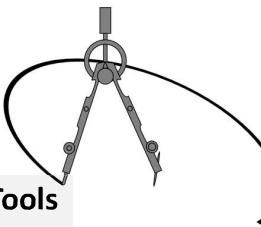


Tools & Environments



- **Graphical Tools:** - CmapTools
- **Formal Tool:** - Protégé
- **Dedicated Tools:** - Tedi



1

CmapTools



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



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.

3

Downloads



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



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

2

CmapTools



Epistemological Principles

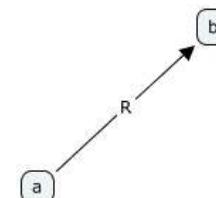
Concept: Concept as a perceived regularity in events or objects, or records of events or objects, designated by a label.

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

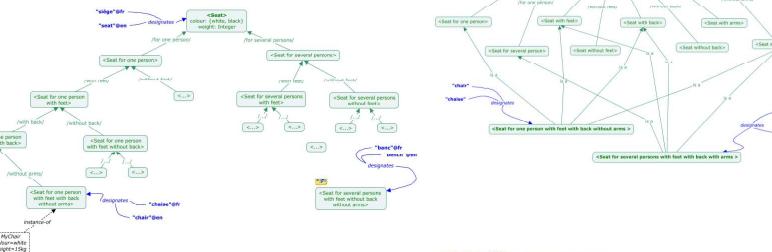
Concept: Node

Proposition: labeled link



4

Graphical Notation



They are all equivalent.

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

<Concept> = { essential characteristics }

13



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.

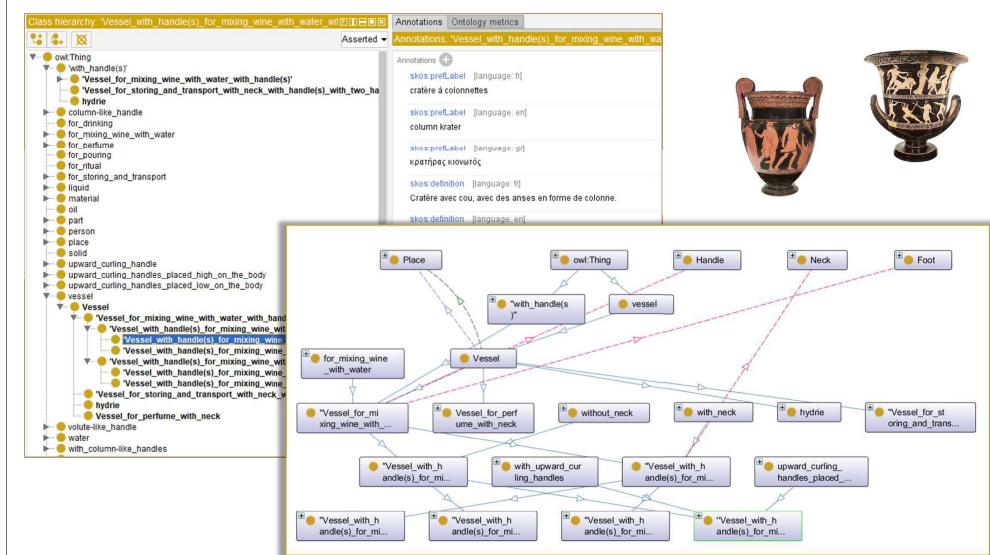


15



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.





<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.5.0

Download platform independent version
(requires a Java Runtime Environment)

17



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

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 Development 101: A Guide to Creating Your First Ontology

Natalya F. Noy and Deborah L. McGuinness
Stanford University, CA 94305
noy@sail.stanford.edu and dm@cs.stanford.edu

1 Why develop an ontology?

In recent years the development of ontologies—explicit formal specifications of the terms in the domain of discourse—has been moving from the realm of Artificial-Intelligence laboratories to the desktops of domain experts. Ontologies have become an increasingly important part of the Semantic Web, from being used for simple taxonomies populating Web sites (such as at Yahoo!) to categorizations of products for e-commerce, to being used for the Semantic Web Service (SWS) layer of the Semantic Web, for developing the Resource Description Framework (RDF) and SPARQL, a language for modeling and querying data on the Semantic Web, and for the Semantic Web itself, for sharing information. The Defense Advanced Research Projects Agency (DARPA) is investigating the use of ontologies for improving decision making in military applications by extending RDF with more expressive constructs aimed at facilitating agent interaction on the Web. The Semantic Web is also being used in many other domains, such as medicine, for example, the Semantic Web in Medicine (SWiM) initiative, which is a joint effort of the National Library of Medicine (NLM), the National Center for Biotechnology Information (NCBI), and the United States Pharmacopeia (USP). In addition, the Semantic Web is being used in other domains, for example, the United Nations Development Program and Dan & Bradstreet created their own ontologies (www.unep.org).

An ontology defines a common vocabulary for researchers and systems to share information in a particular field. It also provides a way for people to know exactly what is meant by the terms used in the ontology.

Why might you want to develop an ontology? Some of the reasons are:

To share common understanding of the structure of information among people or systems
To make reuse of domain knowledge
To make domain assumptions explicit
To analyze domain knowledge
To analyze domain knowledge

sources: <http://www.ontoprise.com/ontology/ontology.html> (a review of approaches among people or software agents); a set of the more common models in developing ontologies (Mosses 1992; Ober 1993). For example, suppose several different Web sites contain similar information or provide similar services. If these sites share a common ontology, then computer agents can extract and aggregate information from all the sites. This allows users to search for information and get information to answer user queries or as input data to other applications. Another example is the Semantic Web, where the Semantic Web is a distributed network of interconnected resources of information, and so on. In our group of researchers, knowledge can be shared in detail; others can simply reuse it for their domains. Additionally, if we need to build large

18



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

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.

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

Matthew Horridge

Contributors

v1.0 Holger Knublauch, Alan Rector, Robert Stevens, Chris Wroe
v1.1 Simon Jupp, Georgina Moultoun, Robert Stevens
v1.2 Nick Drummond, Simon Jupp, Georgina Moultoun, Robert Stevens
v1.3 Sebastian Brandt

THE UNIVERSITY OF MANCHESTER
Copyright © The University Of Manchester

March 21, 2011

19



<http://protegeproject.github.io/protege/>

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.

20

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

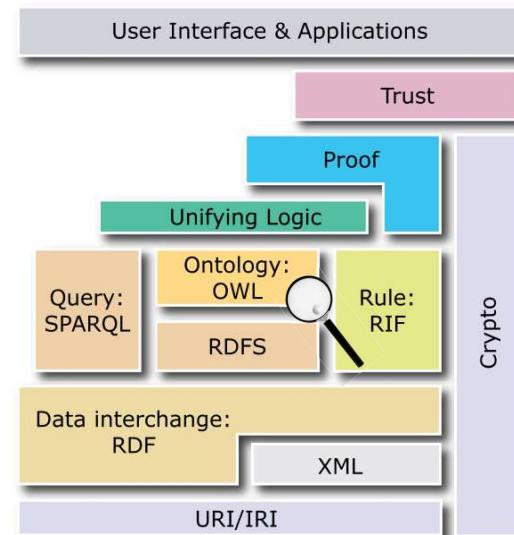
The Open World Assumption, used by OWL, says that "just because we don't know something to be true does not mean that we can assume it to be false"

OWL

OWL provides the theoretical basis for Protege 4 ontologies

21

OWL & the Semantic Web Architecture



22

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

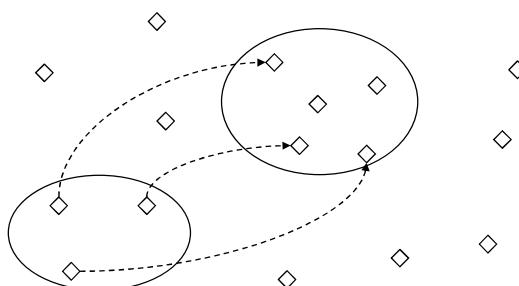
23

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



24

Components of OWL Ontologies: Individuals

1) Individuals

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



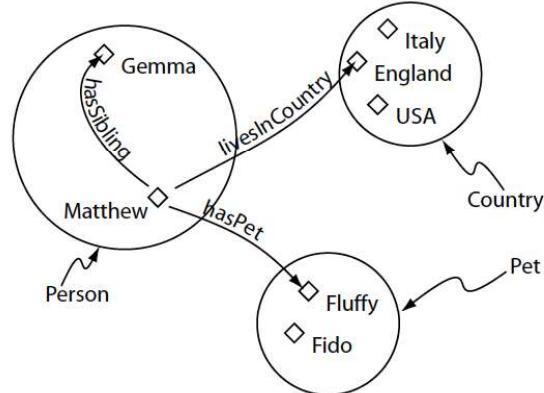
Terminology: « individual », « instance », « object »

25

Components of OWL Ontologies: Classes

3) Classes

OWL classes are interpreted as sets that contain individuals.



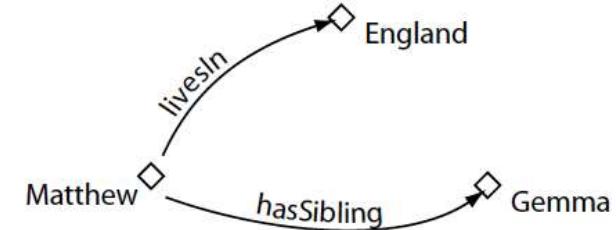
Classes are a concrete representation of concepts.

