

How to install and use Natural OWL

Dimitris Galanis, George Karakatsiotis and Ion Androutsopoulos
Department of Informatics
Athens University of Economics and Business, Greece

March 2008

1. Introduction

This document accompanies NaturalOWL version 1.0.¹ NaturalOWL is an open-source natural language generation engine written in Java. It produces English and Greek descriptions of individuals (e.g., items for sale or museum exhibits) and classes (e.g., types of exhibits) from OWL DL ontologies. The ontologies must have been annotated in RDF with suitable linguistic and user modeling information. NaturalOWL was developed in the Department of Informatics of the Athens University of Economics and Business in the Greek project XENIOS; it is now being extended in the European project INDIGO.² NaturalOWL is heavily based on ideas from the European project M-PIRO, which was in turn based on Edinburgh's ILEX system, but it provides native support for OWL DL and it can be used within the Protégé ontology editor.³ It is also entirely template-based, unlike its predecessors that employed systemic grammars.⁴

NaturalOWL is presented briefly in the following article:

- D. Galanis and I. Androutsopoulos, “Generating Multilingual Descriptions from Linguistically Annotated OWL Ontologies: the NaturalOWL System”. Proceedings of the *11th European Workshop on Natural Language Generation* (ENLG 2007), Schloss Dagstuhl, Germany, pp. 143–146, 2007.

A much more detailed description of NaturalOWL can be found in the following article:

- I. Androutsopoulos and D. Galanis, “Towards Linguistically Annotated Semantic Web Ontologies: Experience from the NaturalOWL System”. Submitted for publication. Available from the authors upon request.

In the rest of this document, we assume that you have already read the two articles above. We also strongly recommend reading the following articles, which present M-PIRO and ILEX's ideas, on which NaturalOWL is based:

¹ NaturalOWL is available from <http://www.aueb.gr/users/ion/publications.html>, where most of the articles that are mentioned here can also be found.

² XENIOS was co-funded by the European Union and the Greek Secretariat of Research and Technology; see <http://www.ics.forth.gr/xenios/>. INDIGO is an FP6 IST project of the European Union; see <http://www.ics.forth.gr/indigo/>.

³ M-PIRO was an FP5 IST project of the European Union; see <http://www.ltg.ed.ac.uk/mpiro/>. ILEX was developed at the University of Edinburgh; consult <http://www.hcrc.ed.ac.uk/ilex/>. For information on Protégé, see <http://protege.stanford.edu/>.

⁴ Consult <http://www.ltg.ed.ac.uk/methodius/> for information on METHODIUS, another descendant of M-PIRO and ILEX, which uses CCG grammars and can handle larger ontologies and databases.

- I. Androutsopoulos, J. Oberlander and V. Karkaletsis, “Source Authoring for Multilingual Generation of Personalised Object Descriptions”. *Natural Language Engineering*, 13(3):191–233, Cambridge University Press, 2007.
- A. Isard, J. Oberlander, I. Androutsopoulos and C. Matheson, “Speaking the Users' Languages”. *IEEE Intelligent Systems*, 18(1):40–45, 2003.
- M. O’Donnell, C. Mellish and J. Oberlander, “ILEX: an Architecture for a Dynamic Hypertext Generation System”. *Natural Language Engineering*, 7(3):225–250, Cambridge University Press, 2001.

NaturalOWL is released with a GNU General Public License (GPL); please consult the file COPYING.txt for more information. Please note that NaturalOWL is a research prototype. It is provided with absolutely no guarantee and absolutely no support.⁵

2. How to install and use NaturalOWL in Protégé

To install NaturalOWL as a Protégé plug-in, follow the following steps:

Step 1: Installing a Java Virtual Machine

You need JDK 6 (or later) or JRE 6 (or later) installed on your computer to use NaturalOWL; they can be downloaded from <http://java.sun.com/>. To check which version of JDK or JRE is currently installed on your computer, type the following in a command-line shell:

```
java -version
```

If this command reports version number 1.6 or later, you can move on to step 1 without installing a new JDK or JRE. Otherwise, you have to install a newer JDK or JRE.

