



D9v01. WSMO Editor

WSMO Working Draft 29 June 2004

This version:

<http://www.wsmo.org/2004/d9/v0.1/20040629>

Latest version:

<http://www.wsmo.org/2004/d9/v0.1/>

Previous version:

<http://www.wsmo.org/2004/d9/v0.1/20040527>

Editors:

Holger Lausen
Michael Felderer

This document is also available in non-normative [PDF](#) version.

Table of contents

- [1. Introduction](#)
 - [2. Download](#)
 - [3. Installation](#)
 - [4. Documentation and Example](#)
 - [5. Discussion, Future and Related Work](#)
 - [6. Change Log](#)
 - [7. References](#)
 - [Acknowledgement](#)
 - [Appendix A: Javadoc](#)
-

1. Introduction

This prototype is intended to enable interested parties to experiment with the formal description provided by WSMO. It is based on the Flora Query Tab [[Sintek, 2001](#)] written

by Michael Sintek. Note that there have been no functional changes to the software from 19 April 2004.

2. Download

There are two download options:

- [Full](#) WSMOTab including Flora
- [Core](#) only the WSMOTab library (updates only)

The latter is intended for people who want to upgrade to a newer version of the plugin, in which case there is no need to reinstall the Flora engine.

3. Installation

The WSMOTab is a plugin for Protege [\[Crubézy et al., 2004\]](#) (build 85, compiled for Protégé 2.0 or above).

Full Installation (first install)

- Get the recommended [build](#) of Protege (see above)
- Download this [zip file](#) and unzip its file to e.g.: "C:\WSMOTab" (make sure that your path does not contain spaces, since it was reported that xsb does not work properly in that case)
- Copy the "WSMOTab.jar" and "antlr.jar" into your protege plugin directory (e.g. "C:\Program Files\Protege_2.0\plugins")
- Add the following configuration parameters to your protege.properties file:

```
flora.directory=C:\\WSMOTab\\tmp\\
flora.xsb.command=C:\\WSMOTab\\XSB2.2-stderr\\config\\x86-pc-
windows\\bin\\xsb
flora.xsb.stderr=true
Adapt the paths to your system if necessary.
```

- Start Protege and open the WSMO_Lite project ("C:\WSMOTab\examples").

Minimal Installation (Core - updates only)

- Download this [zip file](#) and unzip it into your existing WSMOTab directory (e.g. "C:\WSMOTab") and copy WSMOTab.jar and antlr.jar into your protege plugin directory (e.g.: "C:\Program Files\Protege_2.0\plugins")

4. Documentation and Example

The current version of the Editor implements WSMO-Standard V02 [\[Roman et al., 2004\]](#). Please note that this is a early release, a lot of improvements are needed.

Limitations: The meta model for the ontology is simplified (according to the OKBC compliant meta model of Protege, i.e. we support at the moment concepts (classes) and attributes (facets)). It is currently not possible to modularize ontologies. All information (WSMO, domain ontology and instance data are exported to Flora).

Usage: After starting Protege load the WSMO ontology that came with the zip file ("C:\WSMOTab\examples"). In the classes and instances Tab you can maintain your domain ontology. Within the Web Service Tab you can create the annotation for your service. Note, that all instances will be exported into a separate file (when invoking flora), so you can browse through the relevant F-logic [Kifer et al., 1995] syntax.

Within the Flora Tab, you can add additional axioms to the knowledge base and execute specific queries. For a more detailed presentation on this feature, please refer to the explanations given at [Sintek, 2001] of the Flora Tab by Michael Sintek.

