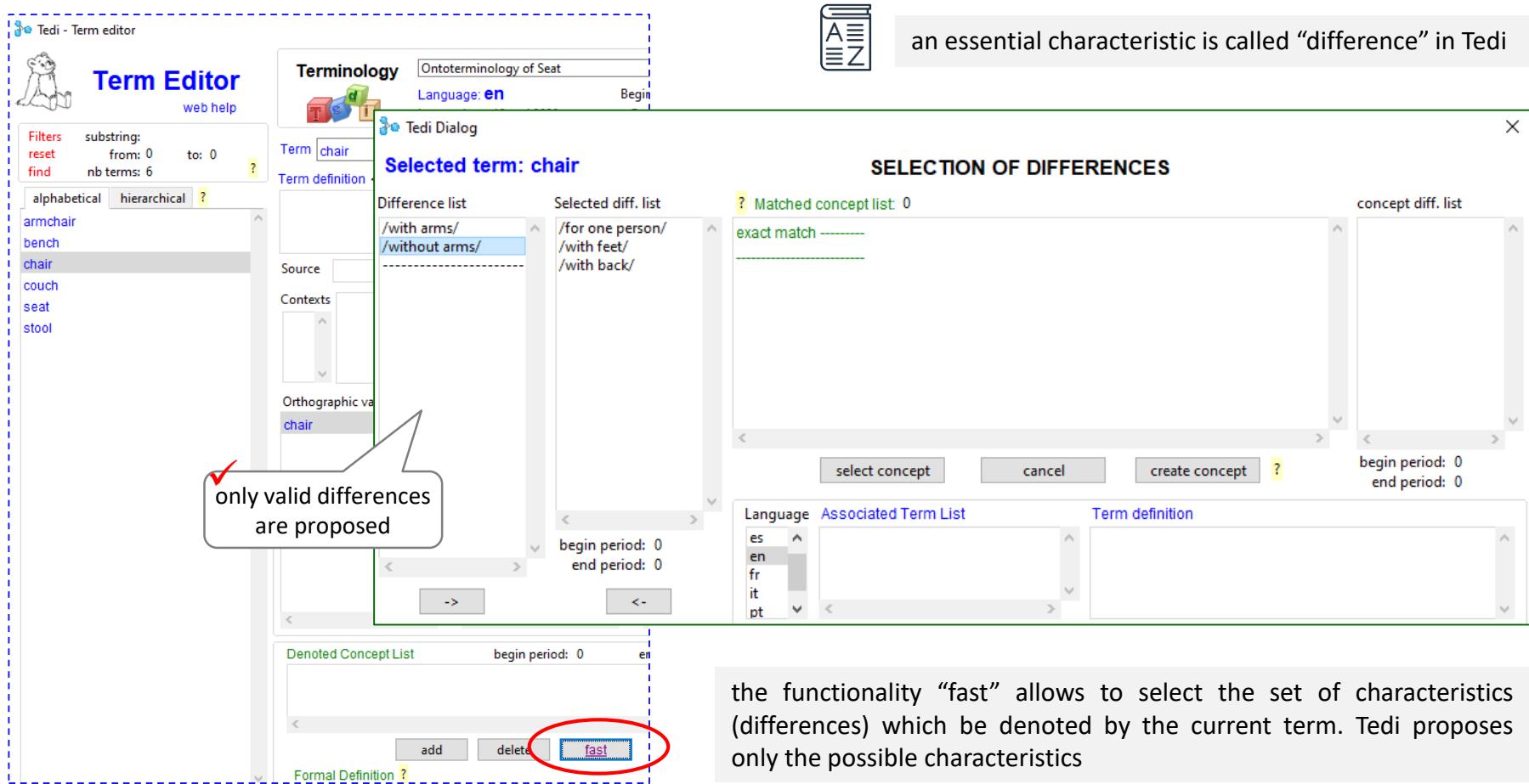
new delete rename edit diff update comment

(3) A Term-Guided Methodology

- 1) Theoretical Foundations
- 2) Environment
- 3) Methodology
- 4) Export

3 Select the set of characteristics denoted by a term

 an essential characteristic is called “difference” in Tedi



only valid differences are proposed

the functionality “fast” allows to select the set of characteristics (differences) which be denoted by the current term. Tedi proposes only the possible characteristics

