

Course: Intelligent Systems

Unit 3: Ontology Engineering

Using Protégé (Desktop)

Mari Carmen Suárez de Figueroa Baonza

Technical University of Madrid



Index

- Brief Introduction
- Main Modelling Components
- How to use it

Introduction (I): What is Protégé?

- A free, open-source **ontology editor** and **knowledge-base framework** for building intelligent systems
- It is based on Java, is extensible, and provides a plug-and-play environment
 - **Plug-ins** for extending the core functionality
 - Examples are OWLViz and Graphviz
- It is supported by a **strong community of developers** and academic, government and corporate users
- It is used **to build knowledge-based solutions** in areas as diverse as biomedicine, e-commerce, and organizational modelling



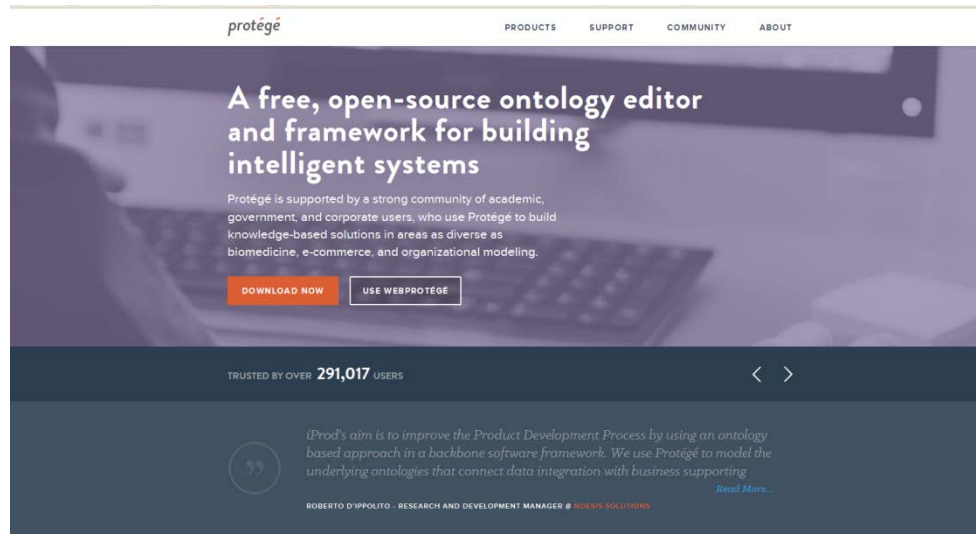
Introduction (II): What can you do with Protégé?

- To create a new **ontology (model) from scratch**
- To download and **extend an existing ontology (model)**
- To **export ontologies (models)** in a variety of formats
 - OWL/XML
 - RDF/XML

Introduction (III): Where you can find Protégé?



- **Homepage:** <http://protege.stanford.edu/>
- It is available for **downloading** at
 - <https://protege.stanford.edu/products.php#desktop-protege>

A screenshot of the Protégé homepage. The header features the Protégé logo and navigation links: PRODUCTS, SUPPORT, COMMUNITY, and ABOUT. The main content area has a large heading: "A free, open-source ontology editor and framework for building intelligent systems". Below this, a paragraph states: "Protégé is supported by a strong community of academic, government, and corporate users, who use Protégé to build knowledge-based solutions in areas as diverse as biomedicine, e-commerce, and organizational modeling." There are two buttons: "DOWNLOAD NOW" (orange) and "USE WEBPROTEGE" (white with a black border). At the bottom, it says "TRUSTED BY OVER 291,017 USERS" with navigation arrows. A testimonial from Roberto D'Ipollito is also visible, mentioning iProd's aim to improve the Product Development Process.

protégé

PRODUCTS SUPPORT COMMUNITY ABOUT

A free, open-source ontology editor and framework for building intelligent systems

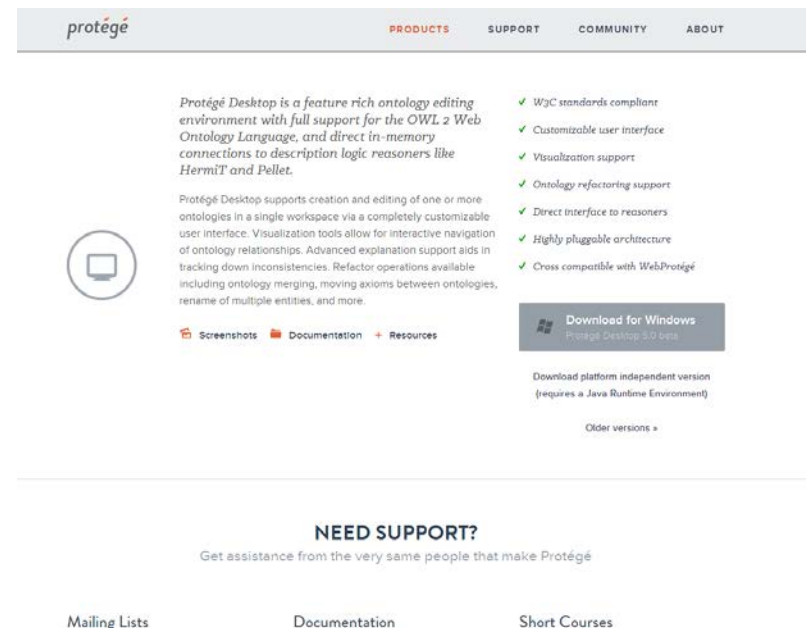
Protégé is supported by a strong community of academic, government, and corporate users, who use Protégé to build knowledge-based solutions in areas as diverse as biomedicine, e-commerce, and organizational modeling.