Screenshots:

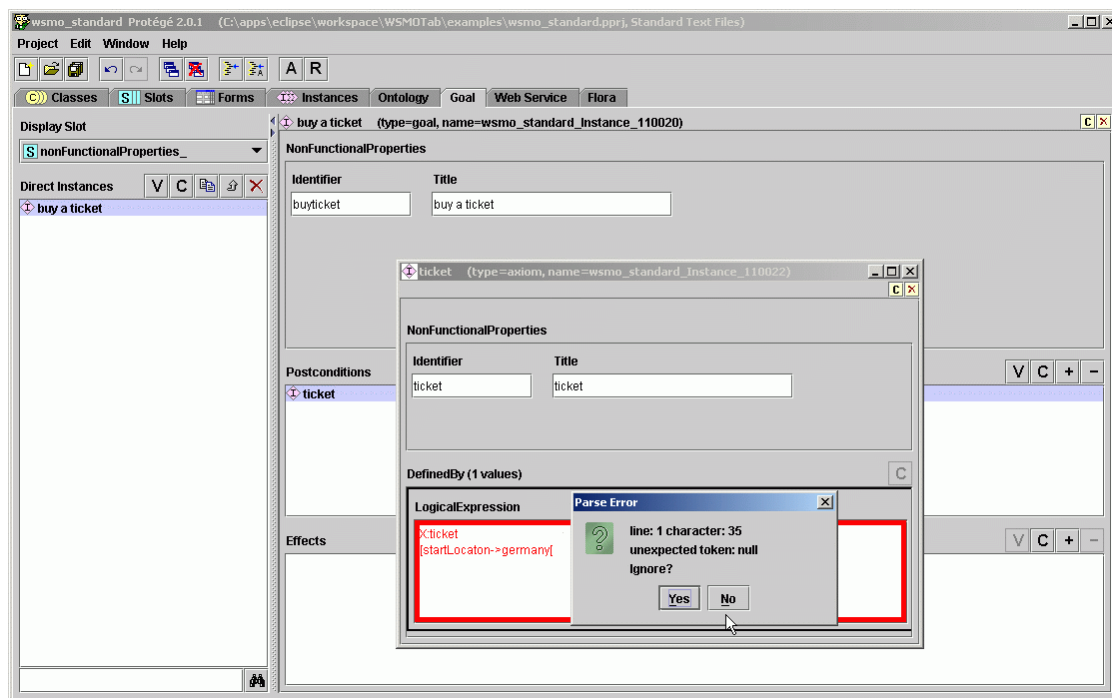


Figure 1: Axiom Definition (syntax check)

