



**NeOn: Lifecycle Support for Networked Ontologies**

**Integrated Project (IST-2005-027595)**

**Priority: IST-2004-2.4.7 – “Semantic-based knowledge and content systems”**

---

## **D6.7.2 Update of the core NeOn Toolkit**

---

**Deliverable Co-ordinator:** Michael Erdmann  
**Deliverable Co-ordinating Institution:** Ontoprise GmbH (ONTO)  
**Other Authors:** Werner Hihn

This document is an accompanying document to the main contribution of this deliverable, which is a software deliverable and embraces an update of the open-source version of the core NeOn Toolkit V1.2 which now contains complete modelling facilities for OWL-DL ontologies..

Document Identifier:	NEON/2008/D6.7.2/v1.1	Date due:	September 30, 2008
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	October 31, 2008
Project start date:	March 1, 2006	Version:	v1.1
Project duration:	4 years	State:	Final
		Distribution:	Public

## NeOn Consortium

This document is a part of the NeOn research project funded by the IST Programme of the Commission of the European Communities by the grant number IST-2005-027595. The following partners are involved in the project:

<p>Open University (OU) – Coordinator          Knowledge Media Institute – KMi          Berrill Building, Walton Hall          Milton Keynes, MK7 6AA          United Kingdom          Contact person: Martin Dzbor, Enrico Motta          E-mail address: {m.dzbor, e.motta} @open.ac.uk</p>	<p>Universität Karlsruhe – TH (UKARL)          Institut für Angewandte Informatik und Formale          Beschreibungsverfahren – AIFB          Englerstrasse 11          D-76128 Karlsruhe, Germany          Contact person: Peter Haase          E-mail address: pha@aifb.uni-karlsruhe.de</p>
<p>Universidad Politécnica de Madrid (UPM)          Campus de Montegancedo          28660 Boadilla del Monte          Spain          Contact person: Asunción Gómez Pérez          E-mail address: asun@fi.upm.es</p>	<p>Software AG (SAG)          Uhlandstrasse 12          64297 Darmstadt          Germany          Contact person: Walter Waterfeld          E-mail address: walter.waterfeld@softwareag.com</p>
<p>Intelligent Software Components S.A. (ISOCO)          Calle de Pedro de Valdivia 10          28006 Madrid          Spain          Contact person: Jesús Contreras          E-mail address: jcontreras@isoco.com</p>	<p>Institut 'Jožef Stefan' (JSI)          Jamova 39          SI-1000 Ljubljana          Slovenia          Contact person: Marko Grobelnik          E-mail address: marko.grobelnik@ijs.si</p>
<p>Institut National de Recherche en Informatique          et en Automatique (INRIA)          ZIRST – 655 avenue de l'Europe          Montbonnot Saint Martin          38334 Saint-Ismier          France          Contact person: Jérôme Euzenat          E-mail address: Jerome.euzenat@inrialpes.fr</p>	<p>University of Sheffield (USFD)          Dept. of Computer Science          Regent Court          211 Portobello street          S14DP Sheffield          United Kingdom          Contact person: Hamish Cunningham          E-mail address: hamish@dcs.shef.ac.uk</p>
<p>Universität Koblenz-Landau (UKO-LD)          Universitätsstrasse 1          56070 Koblenz          Germany          Contact person: Steffen Staab          E-mail address: staab@uni-koblenz.de</p>	<p>Consiglio Nazionale delle Ricerche (CNR)          Institute of cognitive sciences and technologies          Via S. Martino della Battaglia,          44 - 00185 Roma-Lazio, Italy          Contact person: Aldo Gangemi          E-mail address: aldo.gangemi@istc.cnr.it</p>
<p>Ontoprise GmbH. (ONTO)          Amalienbadstr. 36          (Raumfabrik 29)          76227 Karlsruhe          Germany          Contact person: Jürgen Angele          E-mail address: angele@Ontoprise.de</p>	<p>Food and Agriculture Organization          of the United Nations (FAO)          Viale delle Terme di Caracalla 1          00100 Rome          Italy          Contact person: Marta Iglesias          E-mail address: marta.iglesias@fao.org</p>
<p>Atos Origin S.A. (ATOS)          Calle de Albarracín, 25          28037 Madrid, Spain          Contact person: Tomás Pariente Lobo          E-mail address: tomas.parientalobo@atosorigin.com</p>	<p>Laboratorios KIN, S.A. (KIN)          C/Ciudad de Granada, 123          08018 Barcelona, Spain          Contact person: Antonio López          E-mail address: alopez@kin.es</p>

## Work package participants

The following partners have taken an active part in the work leading to the elaboration of this document, even if they might not have directly contributed writing parts of this document:

Ontoprise

## Change Log

Version	Date	Amended by	Changes
0.1	2008-09-01	Michael Erdmann	Initial version of deliverable document
0.2	2008-09-10	Michael Erdmann	Setup of document structure
0.5	2008-10-10	Michael Erdmann	Migration of content of previous document
0.6	2008-11-21	Michael Erdmann	Documentation NTK licenses
0.7	2008-11-22	Michael Erdmann	Documentation of plugins
0.8	2008-11-23	Michael Erdmann	OLW features
0.9	2008-11-23	Michael Erdmann	Intro, Conclusion, Exec Summary
1.0	2008-11-23	Michael Erdmann	Preparation of review-ready version
1.1	2008-12-10	Michael Erdmann	Incorporation of review comments

## Executive Summary

This deliverable consists of two parts, (i) the software representing the third iteration of the NeOn Toolkit (V1.2) and (ii) this accompanying document. The NeOn Toolkit V1.2 now contains complete modelling facilities for OWL-DL ontologies.

The NeOn Toolkit was released at the end of September 2008 in three different versions: a source-code version for developers, a basic and an extended configuration for users. The different versions are released under different software licenses, which range from the open-source Eclipse Public License, over a free-ware license to a proprietary evaluation license for the extended configuration that contains commercial plugins.

This new version of the toolkit provides a greatly improved basis for the plugins developed by the scientific partners in the NeOn project. These plugins were mainly geared towards the OWL language and are now complemented by editing facilities for OWL-DL ontologies.

## Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>6</b>
<b>2</b>	<b>The OWL Modelling Facilities of the NeOn Toolkit.....</b>	<b>7</b>
2.1	The OWL Perspective.....	7
2.2	The OWL Class Entity Properties View .....	8
2.3	Usability Features .....	11
<b>3</b>	<b>The License Model of the NeOn Toolkit.....</b>	<b>14</b>
3.1	Eclipse Public License .....	14
3.2	Freeware License .....	15
3.3	Evaluation License.....	15
<b>4</b>	<b>The Plugins of the NeOn Toolkit .....</b>	<b>17</b>
4.1	The Basic Configuration .....	17
4.2	The Extended Configuration .....	19
4.2.1	<i>Extended F-logic Modelling .....</i>	<i>19</i>
4.2.2	<i>F-logic Testing and Debugging .....</i>	<i>20</i>
4.2.3	<i>F-logic Querying .....</i>	<i>20</i>
4.2.4	<i>Visualizing and Printing F-logic ontologies.....</i>	<i>20</i>
4.2.5	<i>Graphical Mapping of F-logic ontologies.....</i>	<i>20</i>
4.2.6	<i>Support for Optimizing F-logic rule bases .....</i>	<i>21</i>
4.2.7	<i>Online documentation for F-logic and OntoBroker.....</i>	<i>21</i>
4.2.8	<i>Lifting of Relational Data Bases .....</i>	<i>21</i>
4.2.9	<i>Populating F-logic Ontologies from Non-Ontological Sources.....</i>	<i>21</i>
4.2.10	<i>RDF-Schema Modelling .....</i>	<i>21</i>
4.2.11	<i>WebService Support .....</i>	<i>22</i>
<b>5</b>	<b>How to Obtain the NeOn Toolkit.....</b>	<b>23</b>
5.1	The Source Code Version .....	23
5.1.1	<i>Source Code Management .....</i>	<i>23</i>
5.1.2	<i>Contributions to the NeOn Toolkit Development.....</i>	<i>23</i>
5.1.3	<i>Prerequisites for Using and Extending the Source code.....</i>	<i>23</i>
5.2	The Executable Versions of the NeOn Toolkit.....	24
<b>6</b>	<b>Future Steps for the Core NeOn Toolkit .....</b>	<b>25</b>
<b>7</b>	<b>Conclusion .....</b>	<b>26</b>
<b>8</b>	<b>Reference .....</b>	<b>27</b>
<b>9</b>	<b>Appendix – License Texts.....</b>	<b>28</b>
9.1	EPL.....	28
9.2	Freeware License .....	32
9.3	Evaluation License.....	33

## List of Figures

Figure 1: The OWL perspective	7
Figure 2: Defining restrictions for OWL classes	9
Figure 3: Defining taxonomic relations between classes	10
Figure 4: Defining annotations for classes	11
Figure 5: Navigation facility “jump to entity”	11
Figure 6: Autocomplete in the edit-boxes of the entity properties view	12
Figure 7: Different modes for displaying entity names	13

# 1 Introduction

This deliver is the second iteration of its predecessor, D6.6.1 “Realisation & early evaluation of basic NeOn tools in NeOn Toolkit V1” [NeOn D6.6.1]. It describes the advancements of the NeOn Toolkit core plugins which are now contained in the NeOn Toolkit release v1.2.

The goal of this deliverable, which mainly consists of the software artefact (i.e. the NeOn Toolkit v1.2) is the provision of sufficient modelling capabilities to create and maintain OWL DL ontologies. The software that we released end of September 2008 now provides this functionality and thus offers an OWL modelling platform for NeOn partners, the NeOn use cases and other NeOn users. Since the main contribution of this deliverable is a piece of source code, we restrict the presentation here to a brief overview of the newly developed features. A more detailed description is contained in the update to the documentation deliverable “NeOn Toolkit Documentation” [NeOn D6.6.2]

The introduction to the OWL modelling feature is presented in Section 2. Sections 3 and 4 present the licensing decisions and the provided functionalities of the different configurations of the NeOn Toolkit. We close with some concluding remarks and an outlook towards future developments.

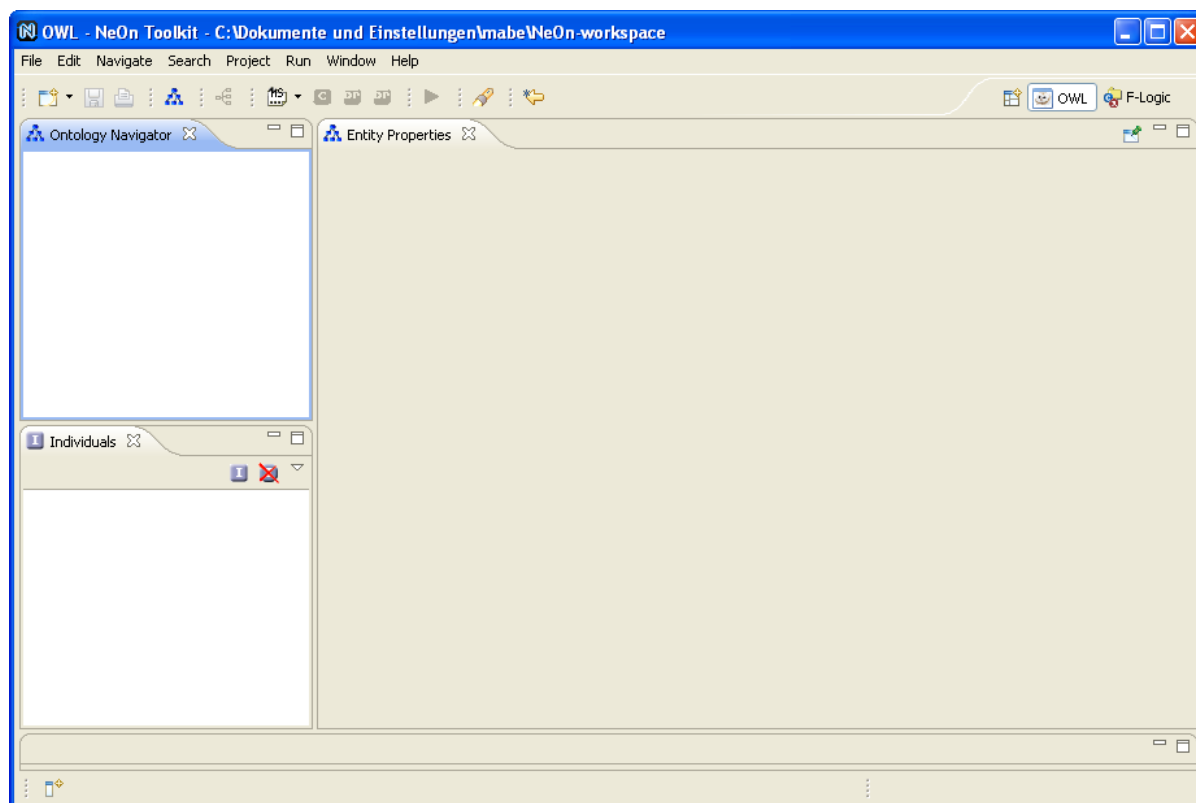
## 2 The OWL Modelling Facilities of the NeOn Toolkit

With the current release V1.2 of the NeOn Toolkit we complete the implementation of the OWL modelling support. The NeOn Toolkit enables users to load, browse, modify and store OWL-DL ontologies. For a complete documentation of this feature cf. [NeOn D6.6.2].