Step 2: Installing Protégé

NaturalOWL version 1.0 was developed and tested with Protégé version 3.3.1, but it may work with later Protégé versions too. The following installation instructions are for Protégé 3.3.1. They may have to be modified for later Protégé versions.

Download Protégé from: <http://protege.stanford.edu/>. Select “Download full Protégé”. *Do not* select an installer that includes a “Java VM”, because this may install an older version of the Java Virtual Machine (VM) than the version required by NaturalOWL.

After downloading Protégé, follow the installation instructions on the same page. When prompted to select components, choose “Everything”. You will be asked to specify an installation directory. If you use Windows Vista, *do not* select a sub-directory of *Program Files*, because this may cause unexpected behavior when experimenting with different versions of Protégé or NaturalOWL. Write down the installation directory you specified. When prompted to select a Java VM, click “Choose another” and use the file browsing window that will appear to specify the path to the Java VM to be used; in Windows, it should be something like:

```
C:\Program Files\Java\jdk1.6.0_01\bin\java.exe
```

⁵ However, please send bug reports to galanis@ueb.gr and gkarak05@ueb.gr.

Step 3: Installing NaturalOWL as a Protégé plug-in

Once the installation of Protégé has been completed, go to the installation directory of step 2 and find the “plugins” sub-directory. Create a new sub-directory inside “plugins” and name it “gr.aueb.cs.nlg.ProtegeAuthoringTool”. Copy both “NL.jar” and “plugin.properties” from NaturalOWL’s distribution to “gr.aueb.cs.nlg.ProtegeAuthoringTool”.

Step 4: Loading an .owl file in Protégé

Start Protégé (in Windows, double-click on “protege.exe” in the installation directory of step 2). Then:

- Close the “Welcome to Protégé” pop-up window (press “Cancel”) without selecting any of its options.
- From the “File” menu, select “New Project”.
- Tick “Create from Existing Sources” and select “OWL/RDF Files”; then press “Next”.
- Click on the browsing button (the one with the “+”) to locate the .owl file you want to load. NaturalOWL’s distribution includes a directory “NLFiles-MPIRO”, which contains a sample .owl file called “mpiro.owl”. This is the OWL version of an ontology of museum exhibits, originally created during the M-PIRO project.⁶ If you want to use that .owl file, *first copy it to a new empty directory*, and then select the copy that you placed in the new directory. Press “Next” to continue.
- When prompted, choose “OWL DL” and press “Next”.
- Select “Logic View” and press “Finish”.

Step 5: Creating a new NaturalOWL-aware Protégé project

Once you have loaded an .owl file, go to Protégé’s “Project” menu, select “Configure” and add (tick) “LexiconTab”, “MicroplansAndOrderingTab”, “UserModellingTab”, and “TextPreviewsTab” to the visible tabs. Then press “OK”. You can now use the four tabs to add linguistic and user modeling annotations (in RDF) to the OWL ontology (.owl file) that you have loaded, as explained in the articles that present NaturalOWL.

To save the (possibly modified) OWL ontology and the RDF annotations that you have created, go to Protégé’s “File” menu and select “Save project”. This will save the .owl file; it will also create a .pprj file (Protégé project file) and the files “Lexicon.rdf”, “microplans.rdf” and “UserModelling.rdf” in the same directory. You can now re-start Protégé at a later time (after exiting) and re-load both the OWL ontology that you were editing and its RDF annotations simply by double-clicking on the .pprj file; this will also re-activate NaturalOWL’s four Protégé tabs.

If you want to use an OWL ontology (.owl file) that is distributed with accompanying RDF annotations for NaturalOWL (“Lexicon.rdf”, “microplans.rdf”, “UserModelling.rdf”), *quit Protégé after creating the .pprj file* from a copy of the .owl file as described in the previous paragraphs; then *copy the three .rdf files that accompany the original .owl file into the directory where you placed the copy of the .owl file, overwriting the .rdf files that will have already been created there*. You can try this process with the three .rdf files that accompany the “mpiro.owl” file in NaturalOWL’s distribution; all four files are in the directory “NLFiles-MPIRO”. *Remember to copy “mpiro.owl” from “NLFiles-MPIRO” to a new empty directory, create the .pprj file in the new directory, quit Protégé, and then overwrite the three*

⁶ Unfortunately, we cannot provide the images of the exhibits, due to copyright restrictions. You can get an idea of what the images look like by reading the article “Source Authoring for Multilingual Generation of Personalised Object Descriptions”, which was mentioned in the introduction.