[DOWNLOAD NOW](#) [USE WEBPROTEGE](#)

TRUSTED BY OVER **291,017** USERS

iProd's aim is to improve the Product Development Process by using an ontology based approach in a backbone software framework. We use Protégé to model the underlying ontologies that connect data integration with business supporting

ROBERTO D'IPOLLITO - RESEARCH AND DEVELOPMENT MANAGER @ **NOESIS SOLUTIONS**

A screenshot of the Protégé products page. The header features the Protégé logo and navigation links: PRODUCTS, SUPPORT, COMMUNITY, and ABOUT. The main content area has a heading: "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." Below this, a paragraph states: "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." There are two buttons: "Screenshots" (orange) and "Documentation + Resources" (white with a black border). At the bottom, it says "NEED SUPPORT?" and "Get assistance from the very same people that make Protégé".

protégé

PRODUCTS SUPPORT COMMUNITY ABOUT

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](#)

Download for Windows
Protégé Desktop 3.0 beta

Download platform independent version (requires a Java Runtime Environment)

[Older versions](#)

NEED SUPPORT?

Get assistance from the very same people that make Protégé

[Mailing Lists](#) [Documentation](#) [Short Courses](#)

Introduction (IV): How does Protégé works?

- Objects in the domain are expressed through a series of interrelated **classes**
- **Class hierarchy** is similar to that used by object-oriented languages such as Java
 - Superclasses
 - Subclasses
 - Sibling Classes
- Heavy reliance on inheritance
 - Protégé supports **multiple inheritance**

Introduction (V): Notes

- Screenshots provided in this presentation were made using **Protégé 4.3**
 - This means layout and options could be different from other versions of Protégé

Introduction (VI): General View

The screenshot displays the Protégé software interface for the 'bibo' ontology. The 'Active Ontology' tab is selected, showing the ontology header with the IRI <http://purl.org/ontology/bibo/> and version <http://purl.org/ontology/bibo/1.0.0>. The 'Annotations' panel lists several annotations, including 'creator' with values 'f. glasson' and 'bdarcus', and 'title' with the value 'The Bibliographic Ontology'. The 'Ontology metrics' panel shows various counts for axioms, classes, and properties. The 'Imported ontologies' panel shows a list of imported ontologies.

The main Protégé workspace opens by default to the **Active Ontology** tab. This tab shows an overview of the current ontology (annotations about the ontology as a whole).

This part includes metrics on the ontology contents.

This part informs about other imported ontologies (if any imports exist).

Metric	Count
Axiom	65
Logical axiom count	53
Class count	55
Object property count	14
Data property count	ALUHN(D)
Individual count	
DL expressivity	
SubClassOf axioms count	76
EquivalentClasses axioms count	3
DisjointClasses axioms count	0
Other axioms	0

Ontology imports: OntoGraf Import View | Ontology Prefixes | General class axioms

Imported ontologies:

Direct Imports: +

Indirect Imports:

Imported ontologies:

Direct Imports: +

Indirect Imports:

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

Introduction (VII): General View

The screenshot displays the Protege OWL editor interface for the 'bibo' ontology. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The main toolbar contains navigation and editing icons. The 'Entities' tab is active, showing a class hierarchy on the left and a detailed view of the 'Article' class on the right.

Class hierarchy (left pane):

- Document = Document
 - Article
 - AudioDocument
 - AudioVisualDocument
 - Book
 - CollectedDocument
 - DocumentPart
 - Image = Image
 - LegalDocument
 - Manual
 - Manuscript
 - Note
 - Patent
 - PersonalCommunicationDocument
 - ReferenceSource

Object property hierarchy (bottom left pane):

- editor
 - citedBy
 - cites
 - contributor
 - director
 - editor
 - interviewee
 - interviewer
 - performer
 - translator

Article class details (right pane):

- Usage:** Shows: ☒ this. No usage.
- Class Annotations:**
 - label** [language: en]: Article
 - comment** [language: en]: A written composition in prose, usually nonfiction, on a specific topic, forming an independent part of a book or other publication, as a newspaper or magazine.
- Description:**
 - Equivalent To:** +
 - Sub Class Of:** Document
 - Sub Class Of (Anonymous Ancestor):** Document
 - Members:** +
 - Target for Key:** +

Modelling Components (I)

- **Classes** are sets that contain individuals
 - *Examples:* Table, Chair, Person, Building, etc.
- **Thing** is a class representing the set containing all individuals
 - All classes are subclasses of Thing
- Classes are assumed to ‘overlap’
 - Individuals of a class A can also be individuals of class B
 - To ‘separate’ a group of classes
 - One must make them **disjoint** from one another
 - If A is disjoint from B, then an individual of class A cannot also be an individual of class B
 - *Example:* Man is disjoint with Woman; Even is disjoint with Odd

Modelling Components (I): Classes Tab

The screenshot displays the BIBO ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The main toolbar contains icons for navigation and editing. The 'Classes' tab is active, showing the Class Hierarchy tree on the left and the Class Description panel on the right. The Class Hierarchy tree lists various classes under the 'Thing' root, including Agent, Collection, Document, Article, AudioDocument, AudioVisualDocument, Book, CollectedDocument, DocumentPart, Image, LegalDocument, Manual, Manuscript, Note, Patent, PersonalCommunicationDocument, ReferenceSource, Report, Slideshow, Standard, Thesis, Webpage, DocumentStatus, Event, Image, List, Organization, Person, Resource, Seq, and ThesisDegree. The Class Description panel shows the 'Document' class with its label, comment, isDefinedBy URI, term status, and sub-classes. The Ontology Metrics table provides a summary of the ontology's structure.