### 2.1 The OWL Perspective

The OWL Editor is structured in a similar way as the F-logic modelling facilities of the NeOn Toolkit. The OWL modelling tools are organized in the so-called *OWL perspective* (cf. Figure 1), which consists of the following main panels:

- *Ontology Navigator*: The Ontology Navigator shows the ontology projects, their corresponding ontologies, and the elements of the ontologies organized in a hierarchical manner. It supports drag-and-drop for moving or copying entities and offers a couple of actions on entities via context-menus.
- *Individuals panel*: The Individuals panel displays all instances of the selected class.
- *Entity Properties panel*: The Entity Properties panel is the main work area for defining and modifying ontology elements.



**Figure 1: The OWL perspective**

The OWL editor has been completely redesigned. Especially, the entity property views of the different OWL entities have changed drastically. The details of each entity are organized on multiple tabs to separate the different aspects. Each tab contains several sections, which typically represent special types of OWL axioms. Each section can be minimized to focus on special information. The whole UI is completely keyboard navigatable which allows experienced users to quickly navigate and modify the model,

## 2.2 The OWL Class Entity Properties View

As an example for the new presentation of the OWL model we explain the layout for the details of OWL classes (i.e. the OWL class Entities Properties View) here. For OWL classes three aspects have been identified:

- Modelling of properties relevant for the class via property restrictions
- Modelling of taxonomic relationships to other classes
- Annotations for the class

### Modelling of properties relevant for the class via property restrictions

OWL can be used to model situations in which the members of a particular class must have certain properties. You can use various restriction statements to indicate that something should apply to all values of a property versus only for some members. In addition, you can even specify a particular value for a property of all members of a class. The OWL editor provides the following restriction types:

- owl:allValuesFrom (ALL)
- owl:someValuesFrom (SOME)
- owl:hasValue (HAS\_VALUE)
- owl:maxCardinality (AT\_MOST/MAX)
- owl:minCardinality (AT\_LEAST/MIN)
- owl:cardinality (EXACTLY/CARD)

The three cardinality restrictions can take class descriptions as range argument. In this case they are treated as qualified number restrictions according to the upcoming OWL 2 specification<sup>1</sup>.

Since often, the ranges of properties are represented by complex classes we adopted the Manchester Syntax<sup>2</sup> for entering and displaying complex class descriptions in the OWL editor. The Manchester Syntax is widely deployed since it is easier to read and write than the abstract description logic notation.

---

<sup>1</sup> Cf. [http://www.w3.org/TR/2008/WD-owl2-new-features-20081202/#F5:\\_Qualified\\_cardinality](http://www.w3.org/TR/2008/WD-owl2-new-features-20081202/#F5:_Qualified_cardinality)

<sup>2</sup> <http://www.w3.org/2007/OWL/wiki/ManchesterSyntax>



The screenshot shows a software interface for defining OWL class restrictions. The main window is titled 'Entity Properties' and contains a section for 'OWL Class'. The class name is 'Person' and the namespace is 'http://www.example.org/sample#'. There are two main sections for defining restrictions: 'Super Restrictions' and 'Equivalent Restrictions'. Each section has a table with columns for 'Quantifier', 'Min/Max', 'Property', and 'Range'. In the 'Super Restrictions' section, two rows are shown, both with 'hasRelative' as the property and 'Person' as the range. Below the table is a 'Create new:' section with a dropdown for 'Select a Quantifier' and an 'Add' button. The 'Equivalent Restrictions' section has a similar 'Create new:' section. At the bottom of the window, there are three tabs: 'Class Restrictions', 'Taxonomy', and 'Annotations'.

**Figure 2: Defining restrictions for OWL classes**

Class restrictions can either be specified as Super Restrictions or Equivalent Restrictions. If you add the class restriction in the Super Restriction section, the class under consideration will become a subclass of the restriction class (i.e. you define a *necessary condition* for the class). On the other hand, if you add the class restriction in the Equivalent Restrictions section, both the restriction class and the restricted class will be interpreted as equivalent, which means that they always have exactly the same members (i.e. you define a *necessary and sufficient condition* for the class).

## Modelling of taxonomic relationships to other classes

On the taxonomy tab you can define classes by relating two class descriptions. Formally restrictions are also classes, but they have a different focus. They essentially talk about properties associated with a class and thus we provide a specialized user interface for them (cf. the section above).

OWL contains three language constructs for taxonomically combining class descriptions into class axioms:

- **Subclasses.** The language construct `rdfs:subClassOf` allows you to say that a class is a subset of another class.
- **Equivalent Classes.** The OWL language construct `owl:equivalentClass` allows you to say that a class has exactly the same members as another class.
- **Disjoint Classes.** The OWL language construct `owl:disjointWith` allows you to say that a class has no members in common with another class.

In addition to these language constructs, the OWL Editor allows you to specify the **super classes** of a class, which is the same as making the class under consideration a subclass of another (parent) class.