27

Components of OWL Ontologies: Properties

2) Properties

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

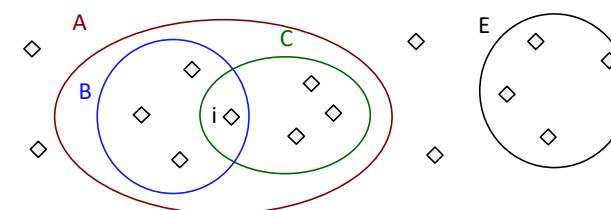


Terminology: « properties », « slots » (Protégé), « roles » (DL), « relations », « attributes »

26

Components of OWL Ontologies: Classes

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

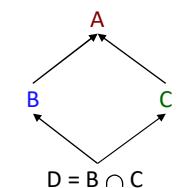


As sets, classes can be defined using set operators : \cup, \cap

$$A = B \cup C$$

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

28

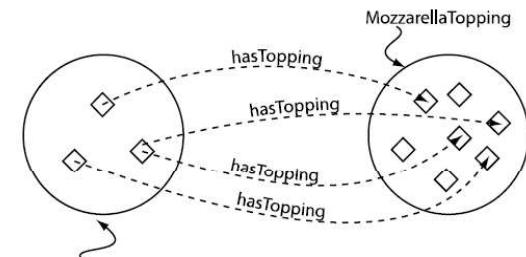
Components of OWL Ontologies: Properties Restriction

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)



29

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



stool

椅子



armchair

chaise



protégé



fauteuil

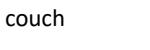
banc

长凳



chair

canapé



couch

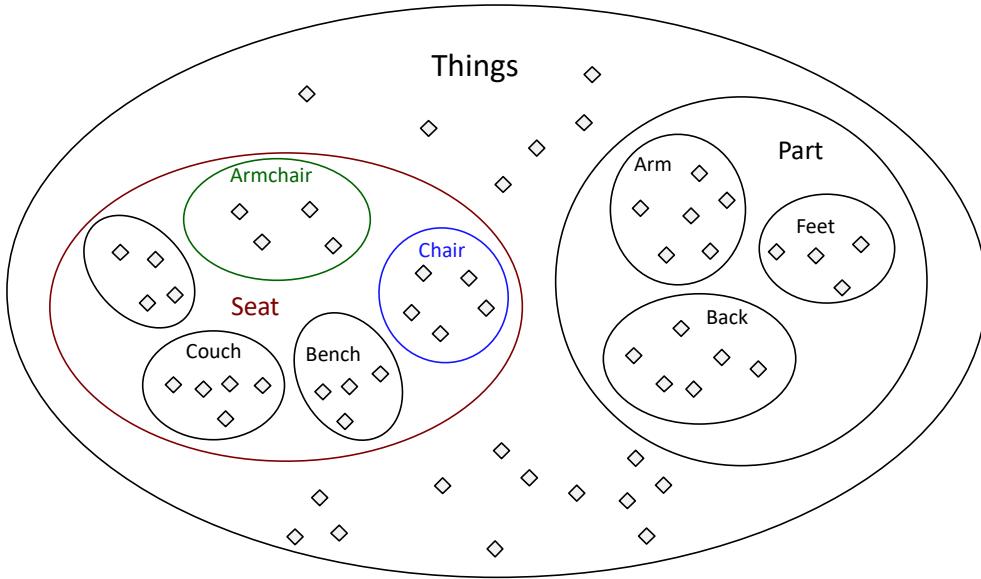


bench

tabouret

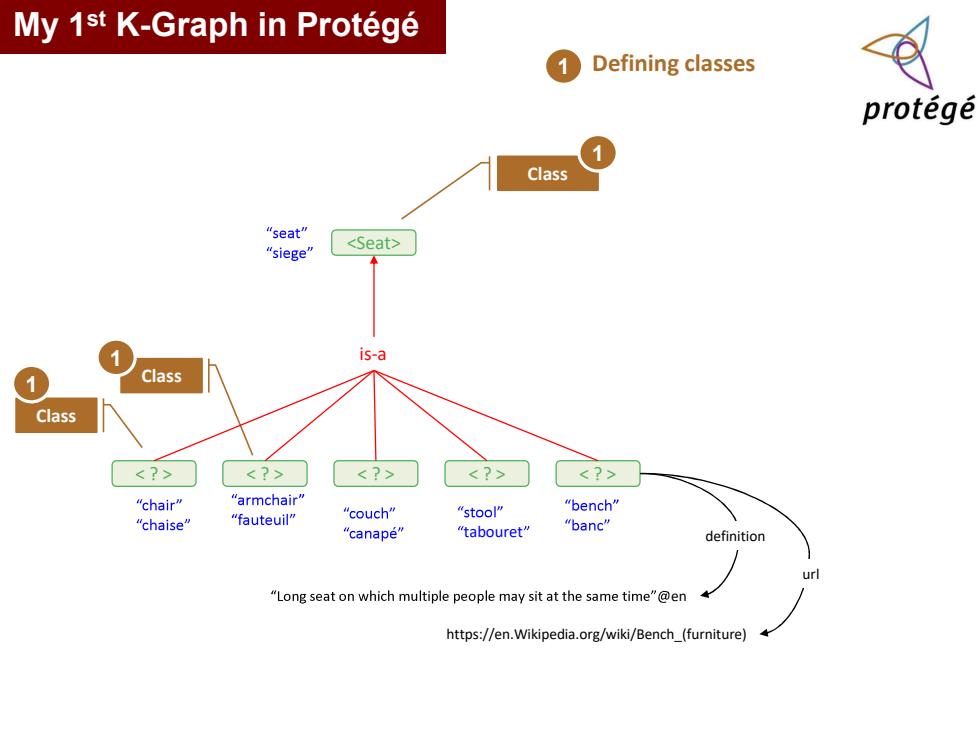
长椅

Building an OWL Ontology using Protégé: Named Classes



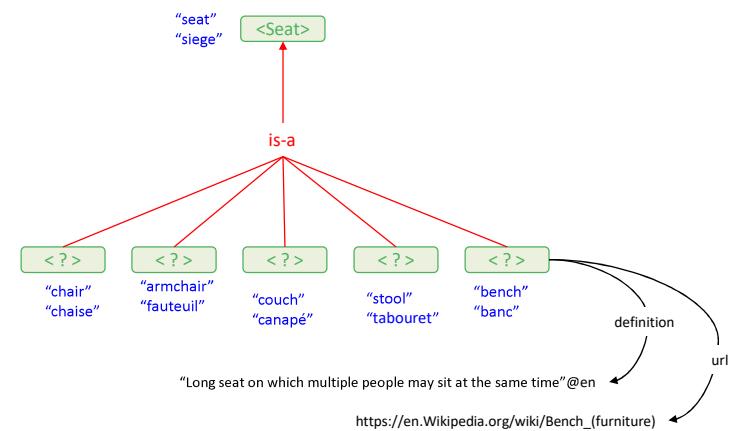
33

My 1st K-Graph in Protégé

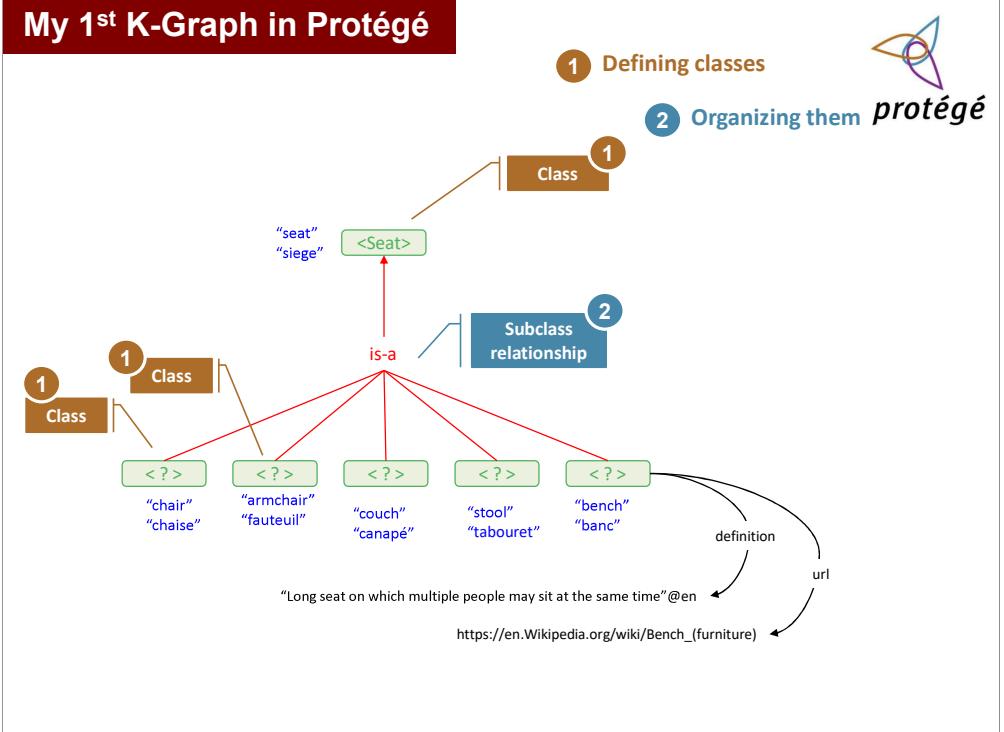


My 1st K-Graph in Protégé

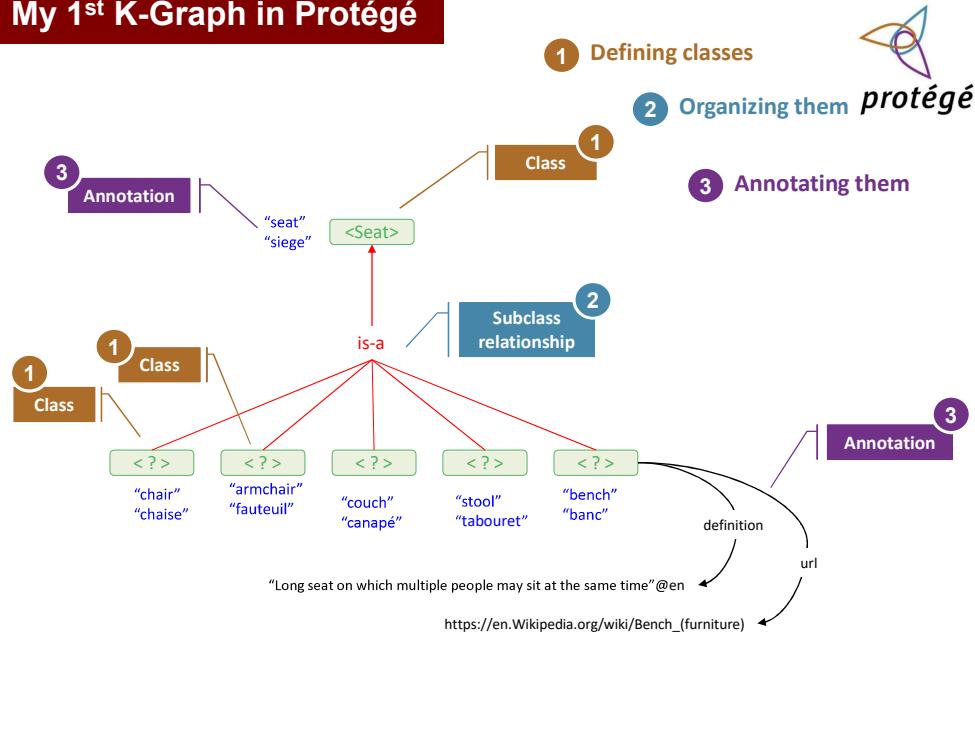
1. Building my 1st K-Graph in Protégé
2. Querying my 1st K-Graph in Protégé



My 1st K-Graph in Protégé



My 1st K-Graph in Protégé



38

Screenshot of the Protégé interface showing the ontology editor and metrics panel:

- File menu** (File, Edit, View, Reasoner, Tools, Refactor, Window) with various keyboard shortcuts.
- Ontology imports** panel: Shows "untitled-ontology-14" with its IRI and version information.
- Metrics** panel: Displays various ontology metrics such as Axiom count, Logical axiom count, Declaration axioms count, Class count, Object property count, Data property count, Individual count, and Annotation Property count.
- Annotations** panel: Shows annotations for the ontology.
- Object property axioms** panel: Shows SubObjectPropertyOf, EquivalentObjectProperties, and InverseObjectProperties counts.
- Imported ontologies** panel: Shows "Direct Imports" and "Indirect Imports" sections.