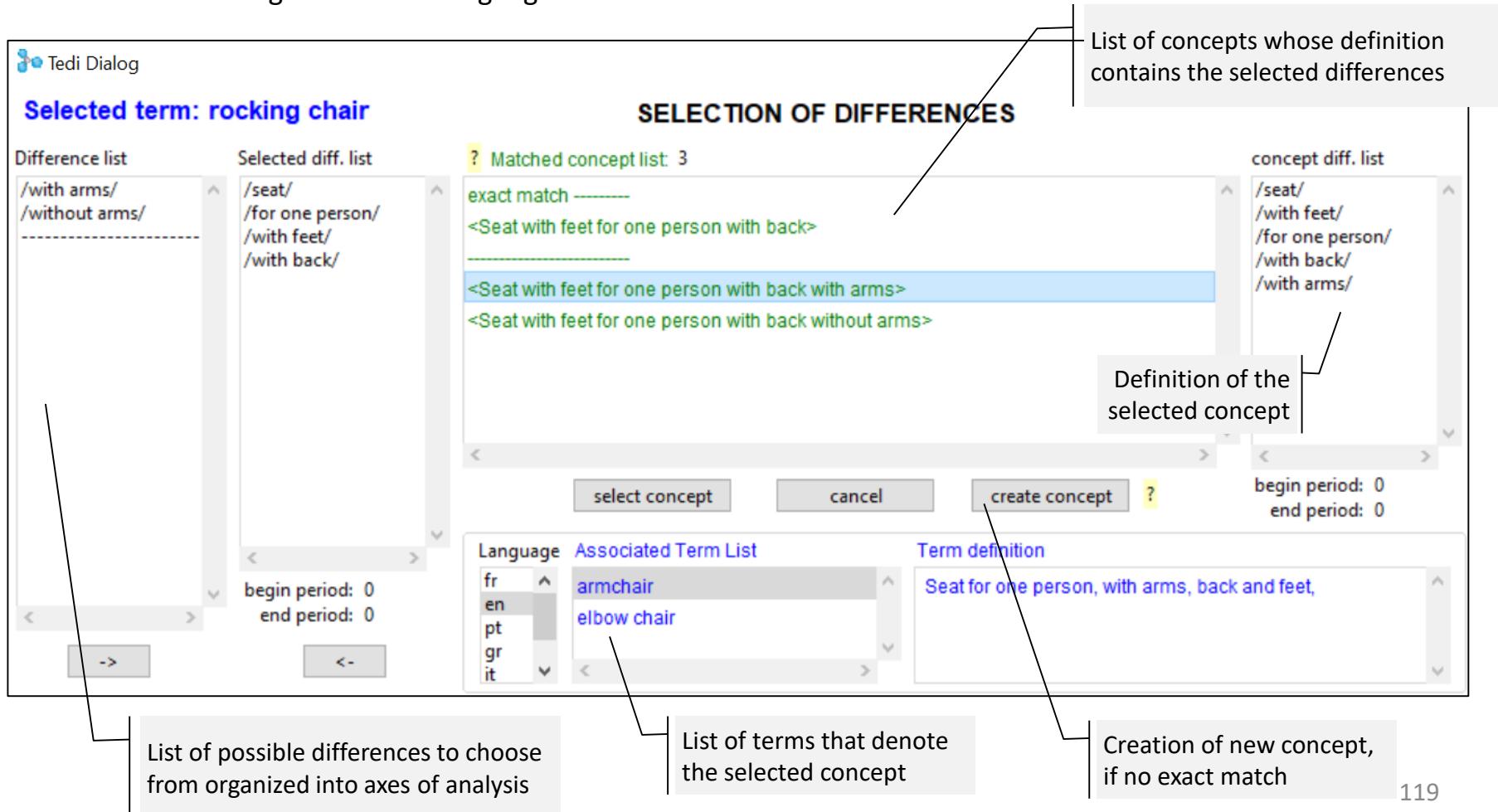
■ Connecting Terms and Concepts

- Terms are known to experts
- A term denotes a set of essential characteristics stable enough to have a name in a given natural language

Tedi Dialog

Selected term: rocking chair

SELECTION OF DIFFERENCES



The screenshot shows the Tedi Dialog software interface with the following components and annotations:

- Difference list:** A list of differences organized into axes of analysis, such as "/with arms/" and "/without arms/".
- Selected diff. list:** A list of selected differences, such as "/seat/", "/for one person/", "/with feet/", and "/with back/".
- Matched concept list:** A list of matched concepts, currently showing 3 exact matches: "<Seat with feet for one person with back>", "<Seat with feet for one person with back with arms>", and "<Seat with feet for one person with back without arms>".
- concept diff. list:** A list of concepts whose definition contains the selected differences, such as "/seat/", "/with feet/", "/for one person/", "/with back/", and "/with arms/".
- Language:** A dropdown menu showing "fr", "en", "pt", "gr", and "it".
- Associated Term List:** A list of terms associated with the selected concept, such as "armchair" and "elbow chair".
- Term definition:** A text field containing the definition of the selected concept: "Seat for one person, with arms, back and feet,".
- Buttons:** "select concept", "cancel", "create concept", and a question mark icon.
- Time Periods:** "begin period: 0" and "end period: 0" for both the difference and concept lists.