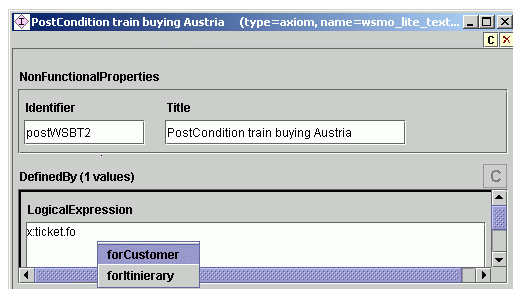


Figure 2: Syntax Completion

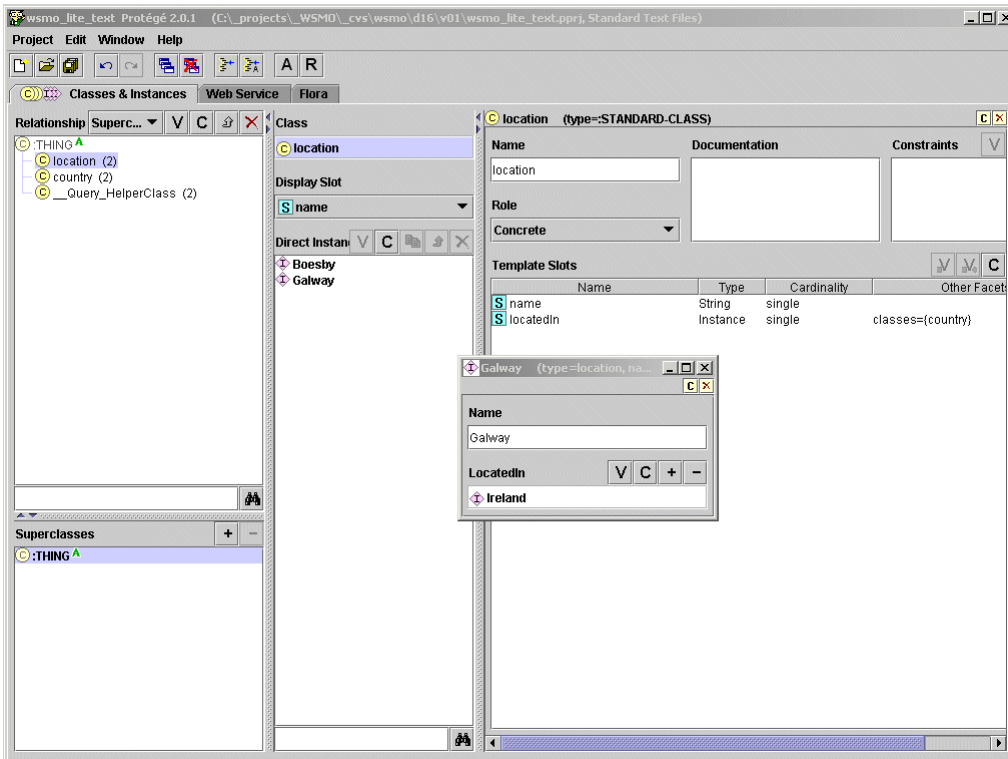


Figure 3: Ontology and instance construction

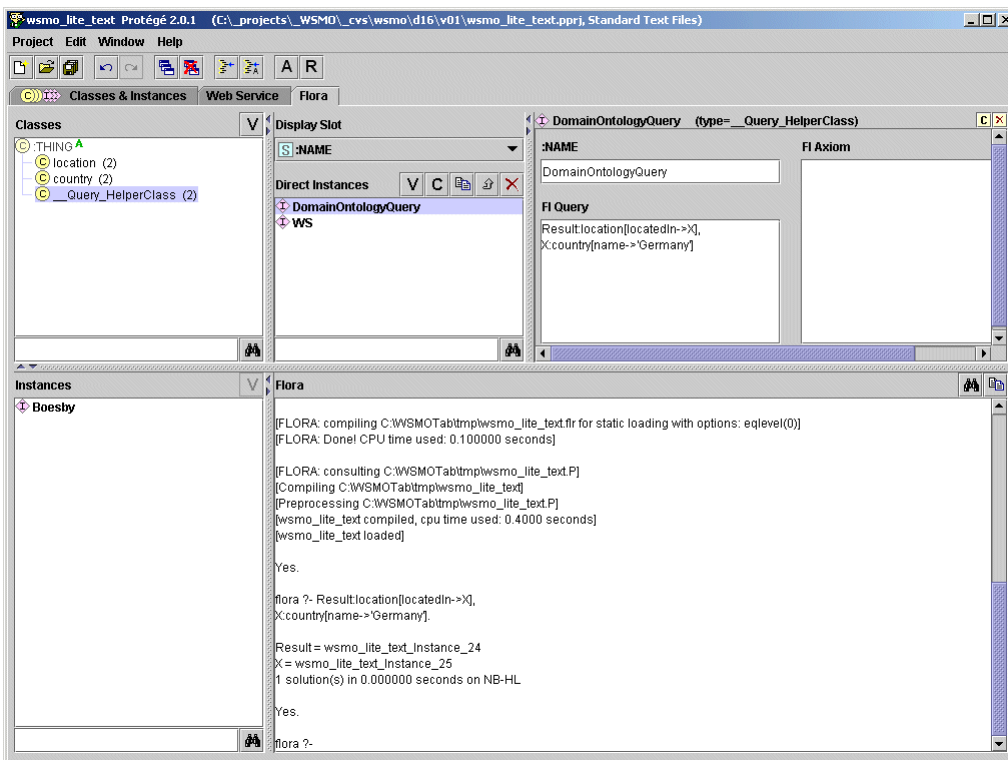


Figure 4: Interface to Flora

Development: A short documentation on the development environment:

Development Notes of Protege-2000 Plugins with Eclipse

This document is about the development of Protege-2000 Plugins in Java on the Eclipse platform. To show how the development works in principle, a tab-widget plugin, the simplest type of plugin, is developed