39

Screenshot of the Protégé interface showing the ontology header and documentation:

- Ontology header** panel: Shows the Ontology IRI (<http://ontology.ei.utoronto.ca/open311>), Ontology Version IRI (<http://ontology.ei.utoronto.ca/open311/1.11.0>), and annotations.
- Metrics** panel: Shows detailed metrics for the ontology, including Axiom count, Logical axiom count, Declaration axioms count, Class count, Object property count, Data property count, Individual count, and Annotation Property count.
- Ontology annotations** panel: Shows annotations for the ontology.
- Protégé 5 Documentation** panel: Provides documentation for the Ontology Header, Ontology IRI and Ontology Version IRI, and Ontology Annotations.

40



Adding a Tab

The screenshot shows the Protege Reasoner interface. The top navigation bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. A magnifying glass icon is positioned over the 'Window' menu. The main workspace displays tabs for Data properties, Active ontology, and Ontology header. Below these are sections for Ontology IRI, Ontology Version IRI, and Annotations. The bottom navigation bar features tabs for Ontology imports, Ontology Prefixes, and General class axioms. The status bar at the bottom right indicates 'To use the reasoner click Reasoner > Start reasoner' and '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 Individuals by class

Active ontology

Ontology header:

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

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

Annotations

Views

Tabs

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

Active ontology

Entities

Classes

Object properties

Data properties

Annotation properties

Individuals by class

OWLviz

DL Query

SWRLTab

OntoGraf

SQWRLTab

SPARQL Query

Classes

DisjointClasses

GCI count

Hidden GCI Count

Ontology imports Ontology Prefixes General class axioms

Imported ontologies:

Direct Imports

Indirect Imports

To use the reasoner click Reasoner > Start reasoner Show Inferences

41

1 Defining classes

The screenshot shows the Protégé 5 Documentation interface. The top navigation bar includes links for Installation, Getting Started, Views, Menus, and Class Expression Syntax. The main content area is titled "Class Hierarchy" under "Window > Views > Class views > Class hierarchy".

The left sidebar displays the "Active ontology" tree, which includes "owl:Thing" at the root level. A red circle highlights the "owl:Thing" node.

The central workspace shows the "owl:Thing" node expanded, revealing its sub-classes. A red circle highlights the "owl:Thing" node again. Below it, the "Annotations" tab is selected in the tab bar.

The right panel is titled "Class hierarchy" and shows a detailed tree structure of biological concepts. The tree starts with "owl:Thing" and branches into "owl:sameAs", "owl:equivalentClass", "owl:intersectionOf", "owl:unionOf", and "owl:complementOf". These further expand into categories like "biological_process", "cellular_component", "axon", "anatomical_part", "bacterial-type flagellum part", "biofilm", "cellular component", "cell", "cell surface part", "cell division site part", "cell junction", "cell junction'", "cell projection part", "cell septum part", "cell wall part", "Chloroplast part", "Contractile part", "ciliary part", "collagen and cuticulin-based cuticle ext", "contractile fiber part", and "cytoplasmatic part".

A female student with long brown hair, wearing a red hoodie with 'ALFA' printed on it, sits at a desk in a computer lab. She is looking down at a blue folder or document she is holding. Several computer monitors are visible in the background.

43



 protégé

The screenshot shows the OntoGraf interface for the ontology 'untitled-ontology-76'. The top navigation bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, Help, and a search bar. Below the header, tabs for Active ontology, Entities, Classes, Data properties, Individuals by class, DL Query, OntoGraf, and SPARQL Query are visible, with 'Data properties' currently selected. A red circle highlights the 'Classes' tab. The main content area displays ontology metrics and imports.

Ontology metrics:

Metrics	Count
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:

- Imported ontologies:
 - Direct Imports
 - Indirect Imports

To use the reasoner click Reasoner > Start reasoner Show Inferences

42

1 Defining classes

Adding Views to a Tab

The screenshot shows the Protégé interface with two windows. The left window displays the 'Annotations' tab in the left panel and the 'Annotations' view in the main area. The right window displays the 'Individuals' tab in the left panel and the 'Individuals' view in the main area. Red arrows highlight the 'Annotations' tab and the 'Individuals' tab, indicating where to click to open their respective views.