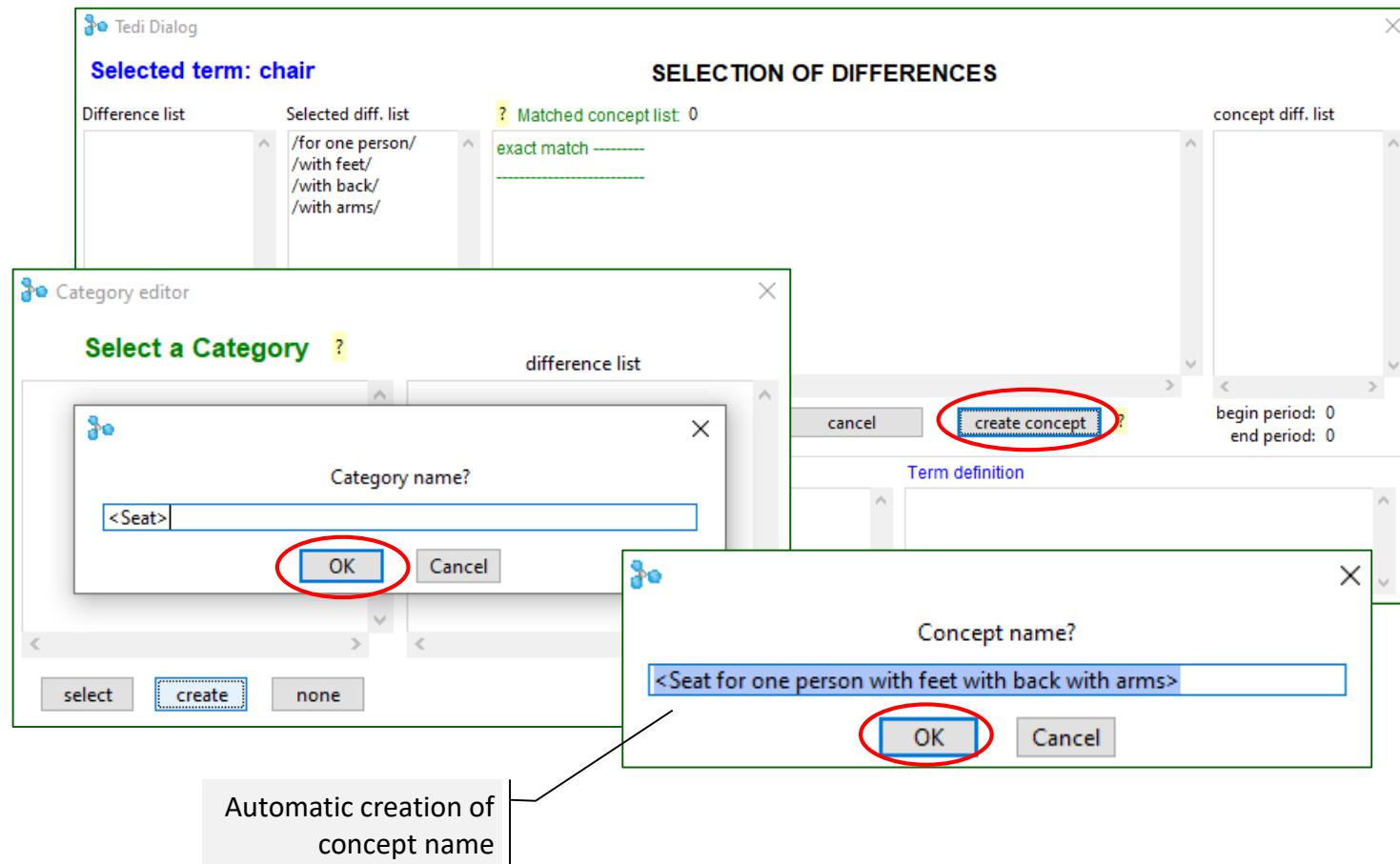
Annotations provide additional context:

- "List of possible differences to choose from organized into axes of analysis" points to the Difference list.
- "List of terms that denote the selected concept" points to the Associated Term List.
- "Creation of new concept, if no exact match" points to the "create concept" button.
- "Definition of the selected concept" points to the Term definition field.
- "List of concepts whose definition contains the selected differences" points to the concept diff. list.

4

Create the concept if it does not exist

Each concept belongs to a category. Create it if it does not exist.



5 Update the concept system if necessary

Tedi - Concept system editor

Concept Editor

Nbr of Cpts: 2 Nbr of Obs: 0 web help

hierarchical alphabetical labeled ?

- <Seat>
- <Seat for one person with feet with back v

Terminology Ontoterminology of Seat

 begin period: 0 axis of analysis editor
end period: 0 relation edit

Concept ? <Seat for one person with feet with back without arms>

international concept name <Seat for one person with feet with back without arms>

Comment update

Generic Concepts (isa) ?

- inferred ---
- declared ---
- <Seat>

inherited and own differences

- inherited differences ---
- own differences -----
- /for one person/
- /with feet/
- /with back/
- /without arms/

Dedicated Editors

Tedi proposes only the valid options (checking « on the fly »)

Tedi - Concept system editor

Concept Editor

The screenshot shows the Tedi Concept Editor interface. On the left, there is a tree view of concepts under the <Seat> category. The selected node is <Seat with feet for one person with back>. The main area contains several editors:

- Terminology:** Shows Ontoterminology of seats, begin period: 0, end period: 0, axis of analysis editor, attribute editor, and relation editor.
- Concept:** Shows the concept <Seat with feet for one person with back without arms>.
- international concept name:** Shows the same concept name.
- Comment:** Shows update.
- Generic Concepts (isa):** Shows inferred and declared concepts, and the selected concept <Seat with feet for one person with back>.
- inherited and own differences:** Shows inherited differences (seat, with feet, for one person, with back) and own differences (without arms).
- inherited and own attributes:** Shows inherited attributes (weight, material, colour) and own attributes (---).
- has-part (concepts):** Shows inherited parts (Feet, Back) and own parts (---).
- Relations:** Shows causal, dependentOf, equivalentTo, hasFunction, madeOf relations.
- linked to (concepts):** Shows linked concepts.
- Formal Definition:** Shows the formal definition <Seat with feet for one person with back> + /without arms/.
- Language Associated Term List:** Shows terms for French (fr), English (en), Portuguese (pt), German (gr), Italian (it), Dutch (de), and Chinese (cn). The term 'chair' is listed under French.
- Status:** Shows preferred status, change button, and update button.
- Term definition:** Shows the term definition: Seat for one person, with back and legs, without arms.
- Source:** Shows the source of the term definition.