See Protege-2000 Programming Development Kit [[Crubézy et al., 2004](#)] for information for information about programming Protege-2000 plugins and Eclipse platform [[Eclipse, 2004](#)] for information about the implementation of Java projects on the eclipse platform

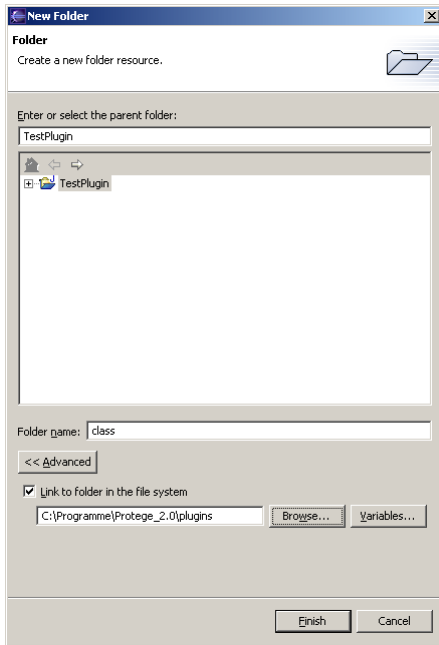
The integration of Protege plugins into Protege is very simple in principal. There are only two steps which need to be done for the integration of new plugins:

1. Copy the compiled .class or .jar file(s) of the plugin to the Protege "plugins" directory
2. Add the new plugin to the plugins directory manifest by editing the file "plugins/meta-inf/manifest.mf"

Eclipse allows one to automise and integrate these steps with the implementation of the plugin itself

Class-based Integration of plugins

1. Create a folder for the source files, e.g. "src", and a separate folder for the binary class files, e.g. "class", and link it to the plugins folder of Protege as shown in the following screensho

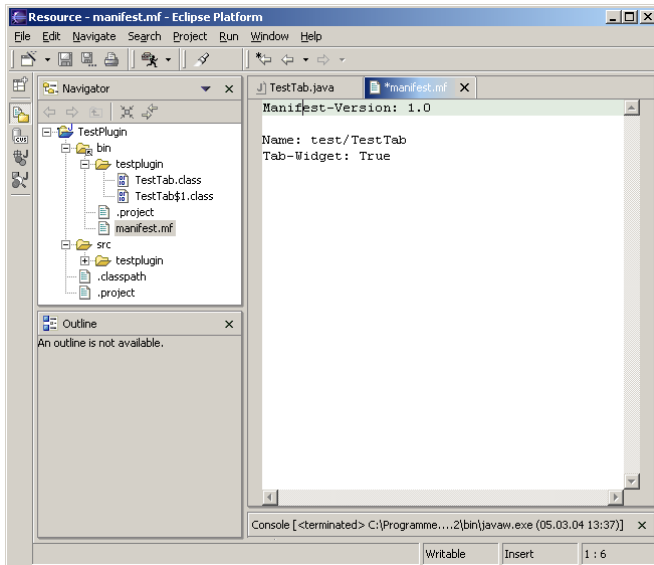


2. Set the new output file to the created directory by changing the "default output folder" in Project/Properties/Java Build Path.

NOTE: Because updating the output folder **overwrites the actual contents** of the plugin file, backup the contents of **Proteges plugins folder** (at least the meta-inf subdirectory) and copy it back after linking the output folder.

3. Develop your plugin as explained in the Protege-2000 Plugin Tutorial [[Crubézy et al., 2004](#)] by adding the protege.jar library to Project/Properties/Java Build Path/Libraries and implementing the source code of your plugin into the source folder.

4. To register the plugin, it must be added to OWL manifest file. This can also be done within Eclipse by restoring the manifest.mf file (see the following screenshot):



5. Compile and run the plugin from within Eclipse

References

[Eclipse, 2004] *Eclipse Universal Tool Platform.*, available from <http://www.eclipse.org/>.