44

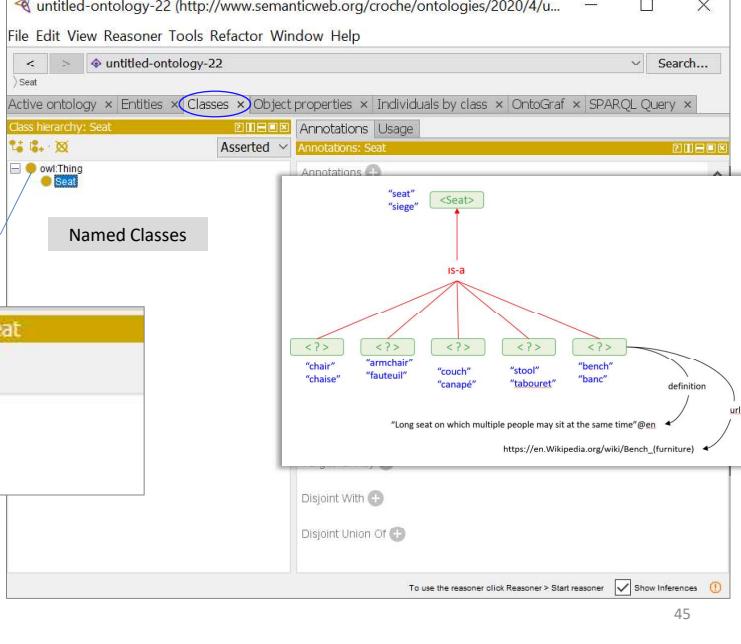
1 Defining classes

2 Organizing them



The class Thing represents the set containing all individuals

Named Classes



Class hierarchy: Seat

Annotations: Seat

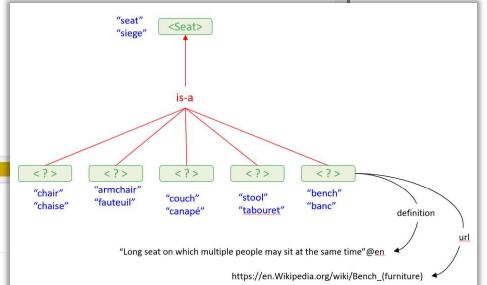
Description: Seat

To use the reasoner click Reasoner > Start reasoner Show Inferences



1 Defining classes

2 Organizing them

Class hierarchy: Chair

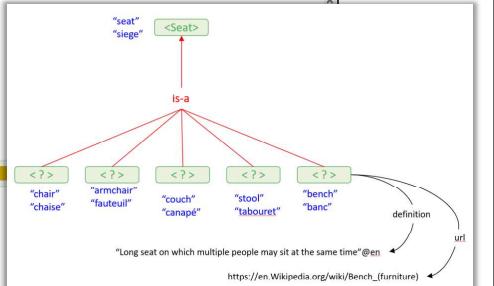
Annotations: Chair

Description: Chair

To use the reasoner click Reasoner > Start reasoner Show Inferences

1 Defining classes

2 Organizing them

Class hierarchy: Seat

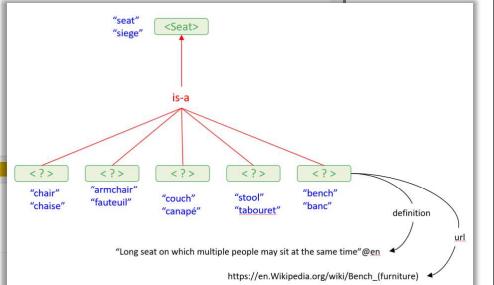
Annotations: Seat

Description: Seat

To use the reasoner click Reasoner > Start reasoner Show Inferences

1 Defining classes

2 Organizing them

Class hierarchy: Chair

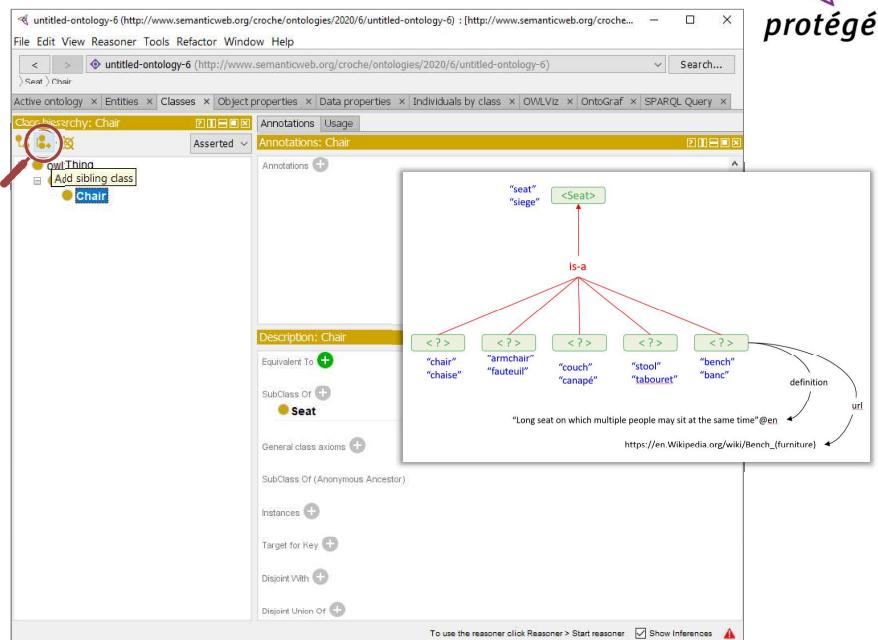
Annotations: Chair

Description: Chair

To use the reasoner click Reasoner > Start reasoner Show Inferences

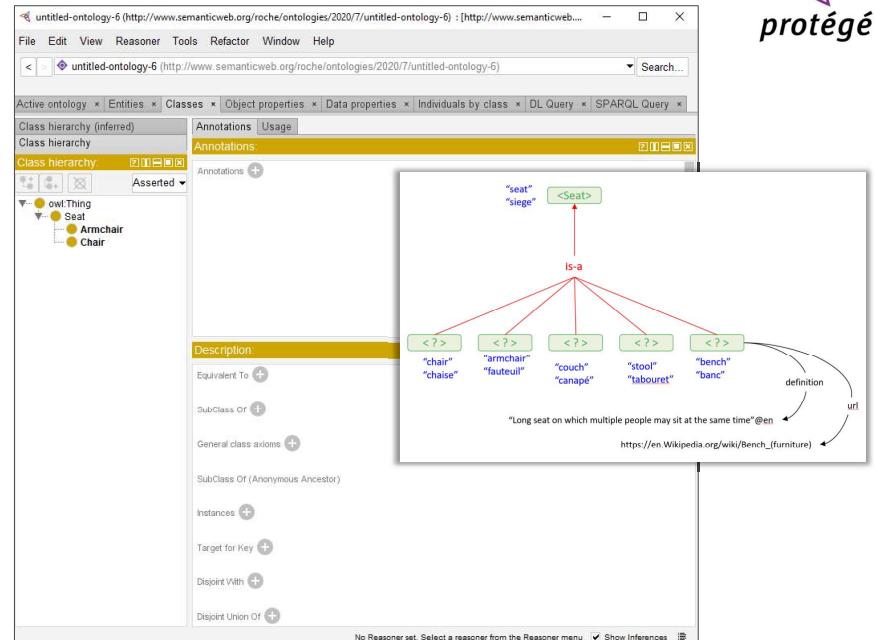
1 Defining classes

2 Organizing them



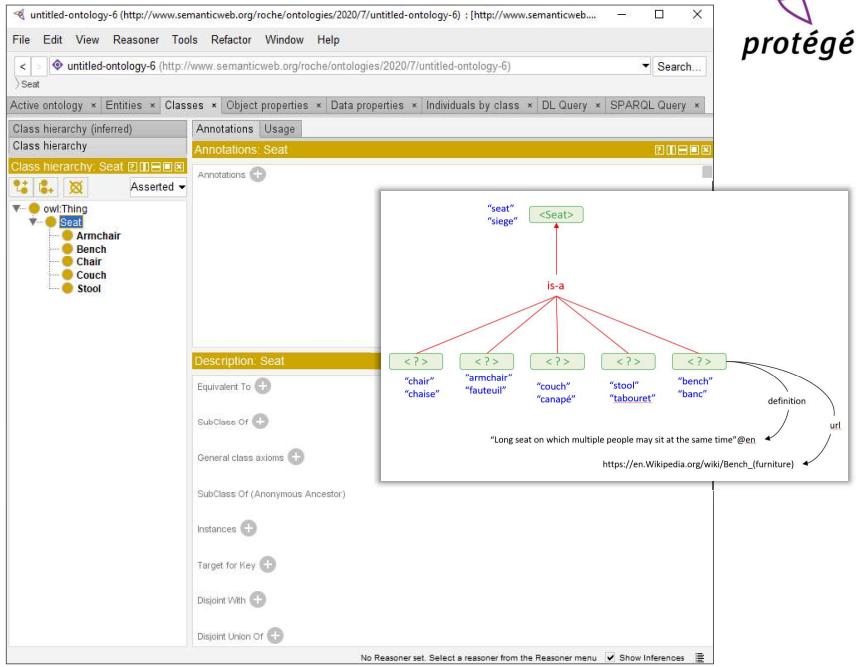
1 Defining classes

2 Organizing them



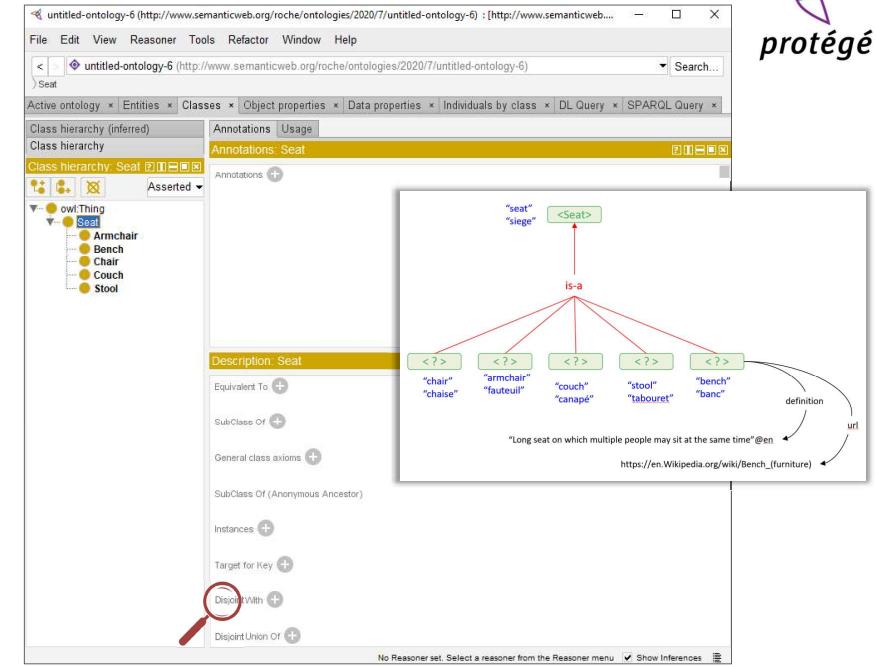
1 Defining classes

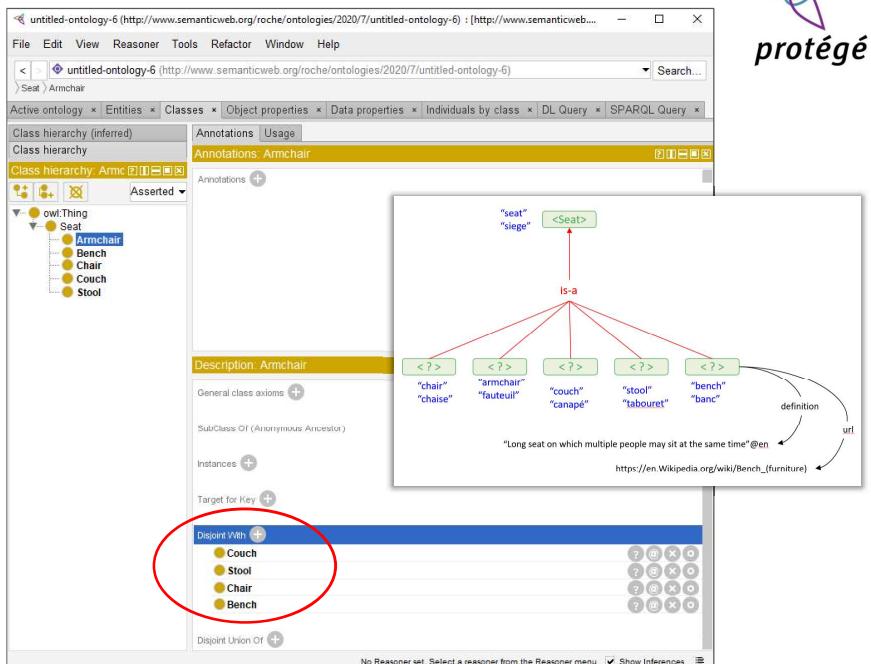
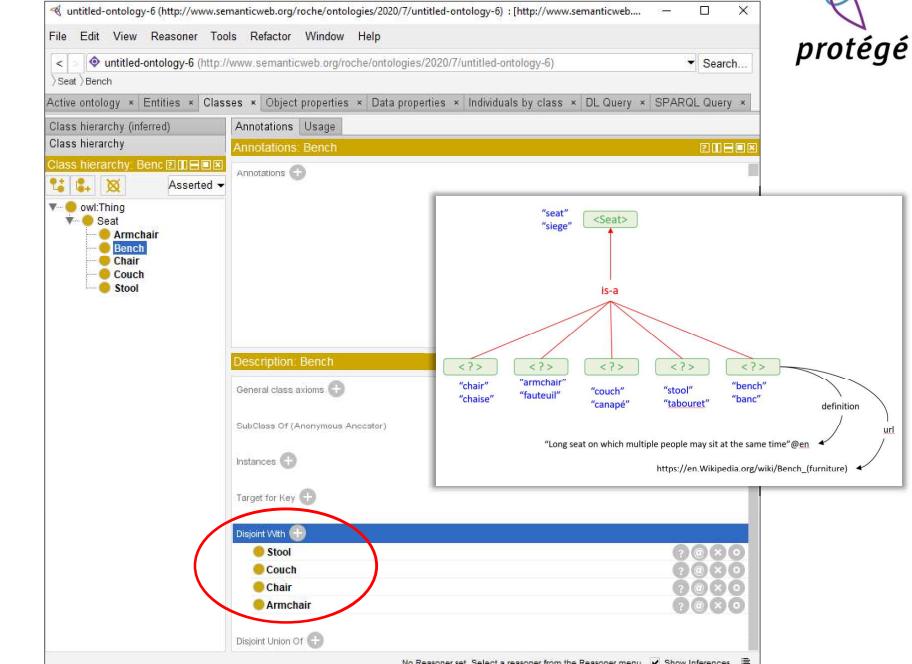
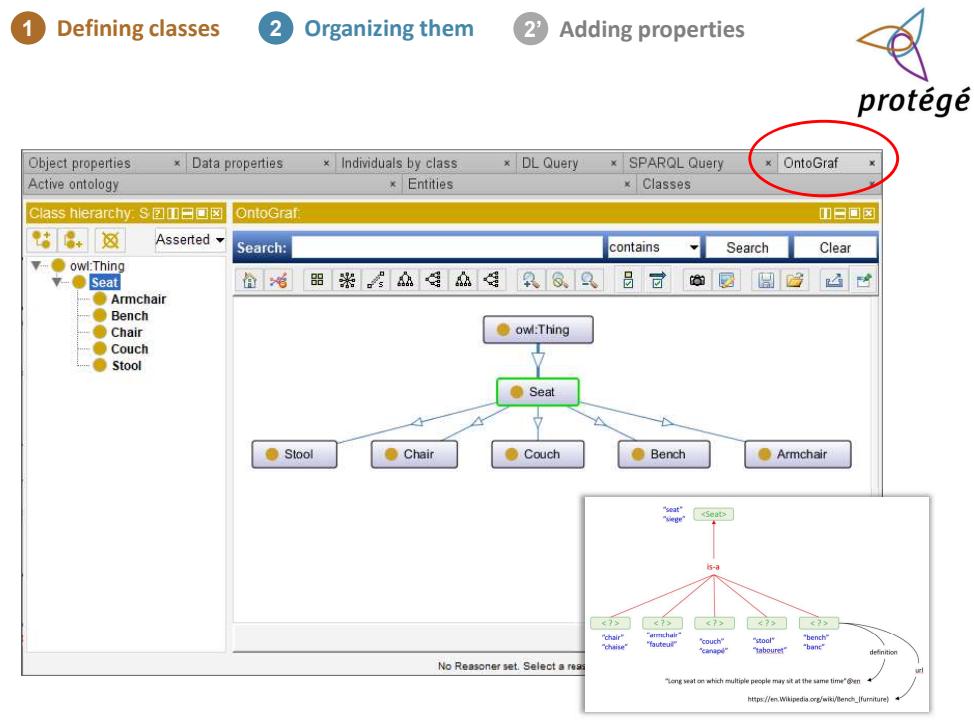
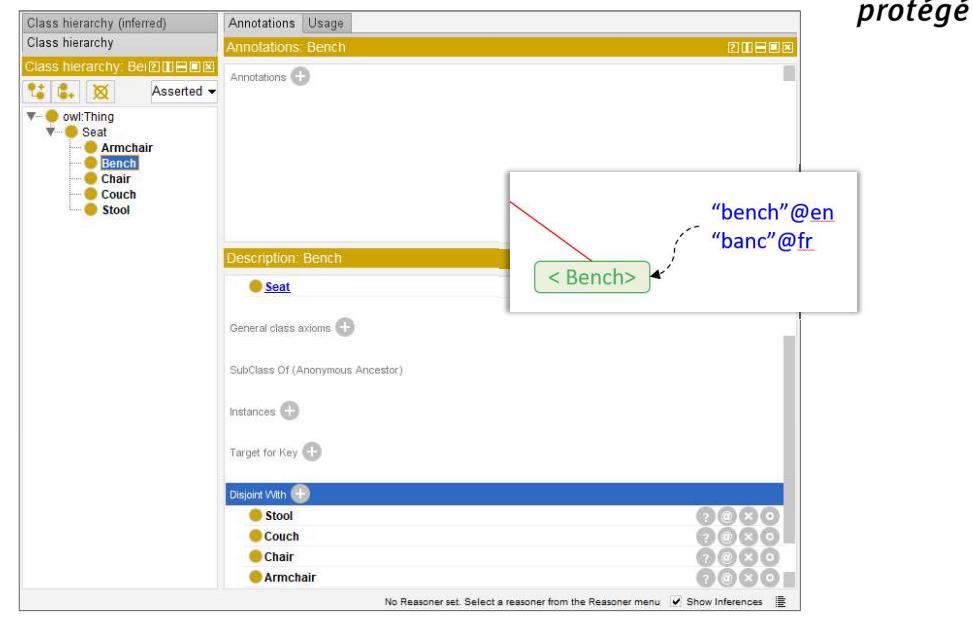
2 Organizing them



