Outline

Protegè:
- Protegé
- Protegé OWL
- Protegé plugins (ex.)

A Case study: Energy ontology
- Ontology Description
- The case of Hydrogen (Nixon’s Diamond)
Ontology Editor

Protegè is an ontology editor software developed at Stanford:
http://protege.stanford.edu/
Protegé (1)

Why Protegé?

• Protégé ontologies can be exported into a variety of formats including RDF(S), OWL, and XML Schema.

• It’s possible to work directly with the Protégé APIs.

• Collaborative Work (Protégé on a Web Server)

• Protégé is supported by a strong community of developers and academic, government and corporate users.
Protegé (2)

Why Protegé?
• In Protegé is possible to import ontologies
• Protegé OWL Plugins
• Two main ways of modeling ontologies:

Protégé Frames Editor: frame-based ontologies. Ontology: a set of classes (in a subsumption hierarchy), a set of slots to describe properties and relationships, and a set of instances of those classes.

Protégé OWL Editor: ontologies in Web Ontology Language (OWL). The OWL formal semantics specifies how to derive its logical consequences.

• Open Sources (last but not least!)
Protegé OWL Editor

OWL DL (it’s the third step) once you start Protegé
Protegé OWL Editor (2)
Create a Ontology Project in 3 steps

1

2

3
Protegé OWL Classes

Creating Classes with Protegé

- Select owl:Thing, go to “Tools” and click on “Create subclasses”;
Protegé OWL Properties

There are three types of properties:

- **Object properties**, link an individual to an individual;

  **Datatype properties**, link an individual to an XML Schema Datatype value or rdf literal

- **Annotation properties**, used to add information to classes, individuals and object/datatype properties.
Creating Properties with Protegé
Protegé OWL Plugins

Tab Widget

- OWLViz
- Instance Tree tab
- Jambalaya
- OntoVizTab
- String Search
- QueriesTab
- ...

Protegé OWL Plugin

Protegé Plugins allow to perform different tasks. For example:
They can be used to load and save OWL files in various formats (backends plugins), to edit OWL ontologies with graphical widgets (slot widgets and tab plugins), and to perform intelligent reasoning (Racer Pro).
Tab Widgets: OWLViz (1)

OWLViz enables the class hierarchies in an OWL Ontology to be viewed and incrementally navigated.

OWLViz integrates with the Protege-OWL plugin, using the same colour scheme so that primitive and defined classes can be distinguished.

Computed changes to the class hierarchy may be clearly seen, and inconsistent concepts are highlighted in red.
Tab Widgets: OWLViz (2)

OWLViz has the facility to save the class hierarchy to various concrete graphics formats including png, jpeg and svg.
Tab Widgets: Instance Tree Tab

The InstanceTree widgets allow navigation of frames that are directly and indirectly referenced from an instance in a tree structure.
Tab Widgets : Jambalaya (1)

Jambalaya is a plug-in created for Protégé to visualize the knowledge bases created by the users.

The Jambalaya Tab is composed of the Protégé class and instance panels as used in the rest of Protégé. The class and instance panels are synchronized with the Jambalaya Main View, so double-clicking a class will zoom to that class/instance and open it.
Tab Widgets: Jambalaya (2)
Tab Widgets: Jambalaya (3)
Tab Widgets: Jambalaya (4)
Tab Widgets: Jambalaya (5)
Tab Widgets: OntoViz Tab (1)

The OntoViz Tab allows you to visualize Protege ontologies with the help of a highly sophisticated graph visualization.
Tab Widgets : OntoViz Tab (2)
Tab Widgets: String search

The String Search tab allows the user to search through all values of type String in a knowledge base. It can be used to search for classes, slots and instances, as well as to search through slot values in the knowledge base.

- **Frame**: the Frame column displays the browser text of the frame
- **Direct Type**: Displays the direct type of the frame
- **Matched Slot**: Displays the slot that had a value matching the search string
- **Matched Value**: The value that matched the search
The queries tab allows you to query knowledgebases and export the results.
Tab Widgets: Queries Tab (2)

Select class
Select property
Select instance
Result

Add the query to Query Library
Add the query to Query Library
Tools: OWLDoc (1)

OWLDoc is a tool that generates JavaDoc style html page documentation for an OWL ontology. OWLDoc has been implemented so that it may be used with Protege-OWL in order to generate documentation for OWL ontologies.
Tools: OWLDoc (2)

Contents

- Ontology
  - All Resources
    - All Classes
      - All Object Properties
        - All Datatype Properties
          - All Annotation Properties
            - Individuals
              - * Resources
                - * Classes
            - * Resources

All Resources

- AlcoholFuel
- Algeria
- Argentina
- Barge
- Biofuel
- Biomass
- Biomass based
- Brazil
- CO2 Emissions
- Canada
- Changes in Climate
- China

Ontology

Namespaces

Default Namespace

http://www.owl-ontologies.com/Ontology1.195308540.ow#

xsd http://www.w3.org/2001/XMLSchema#

rdfs http://www.w3.org/2000/01/rdf-schema#

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

owl http://www.w3.org/2002/07/owl#

assert http://www.owl-ontologies.com/assert.owl#

Imports

http://www.owl-ontologies.com/assert.owl
Tools: Ontology Testing (1)

A mechanism to execute small test cases. Users can press an ontology test button, and the system will execute a configurable list of tests.
Tools: Ontology Testing (2)

These tests are small Java programs that basically take a class, property, individual, or ontology as its input, verify arbitrary conditions on them, and in case of failure, return an error message.

