

Ontology Editors

IDEs for Ontologies

- **Some people use simple text editors**
 - Doing this with the XML serialization will drive you crazy
 - Using Turtle or an abstract syntax works well
- **Others prefer an IDE**
 - Good IDEs include support for reasoning, visualization, and more
- **Protégé is very a very popular IDE**
 - From Stanford, free, lots of plugins
- **TopQuadrant Composer is also good**
 - Feature rich but expensive (\$600 for a single academic license)

Protégé 4.3

The screenshot displays the Protégé 4.3 web interface for the 'pizza' ontology. The browser address bar shows the URL: `http://www.co-ode.org/ontologies/pizza/pizza.owl`. The interface includes a search bar for entities and a navigation menu with tabs for 'Active Ontology', 'Entities', 'Classes', 'Object Properties', 'Data Properties', 'Individuals', 'OWLviz', 'DL Query', and 'OntoGraf'. The 'Annotations' tab is selected, showing the ontology header with the IRI and version IRI, and a list of annotations including 'versionInfo' and 'comment'. The 'Ontology metrics' panel on the right provides a summary of the ontology's statistics, including the number of axioms, classes, and object properties.

Active Ontology | Entities | Classes | Object Properties | Data Properties | Individuals | OWLviz | DL Query | OntoGraf

Annotations | Selected entailments

Ontology header:

Ontology IRI: `http://www.co-ode.org/ontologies/pizza/pizza.owl`

Ontology Version IRI: e.g. `http://www.co-ode.org/ontologies/pizza/pizza.owl/1.0.0`

Annotations +

- versionInfo** [type: string]
version 1.5
- comment** [language: en]
An example ontology that contains all constructs required for the various versions of the Pizza Tutorial run by Manchester University (see <http://www.co-ode.org/resources/tutorials/>)
- versionInfo** [language: en]
v.1.5. Removed protege.owl import and references. Made ontology URI date-independent

Ontology imports | General axioms | RDF/XML rendering | OWL/XML rendering ▶

Imported ontologies:

- Direct Imports +
- Indirect Imports

Ontology metrics:

Metrics

Axiom	939
Logical axiom count	712
Class count	100
Object property count	8
Data property count	0
Individual count	5
DL expressivity	SHOIN

Class axioms

SubClassOf axioms count	259
EquivalentClasses axioms count	15
DisjointClasses axioms count	398
GCI count	0
Hidden GCI Count	2

Object property axioms

SubObjectPropertyOf axioms count	4
EquivalentObjectProperties axioms count	0
InverseObjectProperties axioms count	3
DisjointObjectProperties axioms count	0
FunctionalObjectProperty axioms count	4
InverseFunctionalObjectProperty axioms...	3
TransitiveObjectProperty axioms count	2
SymmetricObjectProperty axioms count	0
AsymmetricObjectProperty axioms count	0
ReflexiveObjectProperty axioms count	0

Reasoner active Show Inferences

Protégé 4.3

- <http://protege.stanford.edu/>
- Free, open source ontology editor and KB framework
- Predates OWL, still supports earlier Frames representation
- In Java, extensible, large community of users
- V4.3 supports OWL 2 but missing other features (e.g., SPARQL)
- v3.5 version has missing features, but only supports OWL 1

Web Protege

The screenshot shows the WebProtégé web interface in a browser window. The browser's address bar displays `webprotege.stanford.edu/#Edit:...`. The page header includes the Protégé logo and navigation buttons for Project, Share, Tim Finin, and Help. Below the header, there are tabs for 'Classes', 'Properties', 'Individuals', 'Notes and Discussions', 'Changes By Entity', and 'Project Dashboard'. The 'Classes' tab is active, showing a tree view of classes: 'owl:Thing', 'Person', and 'Sex'. The 'Person' class is selected, and its description is shown in the main area. The description includes a display name of 'Person' and an IRI of `http://webprotege.stanford.edu/RCcwwdIsdJMKB`. There are also sections for 'Annotations' and 'Properties', each with input fields for property names and languages.

WebProtégé **UMBC691PeepsExample**

Classes Properties Individuals Notes and Discussions Changes By Entity Project Dashboard

Add content to this tab Add tab

Classes

Create Delete Watch Br

- owl:Thing
- Person
- Sex

Class description for Person

Display name

IRI

Annotations

Properties

Discussions for Person

[Post new topic...](#)

YAS: Yet Another Syntax

- Neither OWL's official abstract syntax nor XML serialization is easy to read or use
- Protégé uses the Manchester syntax
- Simpler and more compact: “some” and “only”, not “someValuesFrom” and “allValuesFrom”
- A W3C recommendation (<http://bit.ly/manSyn>), used in the OWL 2 Primer (<http://bit.ly/OWL2Pri>)

Class: man

Annotations: rdfs:label "man"