1 Defining classes

2 Organizing them



1 Defining classes**2** Organizing them**2'** Adding properties**1** Defining classes**2** Organizing them**2'** Adding properties**1** Defining classes**2** Organizing them**2'** Adding properties**1** Defining classes**2** Organizing them**3** Annotating them

1 Defining classes **2 Organizing them** **3 Annotating them**

Bench

Annotations Usage

Annotations Bench

Annotations +

- owl:backwardCompatibleWith
- owl:deprecated
- owl:incompatibleWith
- owl:priorVersion
- owl:versionInfo
- rdfs:comment
- rdfs:isDefinedBy
- rdfs:label
- rdfs:seeAlso

Type: Literal Entity IRI IRI Editor Property values

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**

Bench

Annotations Usage

Annotations Bench

Annotations +

- owl:backwardCompatibleWith
- owl:deprecated
- owl:incompatibleWith
- owl:priorVersion
- owl:versionInfo
- rdfs:comment
- rdfs:isDefinedBy
- rdfs:label
- rdfs:seeAlso

Type: xsd:string Lang: en

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**

Bench

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

1 Defining classes **2 Organizing them** **3 Annotating them**

Bench

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

protégé

Annotations: Bench

- 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

Bench

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

1 Defining classes**2** Organizing them**3** Annotating them

protégé

Annotations: Bench

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

Description: Bench

Bench

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

1 Defining classes**2** Organizing them**3** Annotating them**1** Defining classes**2** Organizing them**3** Annotating them

1 Defining classes

2 Organizing them

3 Annotating them

protégé

Annotations Tab:

- owl:backwardCompatibleWith
- owl:deprecated
- owl:incompatibleWith
- owl:priorVersion
- owl:versionInfo
- rdfs:comment
- rdfs:isDefinedBy
- rdfs:label
- rdfs:seeAlso

Annotation Details:

- Type: xsd:string
- Lang: en
- Value: "long seat on which multiple people may sit at the same time"
- URL: https://en.wikipedia.org/wiki/Bench_(furniture)

Diagram:

```

graph TD
    Bench -- definition --> "long seat on which multiple people may sit at the same time"
    Bench -- url --> https://en.wikipedia.org/wiki/Bench_(furniture)
  
```

Visualisation

OntoGraf

Class hierarchy: Seat

Asserted:

- owl:Thing
- Part
- Seat
- Armcchair
- Bench
- Chair
- Couch
- SeatWithBack
- Stool

Search: contains

Diagram Labels:

- owl:Thing
- Part
- Seat
- Armcchair
- Bench
- Chair
- Couch
- SeatWithBack
- Stool
- Feet
- Back
- Arms

67

1 Defining classes

2 Organizing them

3 Annotating them

rotégé

Annotations Tab:

- 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)

Description Tab:

- Instances +
- Target for Key +
- Disjoint With +
- Stool
- Couch
- Chair
- Armcchair

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

Visualisation

OntoGraf

Class hierarchy: seat

Asserted:

- owl:Thing
- SeatWithBack
- furniture
- part
- seat
- bench
- chair
- couch
- armchair
- Stool

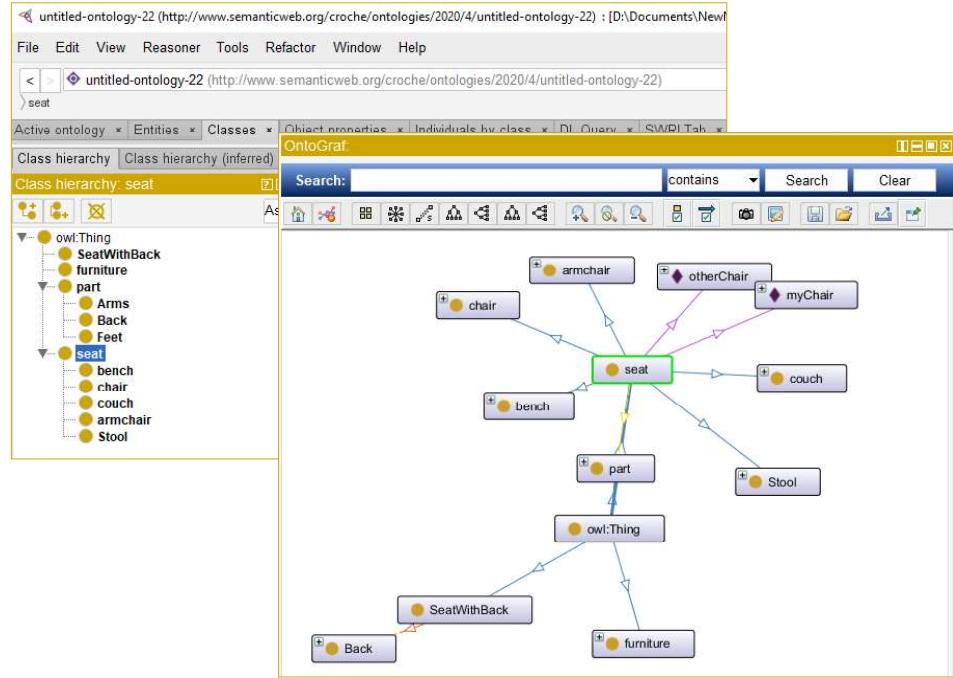
Search: contains

Diagram Labels:

- owl:Thing
- part
- seat
- Back
- Arms
- Feet
- chair
- Stool
- myChair
- couch
- bench
- armchair
- SeatWithBack

68

Visualisation



2. Querying my 1st K-Graph in Protégé

The screenshot shows the Protégé interface. The 'Annotations' tab is selected, and the 'Annotations: Seat' panel is open. It contains annotations for the 'Seat' class, such as rdfs:label in French ('siège') and English ('bench'). A modal dialog box is open, asking to choose an ontology format to save the ontology. The 'RDF/XML Syntax' option is selected. Other tabs visible include 'Class hierarchy (inferred)', 'Object properties', 'Data properties', 'Individuals by class', 'DL Query', and 'SPARQL Query'.



protégé

2. Querying my 1st K-Graph in Protégé

The screenshot shows the Protégé interface with the 'SPARQL Query' tab selected. A query is being typed into the editor:

```

PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX roche: <http://www.semanticweb.org/roche/ontologies/2020/7/untilted-ontology-6#>

SELECT ?x WHERE {
    ?x rdfs:label ?label .
    FILTER (lang(?label) = 'fr')
}
  
```

2. Querying my 1st K-Graph in Protégé

```

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

<!--
// Classes
-->

<!-- http://www.semanticweb.org/roche/ontologies/2020/7/untilted-ontology-6#Armchair -->
<owl:Class rdf:about="http://www.semanticweb.org/roche/ontologies/2020/7/untilted-ontology-6#Armchair">
    <owl:subClassOf rdf:resource="http://www.semanticweb.org/roche/ontologies/2020/7/untilted-ontology-6#Seat"/>
    <owl:disjointWith rdf:resource="http://www.semanticweb.org/roche/ontologies/2020/7/untilted-ontology-6#Couch"/>
    <owl:disjointWith rdf:resource="http://www.semanticweb.org/roche/ontologies/2020/7/untilted-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>

<owl:Class rdf:about="http://www.semanticweb.org/roche/ontologies/2020/7/untilted-ontology-6#Seat">
    <owl:subClassOf rdf:resource="http://www.semanticweb.org/roche/ontologies/2020/7/untilted-ontology-6#Couch"/>
    <owl:disjointWith rdf:resource="http://www.semanticweb.org/roche/ontologies/2020/7/untilted-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>
  
```



protégé

2. Querying my 1st K-Graph in Protégé

The screenshot shows the Protégé interface with the SPARQL Query tab highlighted by a red circle. The main workspace displays annotations for the class 'Seat'. Below the annotations, the 'Description: Seat' panel is visible, containing sections for General class axioms, SubClass Of (Anonymous Ancestor), Instances, Target for Key, Disjoint With, and Disjoint Union Of.