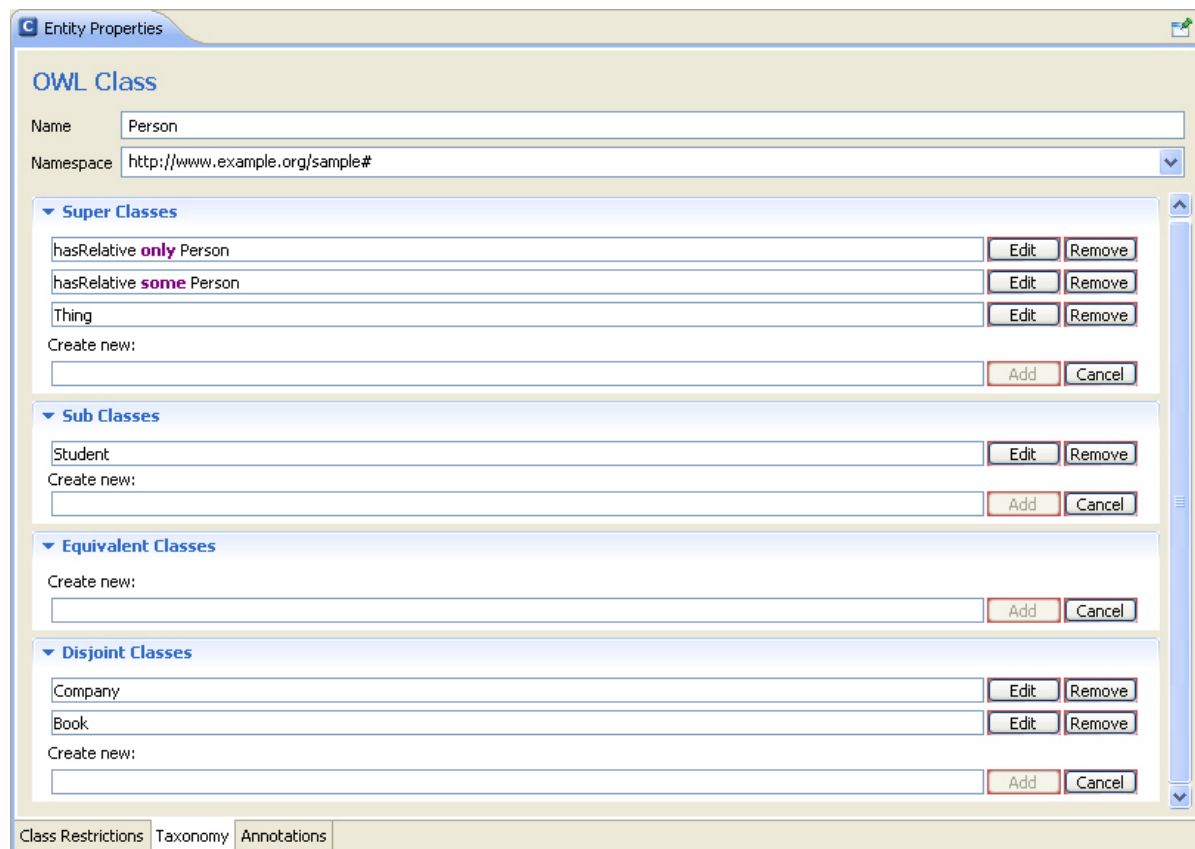


Figure 3: Defining taxonomic relations between classes

## Annotations for the class

The annotations tab is the same for all OWL entities. Here, users can add annotations that are used to add information to resources of the ontology. They contain useful information such as the creator or some human-readable labels (rdfs:label) and comments (rdfs:comment).

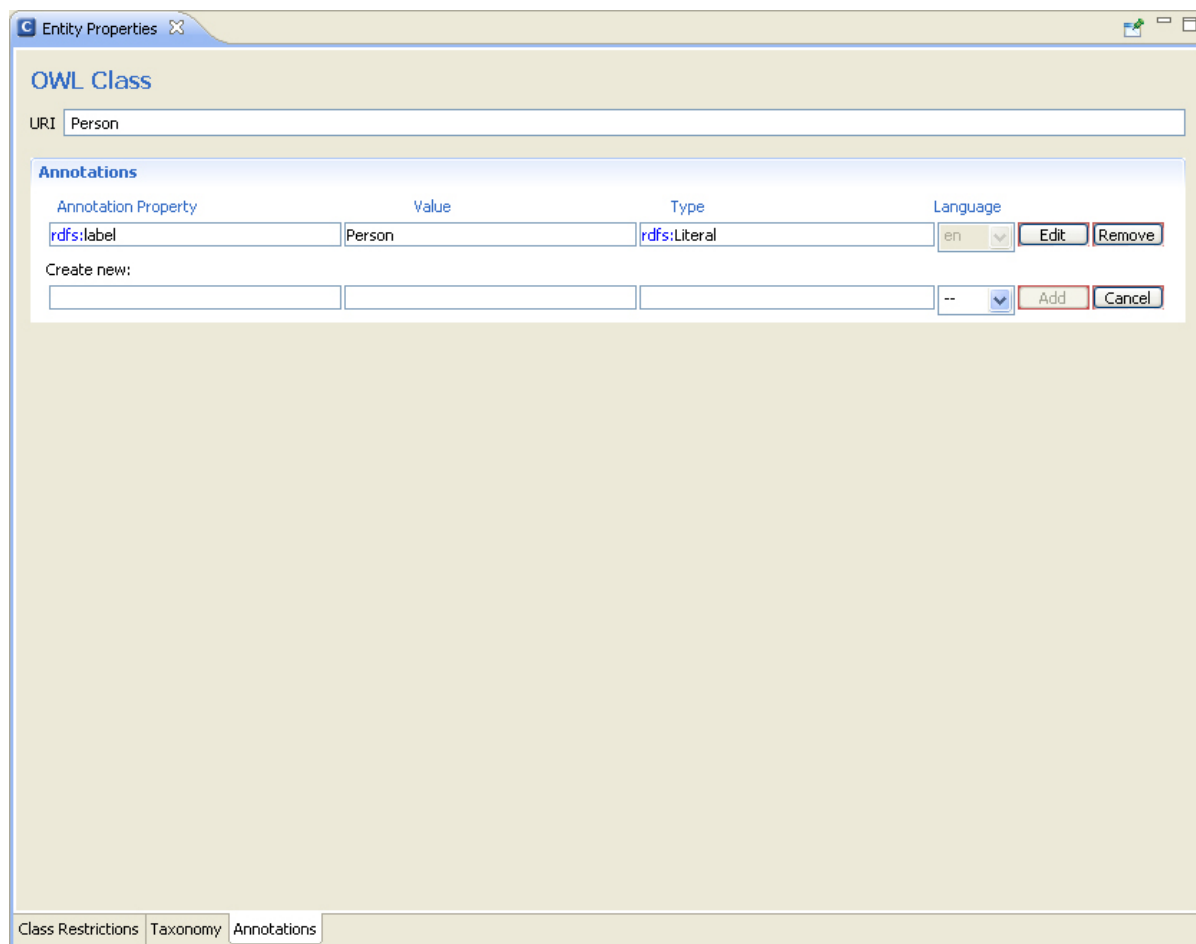


Figure 4: Defining annotations for classes

## 2.3 Usability Features

The OWL editor of the NeOn Toolkit provides a number of helpful features that makes modelling OWL ontologies easier for users:

### Navigation

If a resource is selected in either the Ontology Navigator or the Individuals panel, it will be loaded in the Entity Properties panel. From there you can navigate to other resources by pressing the Control key (Ctrl) and clicking its name.

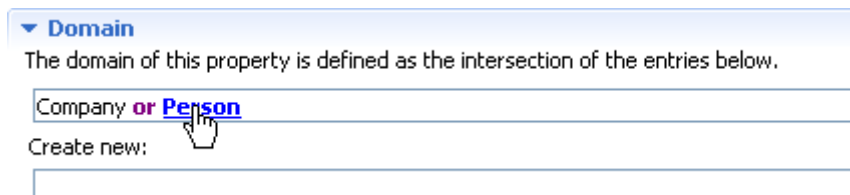


Figure 5: Navigation facility “jump to entity”

## Autocomplete

The OWL Editor also features a useful autocomplete function. Autocomplete involves the OWL Editor predicting an entity name that you want to type without actually typing it completely. It is available in all text boxes while in edit mode. You can start autocomplete by clicking *Ctrl+space* on your keyboard or by simply starting to type; autocomplete starts within a second.