Default list of relations

6 Complete the linguistic dimension as necessary

The screenshot shows the Tedi - Term editor software interface. The main window title is "Term Editor". On the left, there is a sidebar with a cartoon character icon, a search bar with filters (substring, reset, find), and a list of terms: armchair, bench, chair (selected), couch, seat, stool. Below the sidebar are tabs for "alphabetical" and "hierarchical". The right side of the interface displays the "Terminology" section. It includes the project name "Ontoterminology of Seat", language "en", and last update date "19 mai 2020". A red circle highlights the "Term definition" field, which contains the term "chair" and a link to generate a definition pattern. Below this, the definition is shown: "Seat for one person, with feet, with back, without arms.". The "Contexts" section lists a single context entry with the URL <https://www.collinsdictionary.com/dictionary/english/chair>. At the bottom, there are sections for "Orthographic variations" and "Inflected forms".

Automatic generation of patterns of definition in natural language

The screenshot shows the Tedi Term Editor interface. At the top, there's a logo of a bear, the title 'Term Editor', and a 'web help' link. On the left, a sidebar lists terms: armchair, bench, chair, couch, elbow chair, ottoman, seat, and stool. Below this is a 'Filters' section with 'substring' search options. The main area is titled 'Terminology' with 'Ontoterminology of seats'. It shows the term 'chair' with a definition: 'Seat for one person, with back and legs, without arms.' There are buttons for 'check period', 'begin period' (set to 0), and 'end period' (set to 0). To the right, there's a sidebar with user info (user: Tedi, author: Christophe, institution: Condillac) and links for 'proper name editor' and 'external link editor'. Below these are status indicators ('Status: preferred', 'PoS: noun', 'Gender: neuter') and a note about the term 'chair'.

Automatic calculation of terminological hypernyms, hyponyms, synonyms for every term

Automatic calculation of terminological equivalents in other languages (multilingualism)

Automatic generation of patterns of definition in natural language

Inherited and own differences

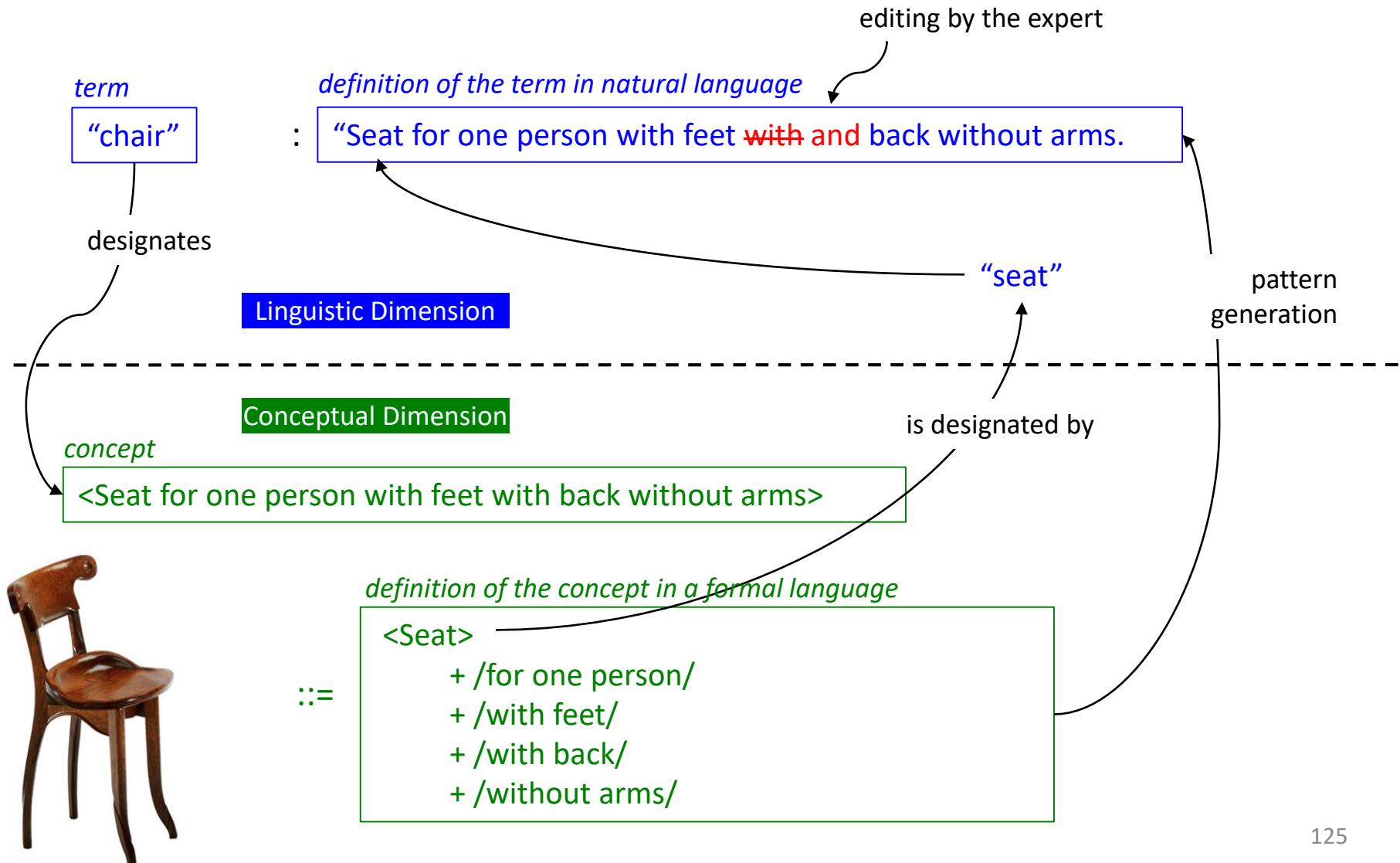
- inherited differences ---
- /seat/
- /with feet/
- /for one person/
- /with back/
- own differences -----
- /without arms/

Annotations with arrows point from the 'T. Hypernyms (direct)', 'T. Hyponyms (direct)', 'T. Synonyms', and 'T. Linked terms' sections to the descriptive text blocks above them.