Class Hierarchy

- Thing
 - Agent = Agent
 - Agent = Agent
 - Collection
 - Document = Document
 - Article
 - AudioDocument
 - AudioVisualDocument
 - Book
 - CollectedDocument
 - DocumentPart
 - Image = Image
 - LegalDocument
 - Manual
 - Manuscript
 - Note
 - Patent
 - PersonalCommunicationDocument
 - ReferenceSource
 - Report
 - Slideshow
 - Standard
 - Thesis
 - Webpage
 - Document = Document
 - DocumentStatus
 - Event
 - Event
 - Image = Image
 - List
 - Organization
 - Person
 - Resource
 - Seq
 - ThesisDegree

Class Description

Annotations: Document

label [language: en]
Document

comment [language: en]
A document (noun) is a bounded physical representation of body of information designed with the capacity (and usually intent) to communicate. A document may manifest symbolic, diagrammatic or sensory-representational information.

isDefinedBy [type: anyURI]
<http://purl.org/ontology/bibo/>

term_status
stable

Ontology metrics:

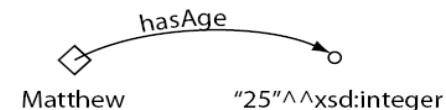
Metrics	
Axiom	978
Logical axiom count	299
Class count	69
Object property count	53
Data property count	55
Individual count	14

Modelling Components (II)

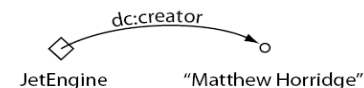
- **Properties** are binary relations (represent relationships between two objects)
 - between two individuals (**Object Property**)
 - Correspond to relationships in UML
 - *Examples:*
 - Man is subclass of Person
 - Person works in Building
 - between one individual and a data values (**Datatype Property**)
 - Correspond to attributes in UML
 - *Examples:*
 - Person has age
 - Book has title
 - **Annotation properties:** to be used to add annotation information to the entire ontology (model), classes, individuals, and properties



An object property linking the individual Matthew to the individual Gemma



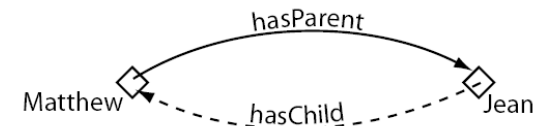
A datatype property linking the individual Matthew to the data literal '25', which has a type of an xml:integer.



An annotation property, linking the class 'JetEngine' to the data literal (string) "Matthew Horridge".

Modelling Components (II)

- **Properties** are binary relations (represent relationships between two objects)
 - Domain and Range
 - Properties link individuals from the **domain** to individuals or datatypes from the **range**
 - Super Properties
 - Properties can be further refined as **sub-properties** inheriting the domain, range, characteristics and restrictions
 - **Inverted properties in object properties**
 - If some property links individual A to individual B then its inverse property will link individual B to individual A
 - *Example*: hasIngredient versus isIngredientOf



Modelling Components (II): Object Properties Tab

The screenshot displays the BiBO ontology editor interface. The main window is titled "bibo (http://purl.org/ontology/bibo/)" and shows the "Object Properties" tab selected. The interface is divided into several panes:

- Object Property Hierarchy:** A tree view on the left showing the hierarchy of object properties. The root is "topObjectProperty", which branches into "agent", "based_near", "citedBy", "contributor", "contributorList", "court", "degree", "depiction", "distributor", "format", "hasPart", "homepage", "isPartOf", "isReferencedBy", "isVersionOf", "language", "organizer", "owner", "place", "produced_in", "producer", "product", "publisher", "recipient", "references", "relation", "rights", "status", "sub_event", "subject", "time", "title", and "value".
- Annotations editor:** A pane showing the annotations for the selected object property. It includes fields for "label", "editor", "comment" (with a language dropdown set to "en"), and "isDefinedBy" (with a type dropdown set to "anyURI"). The comment text is: "A person having managerial and sometimes policy-making responsibility for the editorial part of a publishing firm or of a newspaper, magazine, or other publication." The "isDefinedBy" field contains the URI "http://purl.org/ontology/bibo/".
- Object Property Characteristics:** A pane showing the characteristics of the selected object property. It includes checkboxes for "Functional", "Inverse functional", "Transitive", "Symmetric", "Asymmetric", "Reflexive", and "Irreflexive".
- Object Property Description:** A pane showing the description of the selected object property. It includes fields for "Equivalent To", "SubProperty Of", "Inverse Of", "Domains (intersection)", "Ranges (intersection)", "Disjoint With", and "SuperProperty Of (Chain)". The "SubProperty Of" field is set to "contributor". The "Domains (intersection)" field is set to "Collection or Document". The "Ranges (intersection)" field is set to "Agent".
- Ontology metrics:** A pane showing various metrics for the ontology, including "Axiom", "Logical axiom count", "Class count", "Object property count", "Data property count", "Individual count", and "DL expressivity".
- Class axioms:** A pane showing the count of axioms for various class-related properties, including "SubClassOf axioms count", "EquivalentClasses axioms count", "DisjointClasses axioms count", "GCI count", and "Hidden GCI Count".
- Object property axioms:** A pane showing the count of axioms for various object property-related properties, including "SubObjectPropertyOf axioms count", "EquivalentObjectProperties axioms count", "InverseObjectProperties axioms count", "DisjointObjectProperties axioms count", "FunctionalObjectProperty axioms count", "InverseFunctionalObjectProperty axioms count", "TransitiveObjectProperty axioms count", "SymmetricObjectProperty axioms count", "AsymmetricObjectProperty axioms count", "ReflexiveObjectProperty axioms count", and "IrreflexiveObjectProperty axioms count".