2. Querying my 1st K-Graph in Protégé

The screenshot shows the Protégé interface with a SPARQL query window open. 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 ?subject ?object
WHERE { ?subject rdfs:subClassOf ?object }
```

The main workspace shows the class hierarchy for 'Armchair' with 'Seat' as a subclass. The 'Annotations: Armchair' panel is also visible.



2. Querying my 1st K-Graph in Protégé

The screenshot shows the Protégé menu with the 'Tools' option expanded. The 'SPARQL Query' option is highlighted with a blue selection bar.



2. Querying my 1st K-Graph in Protégé

The screenshot shows the Protégé interface with the results of the SPARQL query. The results table lists 'Bench', 'Chair', 'Armchair', 'Stool', and 'Couch' as subclasses of 'Seat'. The main workspace shows the class hierarchy for 'Armchair' with 'Seat' as a subclass. The 'Annotations: Armchair' panel is also visible.



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
```

Annotations Usage Annotations Bench

Class hierarchy (inferred) Class hierarchy

Class hierarchy Asserted

Annotations +

owl:Thing

- Seat
- Bench
- Chair
- Couch
- Stool

rdfs:label [language: fr]

bank

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 +



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



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
```

untitled-ontology-22

DL Query x OntoGraf x SPARQL Query x Object properties x Active ontology x Entities x Classes x Data properties x Individuals by class 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 ?specificClass ?genericClass
WHERE { ?specificClass rdfs:subClassOf ?genericClass }
ORDER BY ?specificClass
```

specificClass

- Armchair
- Armchair
- Armchair
- Armchair
- Armchair
- Arms
- Back

genericClass

- Seat
- Part
- Part

hasPart some Back

hasPart some Feet

hasPart some Arms

hasPart2 value with-back

Part

Execute

78

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 ?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
---------	---------

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
```

Annotations Usage Annotations Bench

Class hierarchy (inferred) Class hierarchy

Class hierarchy Asserted

Annotations +

owl:Thing

- Seat
- Bench
- Chair
- Couch
- Stool

rdfs:label [language: fr]

bank

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 +



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



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
```

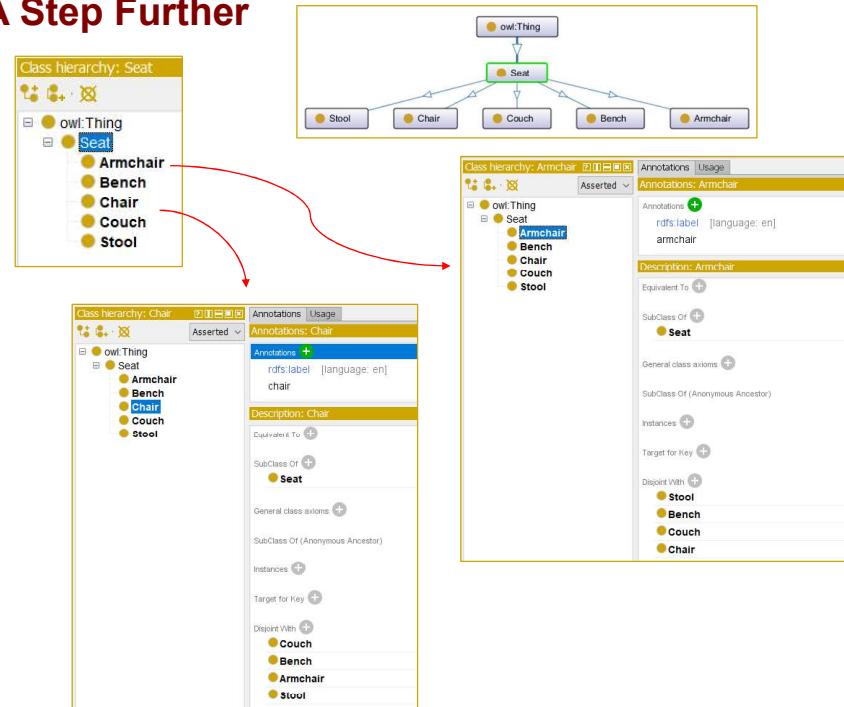
name definition

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



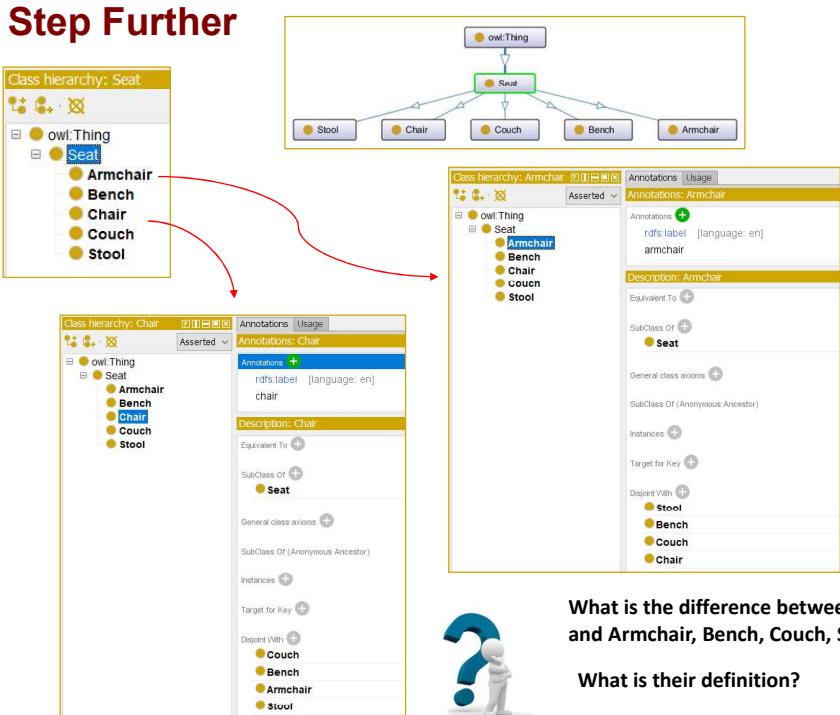
We need go a step further

A Step Further



86

A Step Further



87

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 with 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 with 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>
```

88

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 with 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>
```

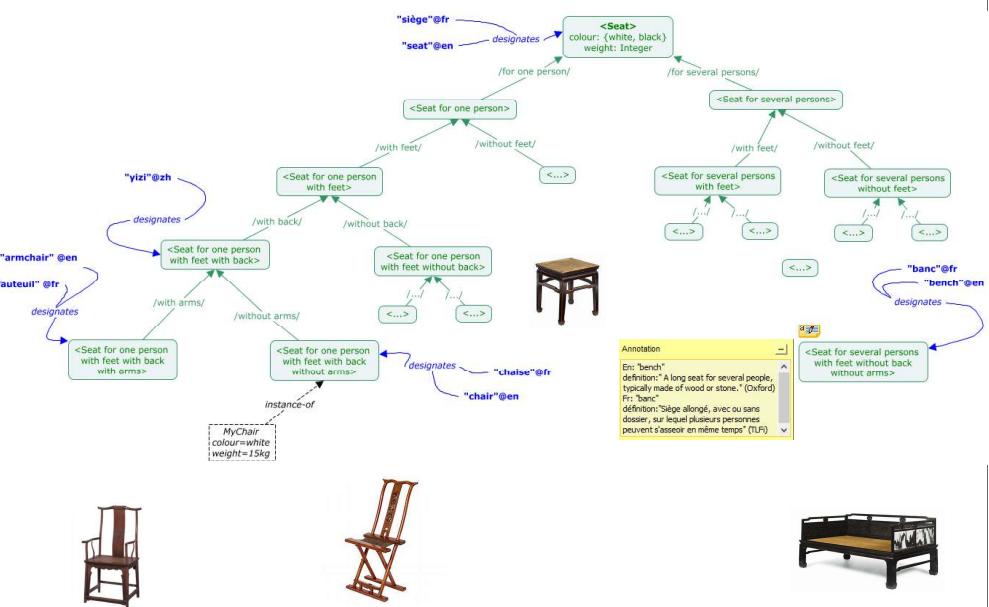
```
<owl:Class rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Armchair">
  <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Seat"/>
  <rdfs:comment xml:lang="en">Seat for one person with feet and back with arms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

General axioms

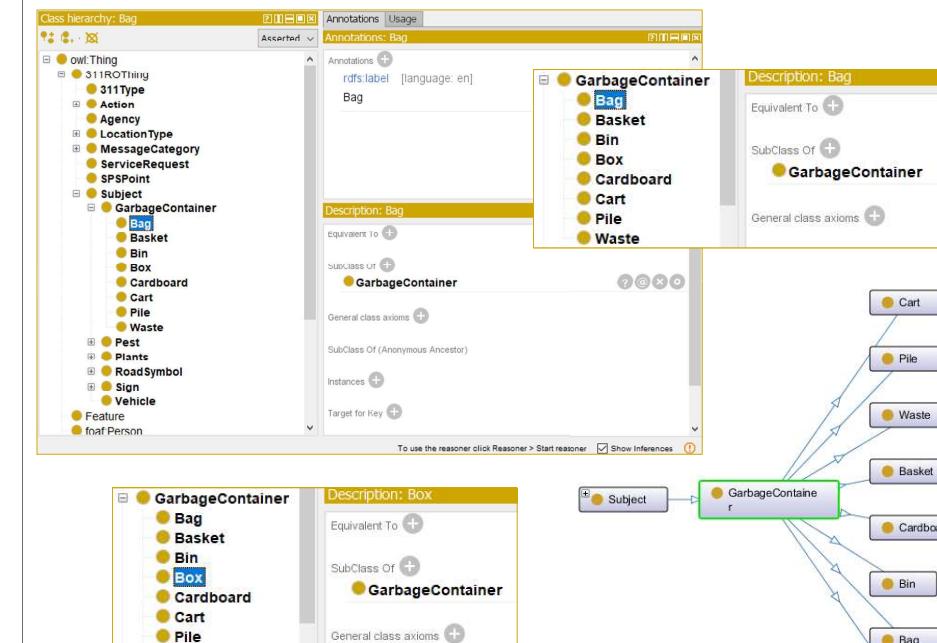
```
<rdf:Description>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#AllDisjointClasses"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22#Armchair"/>
    <rdf:Description rdf:about="http://www.semanticweb.org/croche/ontologies/2020/4/untitled-ontology-22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