(3) A Term-Guided Methodology

- 1) Theoretical Foundations
- 2) Environment
- 3) Methodology
- 4) Export

Generation of definition pattern in natural language



Tedi Onto-Dictionary on "Ontoterminology of seats" (en)

Date: 25 mai 2020 - Time: 13:01:27 - Version: 2.1 - www.ontoterminology.com/tedi

search:

armchair

bench

chair

couch

elbow chair

ottoman

seat

stool

chair

Definition: Seat for one person, with back and legs, without arms.

Status: preferred

Context(s):

1) 'Cafe tables and chairs face a bench seat below a lowered ceiling that curves for an atmospheric effect.'

<https://www.lexico.com/en/definition/chair>

2020 05 25

Note(s):

1) "A chair is a piece of furniture for one person to sit on. Chairs have a back and four legs."

<https://www.collinsdictionary.com/dictionary/english/chair>

2020 05 17

Equivalent(s):

- fr: chaise (preferred)

- fr: chaise d'intérieur (alternative)

- gr: καρέκλα (preferred)

Concept: <Seat with feet for one person with back without arms>

essential characteristic(s): /seat/, /with feet/, /for one person/, /with back/, /without arms/,

a kind of: <Seat with feet for one person with back>,

linked to: <Back>, <Feet>,

rdfs:seeAlso: <https://fr.wikipedia.org/wiki/Chaise>

skos:exactMatch: <http://vocab.getty.edu/page/aat/300037772>

Web reference: [Le grand salon de l'Impératrice au château de Fontainebleau du temps de l'impératrice Eugénie](#)

Illustration: Chaises estampillées Georges Jacob, grand cabinet du Dauphin, château de Versailles.