Modelling Components (II): Data Properties Tab

The screenshot displays the BIBO ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The main toolbar contains icons for navigation and editing. The 'Data Properties' tab is active, showing a tree view of the ontology hierarchy on the left and a list of annotations on the right.

Datatype Property Hierarchy

- topDataProperty
 - abstract
 - argued
 - date
 - created
 - issued
 - description
 - doi = doi
 - edition = edition
 - edition = edition
 - eissn = eissn
 - endingPage = pageEnd
 - family_name
 - givenname
 - identifier
 - isbn = isbn
 - issn = issn
 - issue = issue
 - localityName
 - locator = number
 - chapter
 - issue = issue
 - pageEnd = endingPage
 - pages
 - pageStart = startingPage
 - section
 - volume = volume
 - name
 - number = locator
 - number
 - numPages
 - numVolumes
 - prefixName
 - shortDescription
 - shortTitle
 - startingPage = pageStart
 - suffixName
 - volume = volume
 - content

Information about datatype properties annotations and usages

Annotations: abstract

- label
- abstract
- comment
- A summary of the resource.
- isDefinedBy [type: anyURI]
<http://purl.org/dc/terms/>
- term_status
- stable

Datatype Property Characteristics

Characteristics: abstract

☐ Functional

Datatype Property Description

Description: abstract

- Equivalent To
- SubProperty Of
- Domains (intersection)
 - Resource
- Ranges
 - Literall
- Disjoint With

Modelling Components (III)

- **Individuals** represent objects within the model (members of classes)
 - *Examples:* María, Torre Espacio, UPM, etc.

Modelling Components (III): Individuals Tab

The screenshot displays the bibo ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The main toolbar contains icons for navigating between different views: Class hierarchy, Members list, Annotations, and Usage. The 'Active Ontology' is set to 'bibo (http://purl.org/ontology/bibo/)'. The 'Individuals' tab is selected, showing a list of instances for the 'nonPeerReviewed' class. The 'Annotations' tab is also visible, showing various annotations for the 'nonPeerReviewed' class, including 'label', 'comment', 'term_status', and 'stable'. The 'Instances Description' tab is also visible, showing the types of instances (DocumentStatus, Thing) and the same/different individuals.

Class Hierarchy

- Thing
 - Agent = Agent
 - Agent = Agent
 - Collection
 - Document = Document
 - Document = Document
 - DocumentStatus
 - Event
 - Event
 - Image = Image
 - List
 - Organization
 - Person
 - Resource
 - Seq
 - ThesisDegree

List of Instances

- accepted
- draft
- forthcoming
- legal
- nonPeerReviewed
- peerReviewed
- published
- rejected
- unpublished

Information about individuals annotations and usages

Annotations: nonPeerReviewed

- label [language: en]
non peer reviewed
- comment [language: en]
A document that is not peer reviewed
- term_status
stable

Instances Description

Description: nonPeerReviewed

Types

- DocumentStatus
- Thing

Same Individual As

Different Individuals

Property assertions: nonPeerReviewed

- Object property assertions
- Data property assertions
- Negative object property assertions
- Negative data property assertions

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

Example: DC in Protégé

The screenshot displays the Protégé ontology editor interface, showing the DC (Dublin Core) ontology. The browser address bar at the top indicates the URL <http://purl.org/dc/terms/>. The menu bar includes File, Edit, Ontologies, Reasoner, Tools, Refactor, Tabs, View, Window, and Help. The toolbar contains icons for navigation and editing.

The main workspace is divided into several panes:

- Active Ontology:** Shows the 'BibliographicResource' class hierarchy. The hierarchy includes: Thing, VocabularyEncodingScheme, Collection, Agent, BibliographicResource (selected), Frequency, LinguisticSystem, LocationPeriodOrJurisdiction, MediaTypeOrExtent, MethodOfAccrual, MethodOfInstruction, PhysicalResource, and Policy.
- Class Annotations:** Shows the 'BibliographicResource' class with the following annotations:
 - comment:** "A book, article, or other documentary resource."@en
 - isDefinedBy:** "http://purl.org/dc/terms/"^anyURI
 - label:** "Bibliographic Resource"@en
- Object Properties:** Shows the 'language' property hierarchy. The hierarchy includes: language (selected), publisher, relation, rights, source, subject, type, accrualMethod, accrualPeriodicity, accrualPolicy, and audience.

Index

- Brief Introduction
- Main Modelling Components
- **How to use it**
 - Importing and Extending Ontologies
 - Including Equivalences
 - Importing Ontology Design Patterns
 - Referencing Elements from other Ontologies

Importing Ontologies (I)

documentation (<http://www.semanticweb.org/ontologies/2014/10/documentation>) : [<http://www.semanticweb.org/ontologies/2014/10/untitled-ontology-20>]

File Edit View Reasoner Tools Refactor Window Help

documentaion (<http://www.semanticweb.org/ontologies/2014/10/documentation>) Search for entity

Active Ontology Entities Classes Object Properties

Ontology header:

Ontology IRI <http://www.semanticweb.org/ontologies/2014/10/documentation>

Ontology Version IRI e.g. <http://www.semanticweb.org/ontologies/2014/10/documentation#>

Annotations +

Ontology metrics:

Metrics

Axiom
Logical axiom count
Class count
Object property count
Data property count
Individual count
DL expressivity

Class axioms

Ontology imports OntoGraf Import View Ontology Prefixes

Imported ontologies:

Direct Imports +

Indirect Imports

Imported ontologies:

Direct Imports +

Importing Ontologies

Import type

Please choose an option:

- ☒ Import an ontology contained in a specific file.
- ☐ Import an ontology contained in a document located on the web.
- ☐ Import an ontology that is already loaded in the workspace.
- ☐ Import an ontology that is contained in one of the ontology libraries.