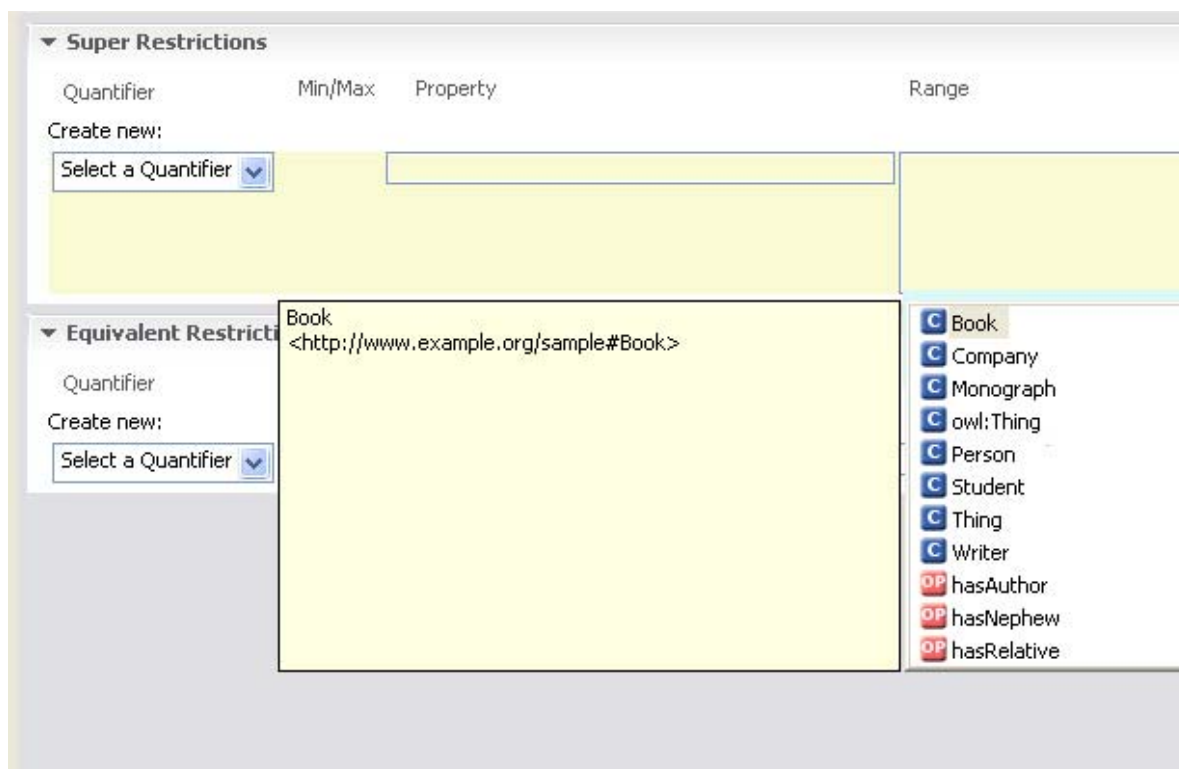


Figure 6: Autocomplete in the edit-boxes of the entity properties view

## Entity Label Mode

The labels that are displayed for entities in the different fields of the user interface depend on the selected *Entity Label Mode*. You can select between four different modes:

- **Complete URI.** Displays the complete URI of an entity (e.g., "http://www.example.org/sampleOntology#Company")
- **Local Name.** Displays only the local name of an entity (e.g., "Company"). This makes the ontology a lot more readable.
- **QName.** Displays the namespace prefix and local name of an entity (e.g., "onto:Company"). This is the recommended setting and is especially useful when multiple ontologies are used or different namespaces are in use.
- **Language.** If you provide a label (`rdfs:label`) in a specific language for an entity, this option displays the language label for every occurrence of the entity, thus providing multi-language support. Notice that only those languages are available which are specified in the Window >> Preferences >> Language Preferences dialog box.

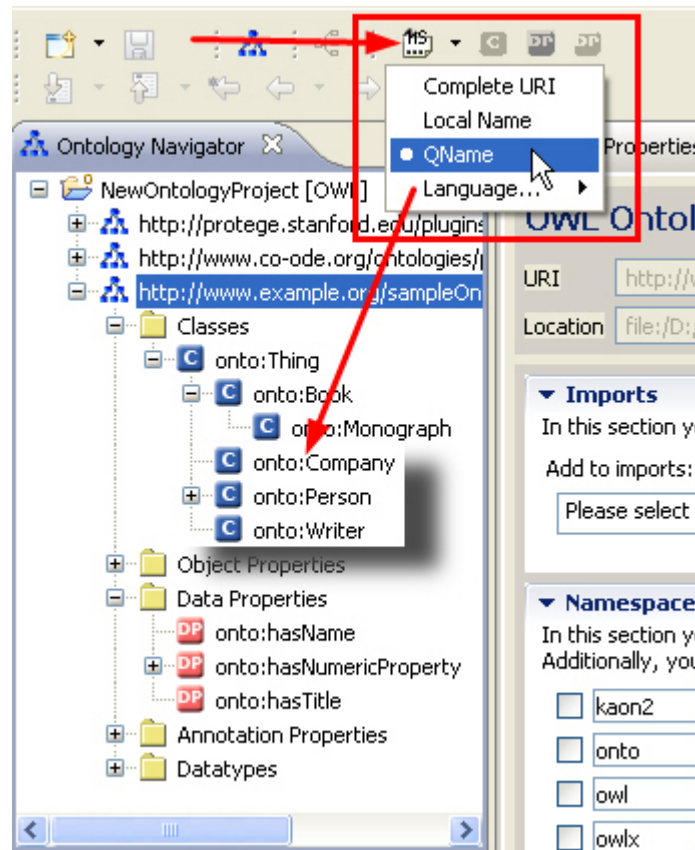


Figure 7: Different modes for displaying entity names

### 3 The License Model of the NeOn Toolkit

The NeOn Toolkit is a complex piece of software. It is built on top of the Eclipse platform and one of its goals is to provide a freely available ontology modelling framework for networked ontologies. The core components of the NeOn Toolkit have been “donated” to the NeOn project by Ontoprise (stemming from Ontoprise’s ontology engineering environment OntoStudio). This basic platform is extended by all NeOn project partners via individual contributions in the form of plugins.