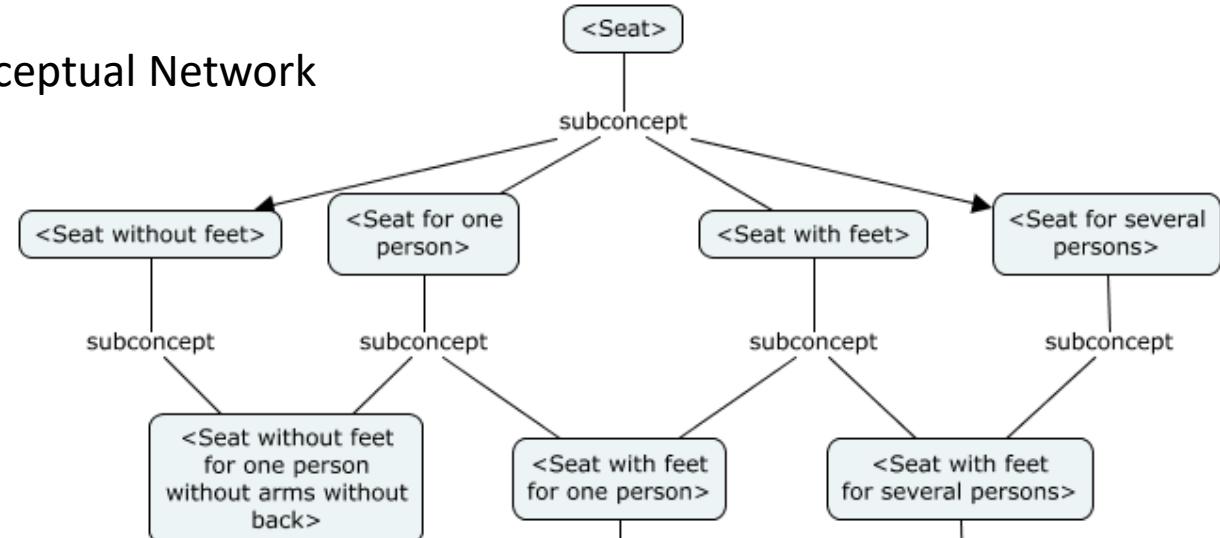


Objects of this type: 2

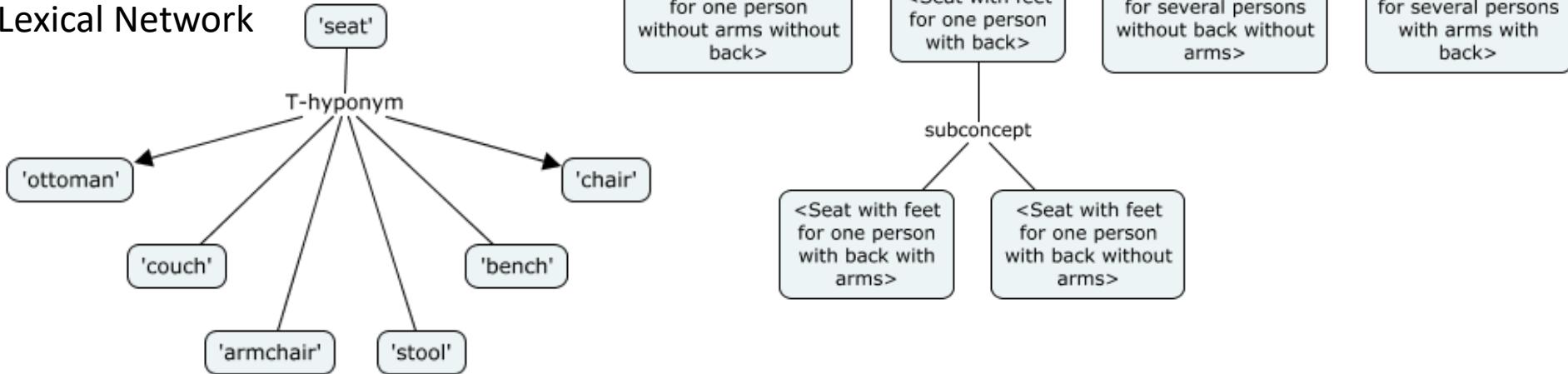
Conceptual Network



CmapTools



Lexical Network



(4) Export: RDF

- 1) Theoretical Foundations
- 2) Environment
- 3) Methodology
- 4) Export

```

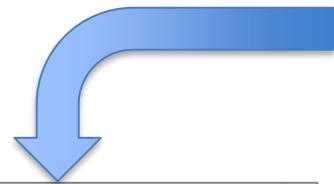
<?xml version="1.0" encoding="utf-8"?>
<!-- Ontoterminology: Ontoterminology of seats -->
<!-- Author: Christophe -->
<!-- Creation date of ontoterminology: 29 juin 2017 -->
<!-- Export date: 25 mai 2020 time: 13:02:51 -->
<!-- Generated by Tedi version: 2.1 - http://christophe-roche.fr/tedi -->

<rdf:RDF xmlns="http://www.ontologia.fr/OTB/Seat#"
    xmlns:base="http://www.ontologia.fr/OTB/Seat"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:skos="http://www.w3.org/2004/02/skos/core#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:vs="http://www.w3.org/2003/06/sw-vocab-status/ns#"
    xmlns:vann="http://purl.org/vocab/vann/">
<owl:Ontology rdf:about="http://www.ontologia.fr/OTB/Seat.rdf">
    <dc:title>Ontoterminology of seats</dc:title>
    <dc:description>Ontoterminology of seats considered as things made or used for sitting on, such as a chair or stool.</dc:description>
    <dc:issued rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2017-6-29</dc:issue
    <dc:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2020-5-25</dc:mod
    <dc:creator>Christophe Roche</dc:creator>
    <dc:publisher>Condillac</dc:publisher>
</owl:Ontology>

<!-- Object Properties: -->

<owl:ObjectProperty rdf:about="#hasPart">
    <owl:inverseOf rdf:resource="#partOf"/>
    <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:range rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:ObjectProperty>

```



=====
Core Vocabularies for RDF Export
=====

Concepts:

- owl:Class
- owl:NamedIndividual
- owl:Ontology
- owl:DatatypeProperty
- owl:ObjectProperty

Object Properties:

- foaf:depiction
- rdf:type
- rdfs:label
- rdfs:subClassOf
- skos:broader
- skos:closeMatch
- skos:exactMatch
- skos:note
- skos:related
- skos:scopeNote
- owl:sameAs
- rdfs:comment
- rdfs:seeAlso
- skos:altLabel
- skos:broadMatch
- skos:definition
- skos:narrowMatch
- skos:prefLabel
- skos:relatedMatch

Concept

```

<owl:Class rdf:about="#Seat_with_feet_for_one_person_with_back_without_arms">
  <skos:prefLabel xml:lang="fr">chaise</skos:prefLabel>
  <skos:definition xml:lang="fr">Siège avec dossier, pour une personne, avec pieds, sans bras.</skos:definition>
  <skos:note xml:lang="fr">Siège à dossier et généralement sans bras. Source : TLFi 31052017</skos:note>
  <skos:example xml:lang="fr">"Les cafés à l'entour avaient depuis longtemps couché sur leurs tables de marbre leurs chaises tendrement rabotées par des derrières peu soucieux de voyages. QUENEAU, Pierrot mon ami, 1942, p. 199." Source : TLFi 31052017</skos:example>
  <skos:altLabel xml:lang="fr">chaise d'intérieur</skos:altLabel>
  <skos:definition xml:lang="fr">Siège avec dossier, pour une personne, avec pieds, sans bras.</skos:definition>
  <skos:prefLabel xml:lang="en">chair</skos:prefLabel>
  <skos:definition xml:lang="en">Seat for one person, with back and legs, without arms.</skos:definition>
  <skos:note xml:lang="en">"A chair is a piece of furniture for one person to sit on. Chairs have a back and four legs." https://www.collinsdictionary.com/dictionary/english/chair
  2020 05 17</skos:note>
  <skos:example xml:lang="en">'Cafe tables and chairs face a bench seat below a lowered ceiling that curves for an atmospheric effect.'
```

<https://www.lexico.com/en/definition/chair>

```

  2020 05 25</skos:example>
  <skos:prefLabel xml:lang="gr">καρέκλα</skos:prefLabel>
  <skos:definition xml:lang="gr">Κάθισμα για ένα άτομο, χωρίς μπράτσα, με πλάτη και πόδια. </skos:definition>
  <skos:broader rdf:resource="#Seat_with_feet_for_one_person_with_back"/>
  <rdfs:subClassOf rdf:resource="#Seat_with_feet_for_one_person_with_back"/>
  <rdfs:subClassOf rdf:resource="#without_arms"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasPart"/>
      <owl:someValuesFrom rdf:resource="#Back"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <foaf:depiction rdf:resource="https://upload.wikimedia.org/wikipedia/commons/thumb/e/e0/Ch%C3%A2teau_de_Versailles%2C_appartement_du_Dauphin%2C_grand_cabinet_du_Dauphin%2C_chaises.jpg/800px-Ch%C3%A2teau_de_Versailles%2C_appartement_du_Dauphin%2C_grand_cabinet_du_Dauphin%2C_chaises.jpg"/>
  <skos:exactMatch rdf:resource="http://vocab.getty.edu/page/aat/300037772"/>
  <rdfs:seeAlso rdf:resource="https://fr.wikipedia.org/wiki/Chaise"/>
</owl:Class>
```