EquivalentTo: adult and male and person

Manchester OWL syntax

OWL	DL Symbol	Manchester OWL Syntax Keyword	Example
someValuesFrom	\exists	some	hasChild some Man
allValuesFrom	\forall	only	hasSibling only Woman
hasValue	\ni	value	hasCountryOfOrigin value England
minCardinality	\geq	min	hasChild min 3
cardinality	$=$	exactly	hasChild exactly 3
maxCardinality	\leq	max	hasChild max 3

Manchester OWL syntax

OWL	DL Symbol	Manchester OWL Syntax Keyword	Example
intersectionOf	\sqcap	and	Doctor and Female
unionOf	\sqcup	or	Man or Woman
complementOf	\neg	not	not Child

Example

```
Person and  
  hasChild some  
    (Person and  
      (hasChild only Man) and  
        (hasChild some Person) )
```

The set of people who have at least one child that has some children that are only men (i.e., grandparents that only have grandsons)

Data values and datatypes

- Data values typed or untyped (e.g., int, boolean, float)
- Constants with or w/o type, e.g.: hasAge value "21"^^long
- Use datatype names as classes: hasAge some int
- XSD facets, e.g.: Person and hasAge some int[>= 65]
- Ranges: Person and hasAge some int[>= 18, <= 30].

XSD facet	Meaning
< x, <= x	less than, less than or equal to x (more info)
> x, >= x	greater than, greater than or equal to x (more info)
length x	For strings, the number of characters must be equal to x (more info)
maxLength x	For strings, the number of characters must be less than or equal to x (more info)
minLength x	For strings, the number of characters must be greater than or equal to x (more info)
pattern regexp	The lexical representation of the value must match the regular expression, regexp (more info)
totalDigits x	Number can be expressed in x characters (more info)
fractionDigits x	Part of the number to the right of the decimal place can be expressed in x characters (more info)

Demonstration

- We'll use Protégé OWL v4.3 to implement a tiny ontology for people
- Start by downloading and installing Protégé 4.3
 - You will need Java
- You may want to install Graphviz
- Configure Protégé
 - E.g., select a reasoner to use (e.g., HermiT or Pellet)

Protégé OWL v 4.3

The screenshot displays the Protégé OWL v 4.3 web interface. At the top, the browser address bar shows the URL `http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-40`. Below the address bar is a search field labeled "Search for entity".

The main interface is divided into several sections:

- Active Ontology:** A horizontal menu with tabs for "Entities", "Classes", "Object Properties", "Data Properties", "Annotation Properties", and "Individuals".
- Annotations:** A sub-menu with tabs for "Annotations", "Selected entailments", and "Rules".
- Ontology header:** A purple header bar with a title "Ontology header:" and a close button. Below it are two input fields:
 - Ontology IRI:** `http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-40`
 - Ontology Version IRI:** `e.g. http://www.semanticweb.org/ontologies/2014/3/untitled-ontology-40/1.0.0`
- Annotations:** A large empty area with a "+" icon and the label "Annotations".
- Imported ontologies:** A sub-menu with tabs for "Ontology imports", "General axioms", "RDF/XML rendering", "OWL/XML rendering", and "OWL functional syntax rendering". Below it is a purple header bar with a title "Imported ontologies:" and a close button. Underneath are two sections:
 - Direct Imports:** A section with a "+" icon.
 - Indirect Imports:** An empty section.

At the bottom of the interface, there is a footer with the text "To use the reasoner click Reasoner->Start reasoner" and a checked checkbox labeled "Show Inferences".

A basic workflow

- Think about usecases
- Preliminaries
 - Choose namespace URL, import other ontologies used
- Identify and define classes
 - Place in hierarchy, add axioms and run reasoner to check for errors or omissions
- Identify and define properties
 - Place in hierarchy, add axioms, run reasoner
- Add individuals & reasoner to check for problems
- Add comments and labels
- Export in desired formats, maybe upload to Web

More workflow steps

- Use [OOPS](#) to find common ontology pitfalls
- Link concepts (and individuals) to common ontologies (e.g., Dbpeia, foaf)
- Generate visualizations
- Produce documentation
- Develop examples with your use case(s)
- Encode data, describe in [VoID](#) (Vocabulary of Interlinked Datasets), add to LOD cloud

Demonstration

Use Protégé OWL (v4.3) to build a simple ontology for people based on the following

- People have just one sex that's either *male* or *female*, an integer age, and two parents, one male, one female
- A person's grandparent is the parent of their parent
- Every person is either a man or a woman but not both
- A man is defined as any person whose sex is male and a woman as any person whose sex is female
- A boy is defined as a person whose sex is male and whose age is less than 18, a girl is ...
- A person is either an adult or (age >18), minor (age <18)

Test cases

AllDifferent people

Alice F

Bob M

Carol F

Don M

Edith F

Pat ?

Other people

Frank M

Gwen F

Some possible test cases

- Alice parent Bob . Bob parent Carol
 - Alice grandparent Carol
- Alice parent Bob . Alice parent Don.
 - Contradiction
- Alice parent Bob . Pat parent Bob
 - Pat a female
- Alice parent Bob . Gwen parent Bob .
 - Alice owl:sameAs Gwen