Go Back Continue Cancel

Importing Ontologies (II)

documentation (<http://www.semanticweb.org/ontologies/2014/10/documentation>) : [<http://www.semanticweb.org/ontologies/2014/10/untitled-ontology-20>]

File Edit View Reasoner Tools Refactor Window Help

documentation (<http://www.semanticweb.org/ontologies/2014/10/documentation>) Search for entity

Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OWLViz DL Query OntoGraf

Ontology header:

Ontology IRI <http://www.semanticweb.org/ontologies/2014/10/documentation>

Ontology Version IRI e.g. <http://www.semanticweb.org/ontologies/2014/10/documentation/1.0.0>

Annotations +

Ontology metrics:

Metrics

Axiom	975
Logical axiom count	297
Class count	69
Object property count	52
Data property count	54
Individual count	14
DL expressivity	ALUHN(D)

Class axioms

Ontology imports OntoGraf Import View Ontology Prefixes General class axioms

Imported ontologies:

Direct Imports +

bibo (<http://purl.org/ontology/bibo/>)

List of Imported Ontologies

Imported ontologies:

Direct Imports +

Importing Ontologies (III): Overview of bibo ontology

documentation (<http://www.semanticweb.org/ontologies/2014/10/documentation>) : [<http://www.semanticweb.org/ontologies/2014/10/untitled-ontology-20>]

File Edit View Reasoner Tools Refactor Window Help

documentation (<http://www.semanticweb.org/ontologies/2014/10/documentation>) template

Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OWLViz DL Query OntoGraf

Class hierarchy Class hierarchy (inferred)

Class hierarchy: Document

- Document = Document
 - Article
 - AudioDocument
 - AudioVisualDocument
 - Book
 - CollectedDocument
 - DocumentPart
 - Image = Image
 - LegalDocument
 - Manual
 - Manuscript
 - Note
 - Patent
 - PersonalCommunicationDocument

Usage:

Show: ☒ this

No usage

Class Annotations Class Usage

Annotations: Document

label [language: en]
Document

comment [language: en]
A document (noun) is a bounded physical representation of body of information designed with the capacity (and usually intent) to communicate. A document may manifest symbolic, diagrammatic or sensory-representational information.

Description: Document

Equivalent To +
Document

SubClass Of +
Document

SubClass Of (Anonymous Ancestor)
Document

Members +

Target for Key +

Individuals by type Annotation property hierarchy Datatypes

Object property hierarchy Data property hierarchy

Object property hierarchy: degree

- citedBy
- cites
- contributor
 - director
 - editor
 - interviewee
 - interviewer
 - performer

Extending Ontologies (I): Specializing the Document taxonomy

The screenshot displays the Protégé ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The address bar shows the URL <http://www.semanticweb.org/ontologies/2014/10/documentation>. The main toolbar contains icons for navigation and editing. The 'Class hierarchy: Template' panel on the left shows a tree of classes, with 'Template' highlighted in bold. A red box highlights the 'Template' class, and a blue dashed box highlights the 'New Class' button. A blue dashed box also highlights the 'Classes in our own ontology are highlighted in bold' text. The 'Annotations: Template' panel on the right shows the 'description' property with the text: 'A document having a preset format, used as a starting point for a particular document so that the format does not have to be recreated each time it is used.' The 'Description: Template' panel at the bottom shows the 'Document' class and its properties.

documentation (<http://www.semanticweb.org/ontologies/2014/10/documentation>) : [<http://www.semanticweb.org/ontologies/2014/10/untitled-ontology-20>]

File Edit View Reasoner Tools Refactor Window Help

documentation (<http://www.semanticweb.org/ontologies/2014/10/documentation>) template

Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OWL Viz DL Query OntoGraf

Class hierarchy: Template

Adding Classes

Class hierarchy (inferred)

- Agent ≡ Agent
- Collection
- Document ≡ Document
 - Article
 - AudioDocument
 - AudioVisualDocument
 - Book
 - CollectedDocument
 - DocumentPart
 - Image ≡ Image
 - LegalDocument
 - Manual
 - Manuscript
 - Note
 - Patent
 - PersonalComm
 - ReferenceSou
 - Report
 - Slideshow
 - Template**
 - Webpage
- Document ≡ Document
- Event
- Event
- Image ≡ Image

New Class

Classes in our own ontology are highlighted in bold

Annotations: Template

Annotations +

description

A document having a preset format, used as a starting point for a particular document so that the format does not have to be recreated each time it is used.

Description: Template

Equivalent To +

Class Of +

Document

Class Of (Anonymous Ancestor)

Document

Document

Members +

Target for Key +

Disjoint With +

<http://www.semanticweb.org/ontologies/2014/10/documentation#Template>

Extending Ontologies (II): Including a new Object Property

The screenshot displays the Protégé ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The address bar shows the current ontology URL: <http://www.semanticweb.org/ontologies/2014/10/documentation>. The main workspace is divided into several panes:

- Class hierarchy:** A tree view showing the hierarchy of classes. The 'Template' class is highlighted.
- Object property hierarchy:** A tree view showing the hierarchy of object properties. The 'isBasedOn' property is highlighted.
- Annotations:** A pane for adding annotations to the selected property.
- Object Property Usage:** A pane for adding usage information to the selected property.
- Characteristics:** A pane for adding characteristics to the selected property.
- Description:** A pane for adding a description to the selected property.