Individual

```
<owl:NamedIndividual rdf:about="#Chaise_T_507_C">
  <skos:prefLabel>Chaise T 507 C</skos:prefLabel>
  <rdf:type rdf:resource="#Seat_with_feet_for_one_person_with_back_without_arms"/>
  <foaf:depiction rdf:resource="https://journals.openedition.org/crcv/docannexe/
    image/13475/img-13-small1480.jpg"/>
  <rdfs:seeAlso rdf:resource="https://journals.openedition.org/crcv/13475?lang=en"/>
</owl:NamedIndividual>
```





(5) Ontoterminology and W3C

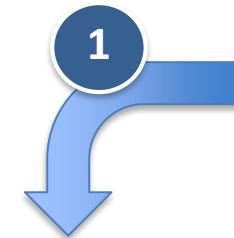
<http://ontologia.fr/OTB/seat.rdf>

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Ontoterminology: Ontoterminology of seats -->
<!-- Author: Christophe -->
<!-- Creation date of ontoterminology: 29 juin 2017 -->
<!-- Export date: 25 mai 2020 time: 13:02:51 -->
<!-- Generated by Tedi version: 2.1 - http://christophe-roche.fr/tedi -->

<rdf:RDF xmlns="http://www.ontologia.fr/OTB/Seat#"
  xml:base="http://www.ontologia.fr/OTB/Seat"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:vs="http://www.w3.org/2003/06/sw-vocab-status/ns#"
  xmlns:vann="http://purl.org/vocab/vann/">
<owl:Ontology rdf:about="http://www.ontologia.fr/OTB/Seat.rdf">
  <dc:title>Ontoterminology of seats</dc:title>
  <dc:description>Ontoterminology of seats considered as things made or used for sitting on, such as a chair or stool.</dc:description>
  <dc:issued rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2017-6-29</dc:issued>
  <dc:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2020-5-25</dc:modified>
  <dc:creator>Christophe Roche</dc:creator>
  <dc:publisher>Condillac</dc:publisher>
</owl:Ontology>

<!-- Object Properties: -->

<owl:ObjectProperty rdf:about="#hasPart">
  <owl:inverseOf rdf:resource="#partOf"/>
  <rdfs:domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:range rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:ObjectProperty>
```



RDF Export



Protégé



Querying in SPARQL

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
```

```
SELECT ?name ?definition
```

① FROM <<http://ontologia.fr/OTB/krater.rdf>>

```
WHERE {
```

```
  ?concept rdf:type owl:Class.
  ?concept skos:prefLabel ?name.
  ?concept skos:definition ?definition.
  FILTER (lang(?name) = 'en')
  FILTER (lang(?definition) = 'en')
```

```
}
```

```
ORDER BY ?name
```

```
LIMIT 100
```

① or ②

<https://data.bnf.fr/current/sparql.html>

Editeur SPARQL de data.bnf.fr

Graphe par défaut (IRI)

② <http://ontologia.fr/OTB/seat.rdf>

Requête

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
3
4 SELECT ?name ?definition
5 FROM <http://ontologia.fr/OTB/krater.rdf>
6
7 * WHERE {
8   ?concept rdf:type owl:Class.
9   ?concept skos:prefLabel ?name.
10  ?concept skos:definition ?definition.
11  FILTER (lang(?name) = 'en')
12  FILTER (lang(?definition) = 'en')
13  }
14 |
15 ORDER BY ?name
16 LIMIT 100
```

②

Sponging:

Retrieve remote RDF data for all missing source graphs

Querying in SPARQL



name	definition
"armchair"@en	"Seat for one person, with arms, back and feet, "@en
"bench"@en	"Seat without back or arms, for more than one persons, with feet, "@en
"chair"@en	"Seat for one person, with back and legs, without arms."@en
"couch"@en	"Seat formore than one people, with arms, back and feet. "@en
"ottoman"@en	"Seat without back or armrests, for one person, without feet, which usually serves as foot rest or box, with the seat part hinged onto the lower part."@en
"seat"@en	"Piece of furniture designed for seating on."@en
"stool"@en	"Seat for one person, with feet, without arm and back. "@en

Querying in SPARQL



PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX skos: <http://www.w3.org/2004/02/skos/core#>

PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?label ?type ?img

WHERE {

?object rdf:type owl:NamedIndividual.
?object skos:prefLabel ?label.
?object rdf:type ?concept.
?concept skos:prefLabel ?type.
?object foaf:depiction ?img.

FILTER (lang(?type)='en')

}

ORDER BY ?label

LIMIT 100

<https://data.bnfr/current/sparql.html>

Graphe par défaut (IRI)

http://ontologia.fr/OTB/seat.rdf

Requête

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
3 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
4
5 SELECT ?label ?type ?img
6
7 WHERE {
8     ?object rdf:type owl:NamedIndividual.
9     ?object skos:prefLabel ?label.
10    ?object rdf:type ?concept.
11    ?concept skos:prefLabel ?type.
12    ?object foaf:depiction ?img.
13    FILTER (lang(?type)='en')
14
15 ORDER BY ?label
16 LIMIT 100
```