.rdf files that will have been created in the new directory with the .rdf files of “NLFiles-MPIRO”.

Step 6: Generating texts with NaturalOWL in Protégé

To generate texts from an OWL ontology, you need to load a NaturalOWL-aware Protégé project, which will have been created as in the previous step. Then:

- Go to the “Text Previews” tab.
- Select a class and/or an individual (instance of a class) of the ontology by using Protégé’s Class Browser and Instance Browser. For example, if you have loaded M-PIRO’s ontology, select “exhibit”, then “vessel”, and then “kylix” in the Class Browser; then select “exhibit22” in the Instance Browser.
- Select the target natural language and user type in the “Text Previews” tab.⁷ If you tick “Generate Comparisons”, the generated texts will occasionally contain comparisons to individuals described in previous texts and/or other individuals of the ontology.
- Specify the “Maximum Graph Distance in Content Selection”; see the articles mentioned in the introduction for an explanation of this parameter. For M-PIRO’s ontology, we recommend setting this value to 2.
- Press “Preview”. A textual description of the selected class or individual should appear.

If you generate previews for several individuals (e.g., try generating previews for the three kylikes of M-PIRO’s ontology, then generate a preview for the lekythos “exhibit15”), you will notice that NaturalOWL avoids repeating information that has already been conveyed; if you have ticked “Generate Comparisons”, it will also generate comparisons to previous individuals. Clicking on “Reset interaction history” will force NaturalOWL to “forget” the previous descriptions it has produced. Ticking “Show Syntactic and Semantic Annotations” will show the generated texts with syntactic and semantic markup in XML.

Please note that currently NaturalOWL is mostly tuned to describe individuals (e.g., “exhibits” in M-PIRO’s ontology). The texts that are generated for classes are not complete descriptions of the classes.

3. Modifying the source code of NaturalOWL and using it within Protégé

The source code of NaturalOWL is included in the distribution (see directory “source”). Provided that you respect its GPL license, you may modify the source code to create a tailored version of NaturalOWL. To use the new version within Protégé, follow the following steps:

Step 1

Modify the Java files of the source code.

⁷ In M-PIRO’s ontology, the texts for experts do not convey information that the experts would already know (e.g., what a hydria was used for), nor information they would be able to infer from the image of the exhibit (e.g., that a particular amphora was painted with the red-figure technique).

Step 2

Create a new file “NL.jar”. We recommend modifying and using the file “create-jar.bat”, which is included in NaturalOWL’s distribution, to produce the new “NL.jar”. You will also need the files “comparisons-lexicon.rdf” and “log4j.properties”, which are also included in NaturalOWL’s distribution; they are used in “create-jar.bat”. Change also “manifest.mf”, if necessary; for example, a new entry in “manifest.mf” is needed when a new Protégé tab is created.

Step 3

Copy “NL.jar” to the “plugins\gr.aueb.cs.nlg.ProtegeAuthoringTool” subdirectory of Protégé’s installation directory.

4. Using NaturalOWL with the ELEON authoring tool

NCSR “Demokritos” has extended M-PIRO’s authoring tool, now called ELEON, to support NaturalOWL.⁸ Compared to Protégé, ELEON provides an easier way to configure NaturalOWL for new application domains. Consult the following article for more information on ELEON:

- D. Bilidas, M. Theologou and V. Karkaletsis, “Enriching OWL Ontologies with Linguistic and User-related Annotations: the ELEON system”. Proceedings of the *IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, Patras, Greece, 2007.

5. How to call NaturalOWL’s generation engine from your software

You may call NaturalOWL directly from another application. File “TestNLGEngine.java” provides an example of how to do this. It is also possible to use NatualOWL with the Personalization Server of NCSR “Demokritos”, which provides persistent personal models and other facilities; contact NCSR “Demokritos” for further details.

⁸ ELEON can be downloaded from <http://www.iit.demokritos.gr/skel/>.