Annotations and Object Property Usage panes are currently empty. The Characteristics pane shows a list of checkboxes: Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, and Irreflexive. The Description pane shows a list of checkboxes: Equivalent To, SubProperty Of, Inverse Of, Domains (intersection), Ranges (intersection), Disjoint With, and SuperProperty Of (Chain). The Domains (intersection) and Ranges (intersection) checkboxes are checked, and the domains and ranges are set to 'Document' and 'Template' respectively.

Annotations: isBasedOn

Annotations +

Characteristics: isBasedOn

Functional
Inverse functional
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive

Description: isBasedOn

Equivalent To +
SubProperty Of +
Inverse Of +
Domains (intersection) +
Ranges (intersection) +
Disjoint With +
SuperProperty Of (Chain) +

Document
Template

Adding Object Properties

New Object Property

Domain and Range

Including Equivalences

The screenshot displays the Protégé ontology editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The address bar shows the current ontology: [documentation \(http://www.semanticweb.org/ontologies/2014/10/documentation\)](http://www.semanticweb.org/ontologies/2014/10/documentation) : [http://www.semanticweb.org/ontologies/2014/10/untitled-ontology-20]. The toolbar contains icons for navigation and editing. The main workspace is divided into several panes:

- Class hierarchy (inferred):** A tree view on the left showing the ontology structure. The 'Movie' class is highlighted with a red box. The URL <http://www.semanticweb.org/ontologies/2014/10/documentation#Movie> is visible next to it.
- Annotations:** A pane on the right showing annotations for the 'Movie' class. It includes a 'comment' (An informal word for film) and a 'description' (A sequence of photographs projected onto a screen with sufficient rapidity as to create the illusion of motion and continuity).
- Description: Movie:** A pane on the right showing the class description. It includes an 'Equivalent To' relation with 'Film', which is highlighted with a red box. A blue dashed box with the text 'Adding Equivalent Relation' is overlaid on this section. Other relations shown include 'SubClass Of' with 'AudioVisualDocument' and 'SubClass Of (Anonymous Ancestor)' with 'Movie', 'Document', and 'Document'.

The bottom status bar indicates: To use the reasoner click Reasoner->Start reasoner ☒ Show Inferences

Importing Ontology Design Patterns (I)

The screenshot shows the Protégé ontology editor interface. The title bar indicates the current ontology is 'patternsExample' with the IRI 'http://www.semanticweb.org/ontologies/2014/10/patternsExample'. The main menu includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The 'Active Ontology' tab is selected, showing the 'Ontology header' section with the 'Ontology IRI' and 'Ontology Version IRI'. Below this is the 'Ontology metrics' section, which displays a table of metrics for the current ontology. At the bottom, the 'Imported ontologies' section is highlighted with a red box, showing a list of direct imports. A blue dashed box with the text 'Importing Ontology Design Patterns' is overlaid on the 'Imported ontologies' section.

patternsExample (http://www.semanticweb.org/ontologies/2014/10/patternsExample) : [http://www.semanticweb.org/ontologies/2014/10/untitled-ontology-21]

File Edit View Reasoner Tools Refactor Window Help

patternsExample (http://www.semanticweb.org/ontologies/2014/10/patternsExample) template

Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OWLViz DL Query OntoGraf

Ontology header:

Ontology IRI http://www.semanticweb.org/ontologies/2014/10/patternsExample

Ontology Version IRI e.g. http://www.semanticweb.org/ontologies/2014/10/patternsExample/1.0.0

Annotations +

Ontology metrics:

Metrics

Axiom	117
Logical axiom count	15
Class count	5
Object property count	4
Data property count	0
Individual count	0
DL expressivity	ALCHI

Imported ontologies:

Direct Imports +

agentrole (http://www.ontologydesignpatterns.org/cp/owl/agentrole.owl)

Importing Ontology Design Patterns

Importing Ontology Design Patterns (II): Overview of the AgentRole Pattern

patternsExample (<http://www.semanticweb.org/ontologies/2014/10/patternsExample>) : [<http://www.semanticweb.org/ontologies/2014/10/untitled-ontology-21>]

File Edit View Reasoner Tools Refactor Window Help

patternsExample (<http://www.semanticweb.org/ontologies/2014/10/patternsExample>) template

Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OWL Viz DL Query OntoGraf

Class hierarchy Class hierarchy (inferred)

Class hierarchy: Role

- Thing
 - Concept
 - Role
 - Object
 - Agent

Usage:

Show: ☒ this
No usage

Class Annotations Class Usage

Annotations: Role

`label` [language: it]
Ruolo

`comment` [type: string]
A Concept that classifies an Object

`isDefinedBy`
<http://www.ontologydesignpatterns.org/cp/owl/objectrole.owl>

Description: Role

Equivalent To +

SubClass Of +
● Concept

SubClass Of (Anonymous Ancestor)

Members +

Target for Key +

Object property hierarchy: topObjectProperty

- classifies
- isRoleOf
- isClassifiedBy
- hasRole

Diagram illustrating the AgentRole pattern structure:

```
graph TD; Agent --|> Object; Object -- hasRole --> Role; Role -- isRoleOf --> Object;
```

The diagram shows a class hierarchy where **Agent** is a subclass of **Object**. The **Object** and **Role** classes are connected by two properties: **hasRole** (from **Object** to **Role**) and **isRoleOf** (from **Role** to **Object**).