Sponging:

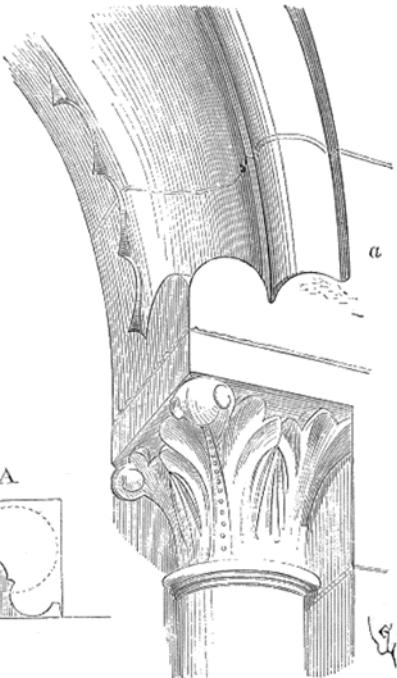
Retrieve remote RDF data for all missing source graph

Querying in SPARQL



label	type	img
"Canapé GMT 21269"	"couch"@en	https://journals.openedition.org/crcv/docannexe/image/13475/img-3-small480.jpg
"Canapé T 504 C"	"couch"@en	https://journals.openedition.org/crcv/docannexe/image/13475/img-10-small480.jpg
"Chaise F 263"	"chair"@en	https://journals.openedition.org/crcv/docannexe/image/13475/img-6-small580.jpg
"Chaise T 507 C"	"chair"@en	https://journals.openedition.org/crcv/docannexe/image/13475/img-13-small480.jpg
"Fauteuil F 914 C"	"armchair"@en	https://journals.openedition.org/crcv/docannexe/image/13475/img-5.jpg
"Fauteuil T 505 C"	"armchair"@en	https://journals.openedition.org/crcv/docannexe/image/13475/img-11.jpg
"Fauteuil T 506 C"	"armchair"@en	https://journals.openedition.org/crcv/docannexe/image/13475/img-12.jpg
"Tabouret GMT 1414/6"	"stool"@en	https://journals.openedition.org/crcv/docannexe/image/13475/img-9-small480.jpg
"Tabouret S 574"	"stool"@en	https://journals.openedition.org/crcv/docannexe/image/13475/img-7-small580.jpg

2



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 skos: <http://www.w3.org/2004/02/skos/core#>

SELECT ?name ?definition
FROM <http://ontologia.fr/OTB/krater.rdf>

WHERE {
 ?concept rdf:type owl:Class.
 ?concept skos:prefLabel ?name.
 ?concept skos:definition ?definition.
 FILTER (lang(?name) = 'en')
 FILTER (lang(?definition) = 'en')
}
}

ORDER BY ?name
LIMIT 100

<http://sparql.org/sparql.html>

The screenshot shows a web-based SPARQL query editor titled "SPARQLer - General purpose processor". The main area contains a text input field for a SPARQL query. The query is identical to the one shown in the left panel, including prefixes, a FROM clause, WHERE clause with filters for English labels and definitions, an ORDER BY clause, and a LIMIT clause. Below the query input, there are several controls: a "Target graph URI" input field, a note about executing against an empty dataset, a note about using VALUES variables, and output options for "Output" (set to XML), "XSLT style sheet" (set to /xml-to-html.xsl), and a checkbox for accepting text/plain regardless of the accept header. At the bottom is a "Get Results" button.

```
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 skos: <http://www.w3.org/2004/02/skos/core#>

SELECT ?name ?definition
FROM <http://ontologia.fr/OTB/krater.rdf>

WHERE {
    ?concept rdf:type owl:Class.
    ?concept skos:prefLabel ?name.
    ?concept skos:definition ?definition.
    FILTER (lang(?name) = 'en')
    FILTER (lang(?definition) = 'en')
}

ORDER BY ?name
LIMIT 100
```

Selected References

Mind the Gap: Ontology Authoring for Humanists

C. Roche M. Papadopoulou. 1st International Workshop for Digital Humanities and their Social Analysis (WODHSA)- Episode V: The Styrian Autumn of Ontology, September 23-25, a Workshop hosted by Joint Ontology Workshops, Medical University of Graz (Austria), September 23-25, 2019

Building ontology-based dictionaries for Greek material culture terms

M. Papadopoulou & C. Roche. 1st International Workshop on Open Data and Ontologies for Cultural Heritage. Rome, Italy, June 3, 2019

Twining Classics and A.I.: Building the new generation of ontology-based lexicographical tools and resources for Humanists on the Semantic Web

M. Papadopoulou, C. Roche. In S. Krauwer, D. Fišer (eds.) – Twin Talks at DHN 2018 – Understanding Collaboration in Digital Humanities. Open access, University of Copenhagen, 5th March 2019

Ontologization of Terminology. A worked example from the domain of ancient Greek dress

M. Papadopoulou, C. Roche. AIDAinformazioni Journal, number 1-2/2018, volume XXXVI

Ontological definition – abstract

C.Roche. Handbook of Terminology, Volume 1, John Benjamins Publishing, 2015, pp.128-152

Ontoterminology: How to unify terminology and ontology into a single paradigm

C.Roche. LREC 2012, Eighth international conference on Language Resources and Evaluation, Istanbul (Turkey), 21-27 May 2012, pp. 2626-2630

Any Comments or Remarks?

send an email to: roche@univ-savoie.fr



<http://ontoterminology.com/>