For example, symmetric properties must have equal ranges and domains. If a property in the ontology violates this conditions, then the system displays a warning. In some cases it even provides a “repair” button, which attempts to remove the source of the violation automatically.
OWL Plugin

Backends Widgets

1. Importing a DataBase

2. UML
Backends Widgets: Importing a Database

JDBC is a standard Java API for SQL database accesses.
The UML back-end plug-in provides an import and export mechanism between the Protege knowledge model and the object-oriented modeling language UML.

- Create an XMI file
- Choose “Export to format"
- Select UML as the import format
- Select the XMI file.
- Use new File exported with a tool support UML (Poseidon, Argo…) to create a class diagram
The OWL Plugin provides direct access to DL reasoners such as Racer. The current user interface supports two types of DL reasoning:

- **Consistency checking** (i.e., the test whether a class could have instances) can be invoked either for all classes with a single mouse click, or for selected classes only. Inconsistent classes are marked with a red bordered icon.
- **Classification** (i.e., inferring a new subsumption tree from the asserted definitions) can be invoked with the classify button on a one-shot basis.
Press the button Connect to connect the graphical interface to the RacerPro reasoning engine.
A Case study

- Energy ontology description
- Nixon’s Diamond
Energy Ontology Description (1)

Energy_Domain is the superclass, a sort of root node that contains all other subclasses.

On the same hierarchy level there are 2 classes also: Risks e Solutions. They have been included in this level because they are transversal across the ontology.

In this way was possible to create horizontal links between classes and instances into the hierarchy tree.
Energy Ontology Description (2)

Energy Ontology is more developed in **width** than in **depth**.

At first a basic structure of concepts has been identified. The taxonomy has been created starting by a database containing **200 documents**

At a later stage this database has been increased using a **spider**. The ontology has been tested with Racer Pro.
Energy Domain has 6 subclasses

- Country
- EnergySecurity
- EnergySources
- EnergyUse
- EnvironmentalConsequences
- Infrastructure
Energy Domain (Protegé Visualization)
Country Class

**Country**: has 2 subclasses that includes OPEC members and NON OPEC members. In each subclass there are **12 instances**

- All instances are connected by **properties** that allows to create many correspondences into hierarchy tree

- For instance **China** has 2 properties:
  - *Is One Of Major Exporter Of*
  - *Is One Of Major Producer Of*
Country Class (Ex. China)
Energy Security Class

- Energy_Security
  - Affordability
  - Reliability to supply
  - Friendliness Environment
Energy Security in Protegé
Energy Use Class

It includes, as subclasses, the most 5 important sector in which the energy domain standards divide the “energy use policies”

- Commercial
- Industrial
- Electric
- Residential
- Transportation
Energy Use Class in Protegé
Environmental Consequences Class
Energy Sources Class

**EnergySources** has 2 subclasses:

- **Primary** sources. A primary source of energy is one that already exists in nature and can be used directly, or converted or re-directed into a form of energy that satisfies our needs.

- **Secondary** sources. Secondary energy sources, such as electric power or refined fuels, do not exist in nature, but can be produced from the primary energy sources. Secondary sources are important because they are frequently easier to use than the primary sources from which they are derived.
Fossil Fuels Subclass (Ex. Oil)

FossilFuels is a Primary Energy Sources subclass, it’s in Non Renewable sources and has 11 instances.

- For instance, one of them is Oil. It has many properties to define very useful links with other instances or even with other classes of the ontology:
  - Is Mostly Produced By
  - Is Used To
An interesting case…

• Hydrogen: Renewable or not Renewable?

• Coke is one of 7 instances classified in Secondary and Non Renewable energy sources; Hydrogen is classified in the same level if it’s extracted by hydrocarbon

• But Hydrogen is classified in Renewable sources when it’s extracted by renewable elements (water, solar…)
Hydrogen: a Nixon’s Diamond

- Quacker are pacifist (usually)
- Republicans are not pacifist (usually)

- Nixon is a Quacker and a Republican

Image from Sowa (1992)
Ex. Hydrogen as Non Renewable Source
LOGICA DEI LINGUAGGI NATURALI E ARTIFICIALI

Antonio Lieto - University of Salerno
alieto@unisa.it
http://antoniolieto.blogspot.com