Instanciating Ontology Design Patterns

patternsExample (http://www.semanticweb.org/ontologies/2014/10/patternsExample) : [http://www.semanticweb.org/ontologies/2014/10/untitled-ontology-21]

File Edit View Reasoner Tools Refactor Window Help

patternsExample (http://www.semanticweb.org/ontologies/2014/10/patternsExample)

Search for entity

Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OWLViz DL Query OntoGraf

Class hierarchy Class hierarchy (inferred)

Class hierarchy: Agent

Members list Members list (inferred)

Members list: Aldo_Gangemi

Annotations Usage

Annotations: Aldo_Gangemi

Annotations

Thing

- Concept
- Role
- Object
- Agent

Object

Role

Agent

isRoleOf

hasRole

instanceOf

saxophonist

seniorResearcher

father

AldoGangemi

Instance of Agent

Relationships with Instances of Role

Description: Aldo_Gangemi

Types

Property assertions: Aldo_Gangemi

Object property assertions

- hasRole Saxophonist
- hasRole Father
- hasRole SeniorResearcher

Data property assertions

Negative object property assertions

Negative data property assertions

```
graph TD
    Object -- isRoleOf --> Role
    Role -- isRoleOf --> Object
    Agent -- instanceOf --> Object
    AldoGangemi -- instanceOf --> Agent
    AldoGangemi -- hasRole --> saxophonist
    AldoGangemi -- hasRole --> seniorResearcher
    AldoGangemi -- hasRole --> father
    saxophonist -- instanceOf --> Role
    seniorResearcher -- instanceOf --> Role
    father -- instanceOf --> Role
```

Referencing Elements from other Ontologies (I)

The screenshot displays the Protégé ontology editor interface. The title bar shows the file path: `0.1 (http://xmlns.com/foaf/0.1/) : [C:\Trabajo\Docencia\Sistemas Inteligentes\Ontologies\foaf.rdf]`. The menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The 'Refactor' menu is highlighted with a red box. Below the menu bar, there are tabs for Active Ontology, Entities, Classes, Object Properties, Data Properties, Annotation Properties, Individuals, OWL Viz, DL Query, and OntoGraf. The main workspace shows the 'Ontology header' with fields for 'Ontology IRI' (http://xmlns.com/foaf/0.1/) and 'Ontology Version IRI' (e.g. http://xmlns.com/foaf/0.1/1.0.0). A 'description' field contains the text 'The Friend of a Friend (FOAF) RDF vocabulary'. The 'Ontology metrics' section lists various counts: Axiom, Logical axiom count, Class count, Object property count, Data property count, Individual count, and DL expressivity. A 'Copy/move/delete axioms' dialog box is open, titled 'Select method'. It contains the text 'Select the method of choosing axioms from the source ontology.' and four radio button options: 'Axioms by definition (select axioms that define entities)', 'Axioms by profile (select axioms in a sublanguage of OWL)', 'Axioms by reference (select entities from the ontology)', and 'Axioms by type (eg subclass axioms, annotation axioms)'. The 'Axioms by definition' option is selected. The dialog box has 'Go Back', 'Continue', and 'Cancel' buttons at the bottom. A large, semi-transparent 'protégé' watermark is visible in the background.

0.1 (http://xmlns.com/foaf/0.1/) : [C:\Trabajo\Docencia\Sistemas Inteligentes\Ontologies\foaf.rdf]

File Edit View Reasoner Tools **Refactor** Window Help

0.1 (http://xmlns.com/foaf/0.1/)

Search for entity

Active Ontology Entities Classes Object Properties Data Properties Annotation Properties Individuals OWL Viz DL Query OntoGraf

Ontology header:

Ontology IRI http://xmlns.com/foaf/0.1/

Ontology Version IRI e.g. http://xmlns.com/foaf/0.1/1.0.0

Annotations +

description

The Friend of a Friend (FOAF) RDF vocabulary

Ontology metrics:

Metrics

- Axiom
- Logical axiom count
- Class count
- Object property count
- Data property count
- Individual count
- DL expressivity

Ontology imports OntoGraf Import View Ontology Prefixes General class

Imported ontologies:

Direct Imports +

Indirect Imports

Imported ontologies:

Copy/move/delete axioms

Select method

Select the method of choosing axioms from the source ontology.