89

Ontoterminology of Seats



91



90

Array of differences

Objects	Concepts	Axis of analysis		Axis of analysis		Axis of analysis		Axis of analysis		Terms	
		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	"chair"	"chaise"
	<Seat 1 person with feet with back with arms>	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"	"canape"
	<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>

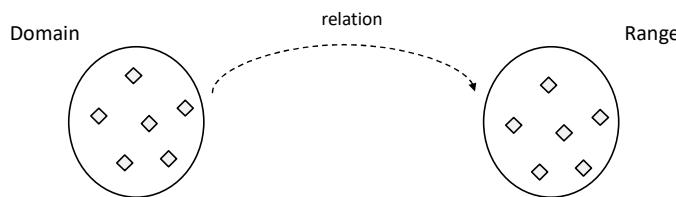
::= <Seat> + /for one person/ + /with feet/ + /with back/ + /with arms/



92

Building an OWL Ontology using Protégé: Object Properties

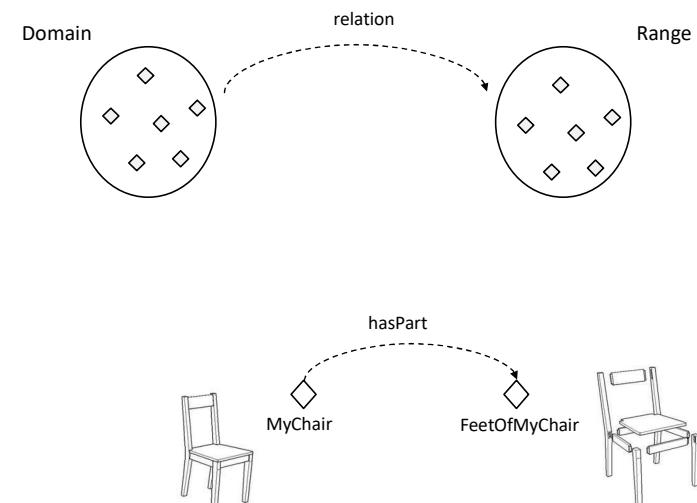
Relationships between individuals



93

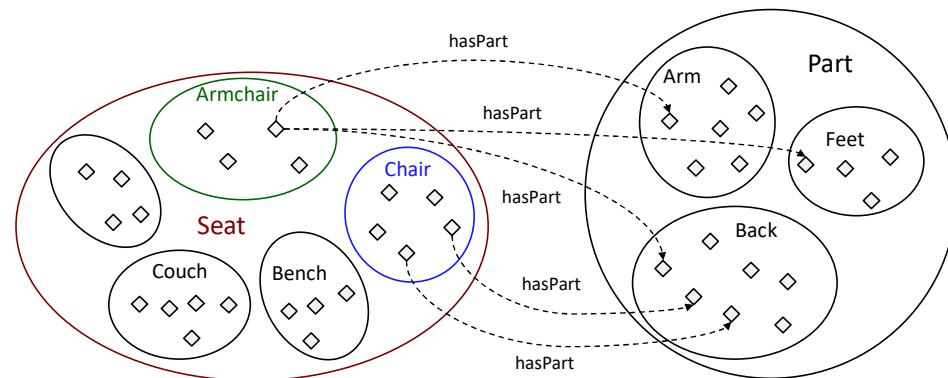
Building an OWL Ontology using Protégé: Object Properties

Relationships between individuals



94

Building an OWL Ontology using Protégé: Object Properties



95

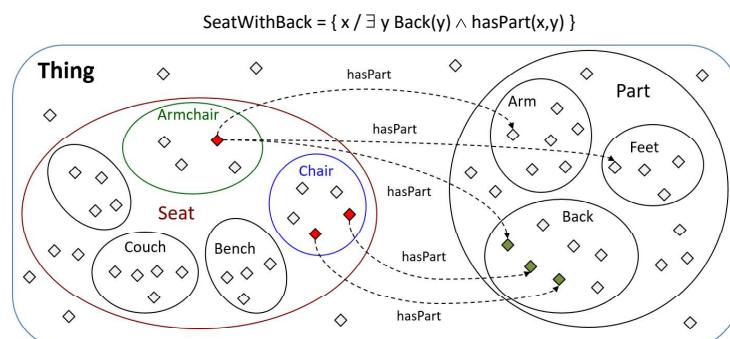
Building an OWL Ontology using Protégé: Defining Classes

(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).



96

Building an OWL Ontology using Protégé: Object Properties

The diagram illustrates a simple ontology with two chair individuals. A dashed arrow connects 'MyChair' to 'FeetOfMyChair' with the label 'hasPart'. Below this, the Protégé interface shows the 'Object property hierarchy: hasPart' panel. The 'owl:topObjectProperty' node is expanded, showing 'hasPart' as a child. The 'Transitive' checkbox is checked, and the 'Ranges (intersection)' section lists 'Seat' and 'Part'.

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 creating a new class 'SeatWithBack'. In the 'Class hierarchy' panel, 'SeatWithBack' is shown as a subclass of 'Seat'. In the 'Annotations' tab, the annotation 'Annotations: SeatWithBack' is selected. In the 'Class expression editor' panel, under 'SubClass Of', there is a '+ Add' button highlighted with a blue circle. The 'Restricted property' section shows 'hasPart' and 'some Back'.

Building an OWL Ontology using Protégé: Data Properties

relationships between an individual and data values.

The diagram shows a brown wooden chair. Below it, the text 'MyChair color: "brown"' is displayed. The Protégé interface shows the 'Data property hierarchy: color' panel. The 'owl:topDataProperty' node is expanded, showing 'color' as a child. The 'Functional' checkbox is checked, and the 'Ranges' section lists 'xsd:string'.

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	Things that have a pet that is a Dog 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).
value	hasPet value Tibbs	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.
only	hasPet only Cat	Things that have a pet that is Tibbs. 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".
min	hasPet min 3 Cat	Things that have at least three pets that are Cats. Things that have at least three pets that are Cats. Also known as a "Min cardinality restriction".

Essential Characteristics: Property Restriction



Class hierarchy: Armchair

Annotations [Usage]

Annotations: Armchair

- owl:Thing
- Part
- Arms
- Back
- Feet
- Seat
- Armchair**
- Bench
- Chair
- Couch
- SeatWithBack
- Stool

Annotations (1)

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



101

Essential Characteristics: Property Restriction



Class hierarchy: Armchair

Annotations [Usage]

Annotations: Armchair

- owl:Thing
- Part
- Arms
- Back
- Feet
- Seat
- Armchair**
- Bench
- Chair
- Couch
- SeatWithBack
- Stool

Annotations (1)

rdfs:label ([language: fr]) fauteuil

rdfs:label ([language: en]) armchair

Description: Armchair

Equivalent To +

SubClass Of +

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



Without arms?



103

Essential Characteristics: Property Restriction



Class hierarchy: Armchair

Annotations [Usage]

Annotations: Armchair

- owl:Thing
- Part
- Arms
- Back
- Feet
- Seat
- Armchair**
- Bench
- Chair
- Couch
- SeatWithBack
- Stool

Annotations (1)

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



102

Essential Characteristics: Property Restriction



Class hierarchy: Armchair

Annotations [Usage]

Annotations: Armchair

- owl:Thing
- Part
- Arms
- Back
- Feet
- Seat
- Armchair**
- Bench
- Chair
- Couch
- SeatWithBack
- Stool

Annotations (1)

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

- owl:Thing
- Part
- Arms
- Back
- Feet
- Seat
- Armchair**
- Bench
- Chair
- Couch
- SeatWithBack
- Stool

Annotations (1)