Ontoprise is a commercial software vendor, thus there exists a natural tension between the goal of providing a free software on the one hand and protecting the business interests on the other hand. After a lot of discussion and the advice by competent law firms, Ontoprise and the NeOn Consortium decided to provide different parts of the NeOn Toolkit under different licenses: an established open-source license for the open-source components and proprietary licenses for the non-open source components:

- The open-source components of the NeOn Toolkit are released under the open source *Eclipse Public License*.
- The core datamodel for ontology-modelling is Ontoprise IP and is released under a *freeware license*.
- The commercial plugins by Ontoprise that are bundled with the extended configuration of the NeOn Toolkit are released under an *evaluation licence*.

In the next subsections these licenses will be briefly presented. The full license texts can be found in the Appendix.

#### 3.1 Eclipse Public License

The NeOn Consortium releases the open-source parts of the NeOn Toolkit under the open-source Eclipse Public License version 1.0 (EPL<sup>3</sup>, cf. Section 9.1). This is the same license that all Eclipse plugins are distributed with. Since we are redistributing a couple of plugins from the Eclipse platform with the NeOn Toolkit the use of EPL is compatible. Actually, the Eclipse foundation recommends using EPL for plugins.

EPL is in spirit similar to the widely used GNU Public License (GPL). Unfortunately the formulation of GPL is not as clear as EPL with respect to bundling GPL and non-GPL code. Possibly GPL-code can contaminate bundled code, i.e. requires that the bundled code must also be released under GPL: The formulation of EPL with respect to bundling is much clearer. Especially it makes clear that individual plugins can be released their own license, i.e. there is no “viral effect” of EPL of any kind. As a side effect it makes the EPL and GPL licenses incompatible with each other.

Other important features of the EPL include, that the EPL-licensed components can be modified and redistributed. EPL also excludes warranties and liabilities by the component developers.

The following table provides the list of NeOn Toolkit core plugins that are released under EPL.

- `com.Ontoprise.jpowersgraph`
- `com.Ontoprise.ontostudio.datamodel`
- `com.Ontoprise.ontostudio.dependencies`
- `com.Ontoprise.ontostudio.gui`
- `com.Ontoprise.ontostudio.io`

---

<sup>3</sup> <http://www.eclipse.org/legal/epl-v10.html>

- `com.Ontoprise.ontostudio.ontovisualize`
- `com.Ontoprise.ontostudio.owl.gui`
- `com.Ontoprise.ontostudio.owl.help`
- `com.Ontoprise.ontostudio.owl.model`
- `com.Ontoprise.ontostudio.refactor`
- `com.Ontoprise.ontostudio.search`
- `com.Ontoprise.ontostudio.swt`
- `org.neontoolkit.gui`
- `org.neontoolkit.help`
- `org.neontoolkit.plugin`

## 3.2 Freeware License

The basic ontology management functionality (API for reading, manipulating and writing ontology models, persistency of ontologies) for both, F-logic and OWL is provided by the so-called *datamodel* plugins. These plugins are provided in binary form and fall under a *freeware license* (cf. Section 9.2). They can be freely used and developers can build new plugins on-top of the freeware plugins. Thus, this license is not an *open-source* license in the strict sense but closer in spirit to an end-user-license agreement (EULA). The freeware license especially prohibits decompiling and changing the code. In a similar manner as the EPL, the freeware license excludes warranties and liabilities by developers. Since the provided software is free, the excluded liabilities are probably also valid under German laws, which is an important argument for Ontoprise to support this license.

The following table provides the list of NeOn Toolkit core plugins that are released under the freeware license.

- `com.Ontoprise.dependencies`
- `datamodel`
- `datamodelBase`
- `dependencies`
- `Flogic-parser`
- `ontobroker-ng_kernel-g3`
- `ontobroker-ng_ontobroker-core`
- `ontobroker-ng_server`
- `Ontoprise-licensechecker`
- `touchgraph`
- `util`

## 3.3 Evaluation License

When building the (extended configuration of the) NeOn Toolkit we include a couple of commercial plugins by Ontoprise (e.g. graphically modelling rules or mappings). These plugins are released under an *evaluation licence* (cf. Section 9.3). The wording of this evaluation license is identical to the freeware license, except for the insertion of one paragraph, which restricts the free use of the provided software to a period of (currently) three months. The software comes with a license key file which makes sure that the extended plugins cannot be used after the evaluation period has expired. This license key file also disables advanced reasoning features like the execution of user-defined rules on the ontology management layer, once the evaluation period has ended.

The following table provides the list of NeOn Toolkit core plugins that are released under the evaluation license.

- com.darkmatter.diagram.editor
- com.Ontoprise.darkmatter.interfaces
- com.Ontoprise.darkmatter.plugin.ruleeditor
- com.Ontoprise.datamodel.objectmodel.api
- com.Ontoprise.flogic.help
- com.Ontoprise.ontobroker.help
- com.Ontoprise.ontostudio.dbschemaimport
- com.Ontoprise.ontostudio.debug
- com.Ontoprise.ontostudio.explanation
- com.Ontoprise.ontostudio.flogiceditor
- com.Ontoprise.ontostudio.genetics
- com.Ontoprise.ontostudio.graphviz
- com.Ontoprise.ontostudio.imports
- com.Ontoprise.ontostudio.imports.collocation
- com.Ontoprise.ontostudio.imports.filesystem
- com.Ontoprise.ontostudio.imports.wsdl
- com.Ontoprise.ontostudio.ontomap
- com.Ontoprise.ontostudio.ontomap.mathtransform
- com.Ontoprise.ontostudio.query
- com.Ontoprise.ontostudio.query.regressiontest
- com.Ontoprise.ontostudio.rdf
- com.Ontoprise.ontostudio.remoteob
- com.Ontoprise.ontostudio.rule
- com.Ontoprise.ontostudio.synonyms
- com.Ontoprise.ontostudio.uml
- com.Ontoprise.ontostudio.webservices
- com.Ontoprise.ontostudio.webservices.generated
- com.Ontoprise.ontostudio.webservices.query
- com.Ontoprise.ontostudio.webservices.query.logview
- com.Ontoprise.ontostudio.xlsannotation



## 4 The Plugins of the NeOn Toolkit

The NeOn Toolkit comes in two configurations: (i) the basic configuration, which coincides with the open-source version and (ii) an extended configuration that in addition to the basic functionality contains some commercial plugins provided by Ontoprise. The following table summarizes the main features of the two configurations:

open source / EPL	evaluation license
F-Logic ontology modelling OWL-DL modelling Import/Export of multiple ontology formats Graphical display of F-logic ontologies Search in OWL and F-logic ontologies On-line user manual for F-logic and OWL modelling	Graphical rule editor and debugger Form-based query interface for F-logic KBs OntoMap for mapping F-logic ontologies RDF(S) ontology modelling On-line user manual for OntoBroker Database schema lifting  Regression-test component for F-logic KBs Facility for formulating explanation texts for rules Web service import and export

### 4.1 The Basic Configuration

The basic configuration of the NeOn Toolkit provides functionality for basic ontology management for both F-logic and OWL-DL ontologies. It consists of the following plugins:

`com.Ontoprise.ontostudio.datamodel:`

This plug-in represents the core plug-in of the NeOn Toolkit since it represents the internal datamodel and thus the knowledge representation aspects of the application. It provides access to the underlying datamodel via an API. The datamodel is realized by OntoBroker and made accessible via the KAON2 API.