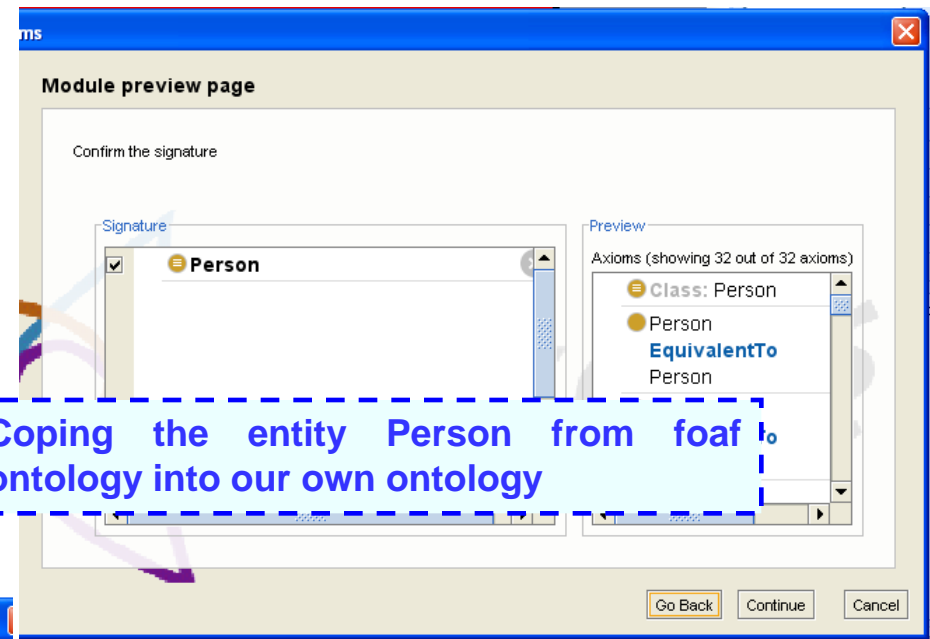
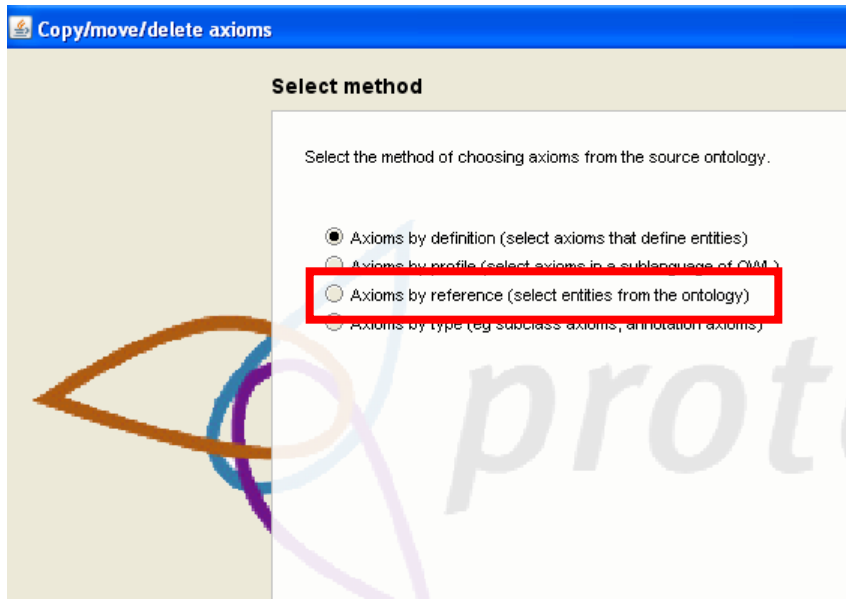
- ☒ Axioms by definition (select axioms that define entities)
- ☐ Axioms by profile (select axioms in a sublanguage of OWL)
- ☐ Axioms by reference (select entities from the ontology)
- ☐ Axioms by type (eg subclass axioms, annotation axioms)

Go Back Continue Cancel

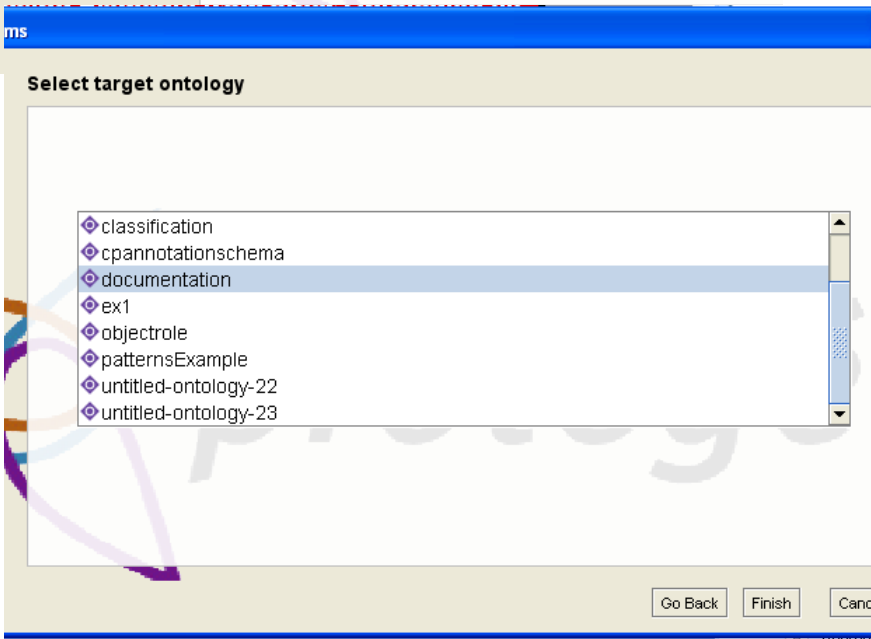
protégé

Coping axioms from other Ontologies

Referencing Elements from other Ontologies (II)



Coping the entity Person from foaf ontology into our own ontology



The same process for coping the entity Thesis from bibo ontology into our own ontology

Referencing Elements from other Ontologies (III)

The screenshot displays the Protégé ontology editor for 'untitled-ontology-23'. The interface includes a menu bar (File, Edit, View, Reasoner, Tools, Refactor, Window, Help), a toolbar, and a search bar. The main workspace is divided into several panes:

- Class hierarchy:** Shows a tree structure starting with 'Thing', followed by 'Agent', 'Document', 'Thesis', 'Organization', 'Project', and 'SpatialThing'. The 'Person' class is highlighted with a red box.
- Object property hierarchy:** Shows the 'isAuthorOf' property being edited. A red box highlights the 'Person' class in the 'Domains (intersection)' section.
- Object property usage:** Shows the 'isAuthorOf' property being used in the 'Ranges (intersection)' section.
- Characteristics:** A list of checkboxes for property characteristics: Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, and Irreflexive.

A blue dashed box highlights the 'isAuthorOf' property in the 'Object property hierarchy' pane, with the text 'Including an object property in our own ontology' overlaid. The URL 'http://www.semanticweb.org/ontologies/2014/10/untitled-ontology-23#isAuthorOf' is visible at the bottom.

Course: Intelligent Systems

Unit 3: Ontology Engineering

Using Protégé (Desktop)

Mari Carmen Suárez de Figueroa Baonza

Technical University of Madrid