rdfs:label ([language: en]) chaise

rdfs:comment ([language: fr]) chaise

Description: Chair

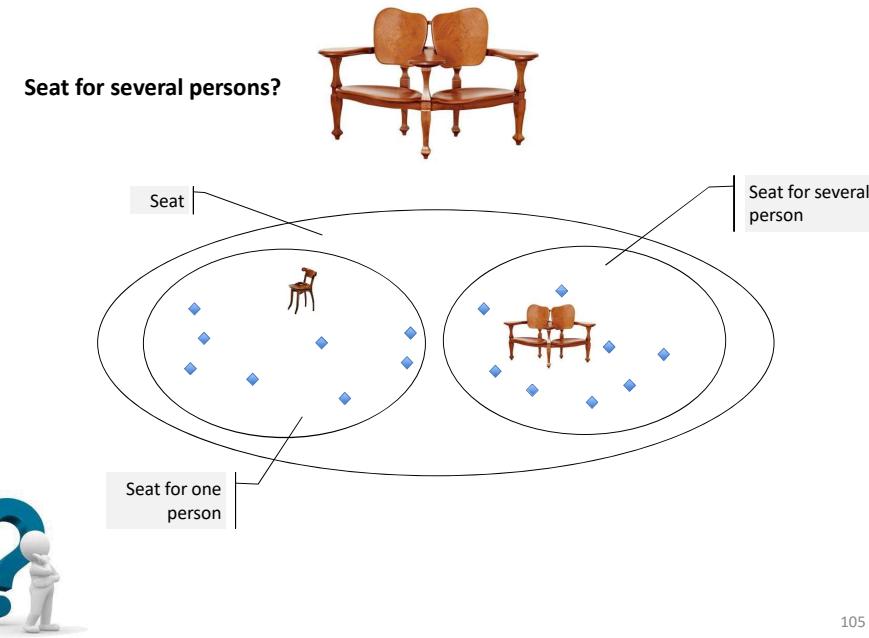
Equivalent To +

SubClass Of +

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

104

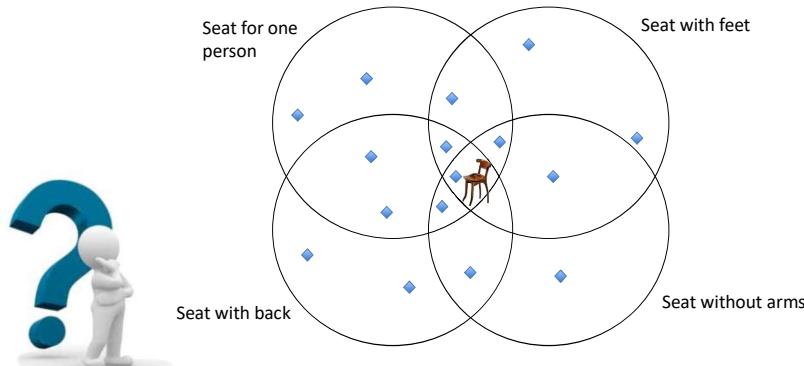
Essential Characteristics: Class



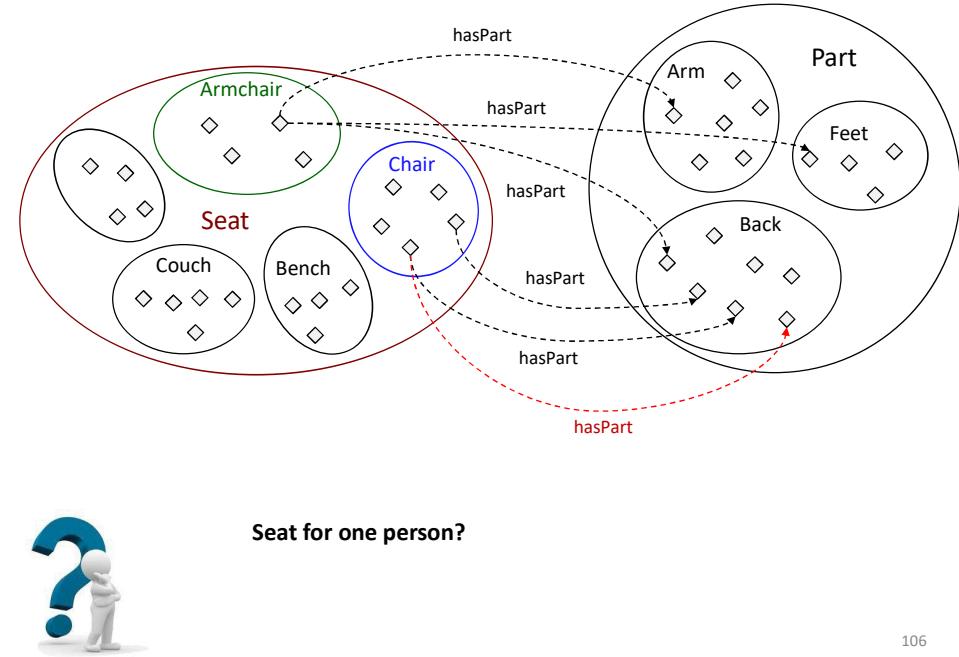
Building an OWL Ontology using Protégé: Essential Characteristics



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



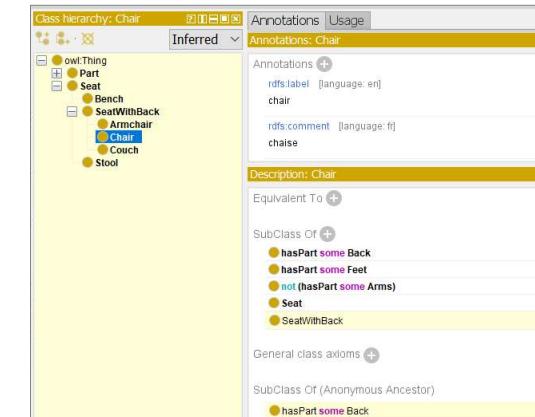
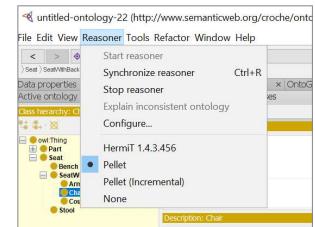
Building an OWL Ontology using Protégé: Essential Characteristics



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.



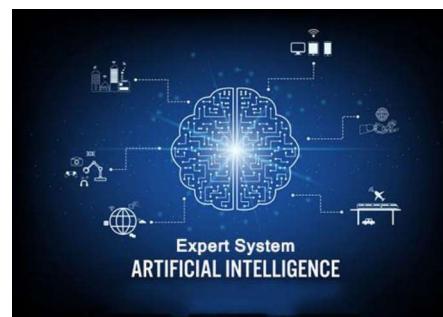
Reasoner

The screenshot shows the Protege reasoner interface with two main panels. The left panel displays the class hierarchy for 'SeatWithBack' under 'Annotations' tab, listing owl:Thing, furniture, part, seat, and various subclasses like armchair, bench, chair, couch, and Stool. A magnifying glass icon highlights the 'SeatWithBack' node. The right panel shows the inferred class hierarchy for 'SeatWithBack' under the 'Inferred' tab, listing the same classes but with additional inferred relationships. Below these panels is a detailed explanation window for 'SeatWithBack SubClassOf seat'. It includes options to show regular or iconic justifications, and a specific entry: 'SeatWithBack SubClassOf hasPart some Back' with 'hasPart Domain seat'. A yellow arrow points from the bottom-left of the main interface towards this explanation window.

Rules and the Semantic Web

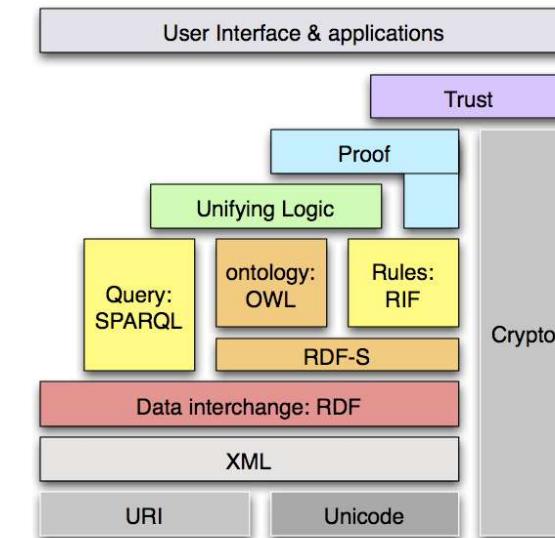
Rule-based Systems (Expert Systems)

- Engineering: Diagnosis rules
- Commerce: Business rules
- Law: Legal reasoning
- Medicine: Eligibility, Compliance
- Internet: Access authentication



Rules and the Semantic Web

<https://protege.stanford.edu/conference/2009/slides/SWRL2009ProtegeConference.pdf>



Semantic web standards (Semantic Web Layer)

Rules and the Semantic Web



<https://www.w3.org/Submission/SWRL/>

SWRL: A Semantic Web Rule Language Combining OWL and RuleML

The proposed rules are of the form of an implication between an antecedent (body) and consequent (head). The intended meaning can be read as: whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold.

Rules saved as part of ontology

Rules and the Semantic Web

The screenshot shows the Crotchet Ontology Editor interface. The 'untitled-ontology-22' tab is active. The 'Window' menu is open, with 'Tabs' selected. The 'SWRLTab' option is highlighted with a blue box.

113

Rules and the Semantic Web

The screenshot shows the Crotchet Ontology Editor interface. The 'untitled-ontology-22' tab is active. A magnifying glass icon highlights the 'Edit' button in the 'Control' panel. The 'OWL+SWRL->Drools' button is also circled.

114

Rules and the Semantic Web: Examples

Reclassification

$\text{Man}(\text{?m}) \rightarrow \text{Person}(\text{?m})$

$\text{Seat}(\text{?x}) \rightarrow \text{Furniture}(\text{?m})$

The screenshot shows the Crotchet Ontology Editor interface. The 'Class hierarchy: Seat' tab is active. A magnifying glass icon highlights the 'Annotations' section for 'myChair'. Below it, the 'Description: myChair' section is shown with an explanation for 'myChair Type Furniture'.

Rules and the Semantic Web: Examples

Property value assignment

$\text{Person}(\text{?p}) \wedge \text{hasSibling}(\text{?p}, \text{?s}) \wedge \text{Man}(\text{?s}) \rightarrow \text{hasBrother}(\text{?p}, \text{?s})$

$\text{hasPart}(\text{?x}, \text{?y}) \wedge \text{hasPart}(\text{?x}, \text{?z}) \rightarrow \text{linked}(\text{?y}, \text{?z})$

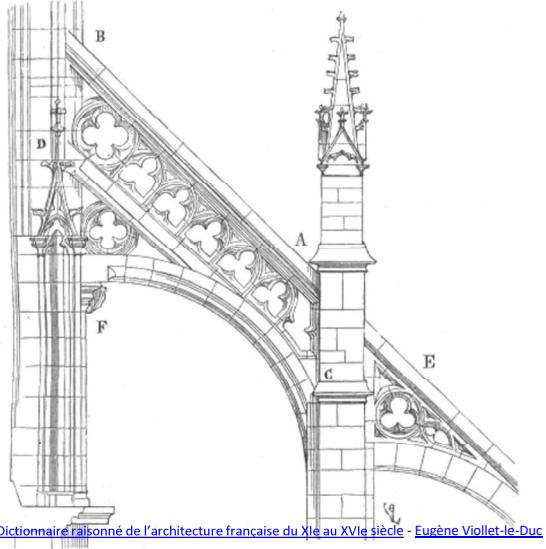
The screenshot shows the Crotchet Ontology Editor interface. The 'Annotations: feetOfMyChair' tab is active. A magnifying glass icon highlights the 'Object property assertions' section for 'linked backOfMyChair'. To the right, three explanations are listed:

- Explanation 1: Explanation for: feetOfMyChair linked backOfMyChair
- Explanation 2: Explanation for: feetOfMyChair linked backOfMyChair
 - Symmetric: linked
 - backOfMyChair linked feetOfMyChair
- Explanation 3: Explanation for: feetOfMyChair linked backOfMyChair
 - myChair hasPart backOfMyChair
 - myChair hasPart feetOfMyChair
 - hasPart(?x, ?y), hasPart(?x, ?z) > linked(?y, ?z)



OntoTerminology Editor

<http://ontoterminology.com/> May 2020



Prof. Christophe Roche

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

KETRC Research Centre
University of Liaocheng (China)
<http://christophe-roche.fr/>

117