This plug-in is released in source code form under the EPL licensing model.

`com.Ontoprise.ontostudio.gui:`

This plug-in contains the core UI components and connects them with the underlying datamodel via the `com.Ontoprise.ontostudio.datamodel` plug-in. The main UI elements include the ontology navigator and the property editors for the different ontology entities. The user interface for the F-logic aspects of the NeOn Toolkit are also realized in this plug-in.

This plug-in is released in source code form under the EPL licensing model.

`com.Ontoprise.ontostudio.io:`

This plug-in provides functionality to import and export ontologies in different formats from and to the local file system or a WebDAV server (Web-based Distributed Authoring and Versioning).

This plug-in is released in source code form under the EPL licensing model.

`com.Ontoprise.ontostudio.swt:`

Here, the basic Eclipse SWT (Standard Widget Toolkit<sup>4</sup>) classes are extended and customized to support the GUI plug-in.

This plug-in is released in source code form under the EPL licensing model.

<sup>4</sup> <http://www.eclipse.org/swt/>

`com.Ontoprise.ontostudio.ontovisualize:`

The `ontovisualize` plug-in contains a view to graphically visualize F-logic ontologies using the `JPowerGraph` library<sup>5</sup> which is customized in the following plug-in.

This plug-in is released in source code form under the EPL licensing model.

`com.Ontoprise.jpowersgraph:`

Integration of the `jpowersgraph` library and some extensions for the visualization plug-in.

This plug-in is released in source code form under the EPL licensing model.

`com.Ontoprise.ontostudio.search:`

This plug-in provides some search functionalities for ontological entities like concepts, attributes, relations and instances.

This plug-in is released in source code form under the EPL licensing model.

`com.Ontoprise.ontostudio.refactor:`

The refactor plug-in provides functionalities and extension points for operations that modify ontologies in a complex manner, e.g. removing a property. It provides wizards that support users in monitoring and controlling their actions on ontology elements.

This plug-in is released in source code form under the EPL licensing model.

`org.neontoolkit.plugin:`

This is the branding plug-in. With this plug-in the toolkit can be customized regarding the splash-screen, the about-dialog etc.

This plug-in is released in source code form under the EPL licensing model.

`org.neontoolkit.gui:`

In this plug-in the extension points and associated Java interfaces are specified that allow for extending the toolkit with additional functionality. It is extended by the `com.Ontoprise.ontostudio.gui` plug-in for the F-logic modelling and by `com.Ontoprise.ontostudio.gui` plug-in for the OWL modelling user interfaces.

This plug-in is released in source code form under the EPL licensing model.

`org.neontoolkit.help:`

In this plug-in we define the on-line documentation for the basic features of the NeOn Toolkit as specified in [NeOn D6.7.1] and [NeOn D6.6.2]. The documentation is available via the `Help Contents` entry of the `Help` menu.

This plug-in is released in source code form under the EPL licensing model.

`com.Ontoprise.ontostudio.dependencies:`

This plug-in is just a wrapper for a number of 3<sup>rd</sup> party libraries that are used by the basic NeOn Toolkit plugins.

---

<sup>5</sup> <https://sourceforge.net/projects/jpowergraph/>

This plug-in is released under the EPL licensing model. Since this plug-in does not contain any source-code or own developments the associated licenses for the contained libraries apply.

`com.Ontoprise.ontostudio.owl.model:`

This plug-in provides means for conveniently accessing and manipulating OWL models from other plugins within the NeOn Toolkit. Its primary “client” is the `com.Ontoprise.ontostudio.owl.gui` plug-in.

This plug-in is released in source code form under the EPL licensing model.

`com.Ontoprise.ontostudio.owl.gui:`

In this plug-in we implemented the user interface for loading, storing, displaying, and modifying OWL ontologies. This is the central plug-in for all OWL modelling GUI elements.

This plug-in is released in source code form under the EPL licensing model.

`com.Ontoprise.ontostudio.owl.help:`

In this plug-in we define the on-line documentation for new OWL features of the NeOn Toolkit as specified in [NeOn D6.6.2]. The documentation is available via the `Help Contents` entry of the `Help` menu.

This plug-in is released in source code form under the EPL licensing model.

In addition to these open source plugins that are also delivered in source code form, the NeOn Toolkit brings a few more plugins that are only delivered in binary form. For the purposes of NeOn these plugins come under a freeware license. The following list of plugins together represent the underlying ontology modelling, storage and reasoning facilities provided by the NeOn Toolkit.

- `datamodel`
- `datamodelBase`
- `com.Ontoprise.dependencies`
- `dependencies`
- `flogic-parser`
- `kernel-g3`
- `ontobroker-core`
- `Ontoprise-licensechecker`
- `server`
- `touchgraph`
- `util`

## 4.2 The Extended Configuration

A long list of plugins is distributed in the so-called extended configuration of the NeOn Toolkit and fall under the proprietary evaluation license. In this section we will not list all of them individually; rather we will group them according to their functionality and give a brief introduction into the features provided by each group.

### 4.2.1 Extended F-logic Modelling

The modelling facilities provided by the basic configuration support the basic means for modelling ontologies in F-logic. The extended configuration adds plugins that greatly improve the expressivity of F-logic models that can be created with the NeOn-Toolkit. Most importantly, the *rules* plug-in implements a graphical rule-editor for authoring F-logic rules

that obey the schema defined in the ontologies. Along with the rules, user now can formulate *explanation* templates that can be used to justify the deduction process of retrieving query results. For each ontology element the user now can also define *synonyms* in different languages to improve the user experience in F-logic-based applications. The last plug-in in this group of extended F-logic modelling functionality provides a textual *F-logic editor* that enables users to textually create and modify ontologies. It offers support in the form of syntax highlighting and easy navigation even in bigger ontology files.

Plugins:

- `com.Ontoprise.ontostudio.rule`
- `com.Ontoprise.ontostudio.explanation`
- `com.Ontoprise.ontostudio.synonyms`
- `com.Ontoprise.ontostudio.flogiceditor`
- `com.darkmatter.diagram.editor`
- `com.Ontoprise.darkmatter.interfaces`
- `com.Ontoprise.darkmatter.plugin.ruleeditor`

#### 4.2.2 F-logic Testing and Debugging

The extended configuration allows for creating complex F-logic rule bases. Thus verifying their correctness and completeness is important. The *debug* plug-in allows to step-by-step evaluate the execution of the reasoning involved in answering a query. The *regression test* facility allows formulating unit tests that can be executed easily to make sure that new rules or updates of existing rules do not break the intended behaviour of the overall knowledge base.