[Crubézy et al., 2004] Monica Crubézy, Olivier Dameron, Ray Ferguson, Holger Knublauch, Mark Musen, Natasha Noy, Daniel Rubin, Samson Tu, Jennifer Vendetti. *Protege Programming Development Kit*, the Stanford University School of Medicine, available from <http://protege.stanford.edu/doc/pdk/index.html>.

JavaDoc: The Java Documentation can be found [here](#).

5. Discussion, Future and Related Work

This Protege Plug In can be seen as early prototyping that explores the possibility to integrate a WSMO Editor in a common ontology environment such as Protege. Other efforts, such as the second version of the SWWS Studio (WSMO Edition) are implemented as stand alone java application. It is available from <http://stronghold.sirma.bg/swws/> and offers support for all language constructs defined in the WSMO version from the 18th of April (<http://www.wsmo.org/2004/d16/d16.1/v0.2/20040418/>).

However the concrete syntax for the annotation of WSMO concepts significantly changed in the past view month. The concrete syntax specification can be found in WSMO Deliverable 16.1 (BNF grammar for WSMO language), which is applied in WSMO Deliverable 3.2 (WSMO Use Case Modeling and Testing) and the corresponding conceptual model is defined in WSMO Deliverable D2 (Web Service Modeling Ontology - Standard). So far only the aspects dealing with ontology definition and their usage in the web service annotation have been studied in more detail in those documents, but a full WSMO Editor is also expected to deal with the interfaces (choreography and

orchestration) including a way to ground a Service Description to a concrete technology (e.g. with a WSDL document). Additionally also Mediators need to be handled. Given that, a plug-in designed for a pure ontology editing environment, such as Protege, might in future be too inflexible to handle all the requirements, more over WSMO has defined its own Meta-Model for ontologies, which uses different terminology and concepts (w.r.t. to the OKBC model used by Protege).

For above mentioned reasons we are currently considering building a new editor not based on Protege and currently gathering requirements for it. Here we just give a brief list of aspects to be considered:

- Graphical User Interface to design all description elements of WSMO
- Export and Import Interface to different repositories (e.g. WSMO Registry)
- Export and Import of Ontologies in different formats (e.g. OWL)
- Integration of Reasoning Support (e.g. by conversion to a specific syntax (e.g. flora2) and execution in an appropriate reasoner).
- Mapping Support for a concrete grounding of the conceptual descriptions (e.g. WSDL)

Note that this list has no claims to be sound or complete by any means, but it should outline the broad scope of such an editor. It also illustrates that a plug-in architecture is needed.

6. Change Log

The following changes have been made with respect to the version 0.1 from the 27th of May 2004:

- Editorial changes and included a new Section "Discussion, Future and Related Work."

7. References

[Crubézy et al., 2004] Monica Crubézy, Olivier Dameron, Ray Fergerson, Holger Knublauch, Mark Musen, Natasha Noy, Daniel Rubin, Samson Tu, Jennifer Vendetti. *Protege Ontology Editor*, the Stanford University School of Medicine, available from <http://protege.stanford.edu/>.

[Kifer et al., 1995] M. Kifer, G. Lausen, and James Wu: Logical foundations of object oriented and frame-based languages. *Journal of the ACM*, 42(4):741-843, 1995.

[Sintek, 2001] M. Sintek. *The Flora Query Tab*, German Research Center for Artificial Intelligence, available from <http://www.dfki.uni-kl.de/~sintek/FloraTab/>.

[Roman et al., 2004] D. Roman, U. Keller, H. Lausen (Eds.): *D2v03. Web Service Modeling Ontology - Lite (WSMO-Lite)*, version 0.3 available at <http://www.wsmo.org/2004/d2/v0.3/20040329/>

Acknowledgement

The work is funded by the European Commission under the projects DIP, Knowledge Web, Ontoweb, SEKT, SWWS, Esperanto, and h-TechSight; by Science Foundation Ireland under the DERI-Lion project; and by the Vienna city government under the CoOperate programme.

The editors would like to thank to all the [members of the WSMO working group](#) for their advises and inputs to this document.



[webmaster](#)