Plugins:

- `com.Ontoprise.ontostudio.debug`
- `com.Ontoprise.ontostudio.query.regressiontest`

#### 4.2.3 F-logic Querying

The *query* plug-in provides a form based query builder that enables users to create F-logic queries that follow a given ontology, i.e. it shields users from the F-logic syntax and interacts with them solely on the conceptual level.

Plugins:

- `com.Ontoprise.ontostudio.query`

#### 4.2.4 Visualizing and Printing F-logic ontologies

This plug-in displays the taxonomy of F-logic ontologies as a two-dimensional graph or tree. This graph can be printed or saved as PDF.

Plugins:

- `com.Ontoprise.ontostudio.graphviz`

#### 4.2.5 Graphical Mapping of F-logic ontologies

This set of plugins provides a graphical tool for aligning and mapping F-logic ontologies. The user can create mapping rules which will be formalized as F-logic and thus are executable at run time.

Plugins:

- `com.Ontoprise.ontostudio.ontomap`
- `com.Ontoprise.ontostudio.ontomap.mathtransform`
- `com.Ontoprise.datamodel.objectmodel.api`

#### 4.2.6 Support for Optimizing F-logic rule bases

This plug-in is more an OntoBroker plug-in than a NeOn Toolkit plug-in. It supports users in enhancing the performance of F-logic rule bases with a genetic optimizer algorithm.

Plugins:

- `com.Ontoprise.ontostudio.genetics`

#### 4.2.7 Online documentation for F-logic and OntoBroker

The following plugins also are essentially OntoBroker plugins. They provide an on-line manual for the F-logic language and for the OntoBroker reasoning system that serves as datamodel and reasoner for the NeOn Toolkit.

Plugins:

- `com.Ontoprise.flogic.help`
- `com.Ontoprise.ontobroker.help`

#### 4.2.8 Lifting of Relational Data Bases

This plug-in supports users to reuse content from relational data bases. It works by translating the schema information from a data base, i.e. the tables, columns, keys, datatypes etc., into an F-logic ontology. This lifting is realized by formal F-logic rules that can actually retrieve data from the data bases at query time. In this way ontology-based applications can be built that work on real data from an RDBMS.

Plugins:

- `com.Ontoprise.ontostudio.dbschemaimport`

#### 4.2.9 Populating F-logic Ontologies from Non-Ontological Sources

Companies utilize a number of data and information stores, e.g. file servers, Excel spreadsheets, UML diagrams, or full text indexers often contain valuable information that should be made available to certain ontology-based applications. These plugins can access the different stores and *lift* them to an ontological level. Either at lifting time or at query-time facts are created to populate the ontology. These facts can be used to answer F-logic queries.

Plugins:

- `com.Ontoprise.ontostudio.imports`
- `com.Ontoprise.ontostudio.imports.collocation`
- `com.Ontoprise.ontostudio.imports.filesystem`
- `com.Ontoprise.ontostudio.remoteob`
- `com.Ontoprise.ontostudio.xlsannotation`
- `com.Ontoprise.ontostudio.uml`

#### 4.2.10 RDF-Schema Modelling

Starting with this release of the NeOn Toolkit we deliver the plug-in for modelling RDF-Schema ontologies. This completes the set of relevant ontology languages for the Semantic Web (in addition to F-logic and OWL).

Plugins:

- `com.Ontoprise.ontostudio.rdf`

#### 4.2.11 WebService Support

The following set of plugins serves two purposes. Firstly, it provides means to lift web services on the ontological level (similar to the DBSchema lifting). Secondly, it creates a web-service wrapper around F-logic queries (against arbitrary (external) sources) and, thus, makes knowledge bases available to the world in a controlled way using open standard interfaces.

Plugins:

- `com.Ontoprise.ontostudio.imports.wsdl`
- `com.Ontoprise.ontostudio.webservices`
- `com.Ontoprise.ontostudio.webservices.generated`
- `com.Ontoprise.ontostudio.webservices.query`
- `com.Ontoprise.ontostudio.webservices.query.logview`

## 5 How to Obtain the NeOn Toolkit

Currently the NeOn Toolkit is available in three different versions:

- the source code version for developers
- a basic configuration for users
- an extended configuration for users

The source code is provided via the source code management platform Ontoware for developers. Users can retrieve the executable versions via the NeOn-Toolkit.org web site.

### 5.1 The Source Code Version

#### 5.1.1 Source Code Management

The source-code of the NeOn Toolkit is hosted on the Ontoware software repository at

<http://Ontoware.org/projects/neon-toolkit/>

The repository contains two packages that together constitute the source-code version of the NeOn Toolkit. The `neon-toolkit.sources` package contains the NeOn components that essentially represent the user interface of the NeOn Toolkit as Eclipse plugins. These plugins fall under the open-source EPL (cf. Section 9.1).

In order to work properly these plugins depend on some other plugins that represent the ontology management, storage, and reasoning functionality. Developers who want to implement their own functionality into the NeOn Toolkit and thus have compile-time dependencies should build ontop of these datamodel plugins *and* the NeOn Toolkit open-source plugins. This functionality is provided in the form of a number of freeware plugins (cf. Section 9.2).

#### 5.1.2 Contributions to the NeOn Toolkit Development

We envision and support two ways for contributing to the NeOn Toolkit development. Firstly, users and developers are invited to provide feedback, bug-reports and patches. This can be done via the NeOn Toolkit portal at <http://www.neon-toolkit.org/>, which hosts mailing-lists, a forum and a Bugzilla installation for issue-tracking. Secondly, all developers (including NeOn Partners) can implement their own functionality, e.g. scientific achievements, as plugins for the Toolkit. These plugins can be uploaded to the NeOn Toolkit portal or to Ontoware, and thus, made available to the public.

Currently, we restrict direct, unsupervised contributions to the open-source plugins, which constitute the basis of the NeOn Toolkit. That means, that the source code uploaded to Ontoware is essentially “read-only”. Contributions are welcome but must not be committed to the repository directly but will be scrutinized by the core development team. This, of course, does not prevent any development of plugins or third party contributions.

#### 5.1.3 Prerequisites for Using and Extending the Source code

In order to build and extend the NeOn Toolkit developers must install JDK 1.5. Since Eclipse is a *development* environment make sure that the *development kit* of Java is installed and not only the runtime environment (JRE). Make sure the environment variables are properly set to refer to the JDK’s bin folder rather than the JRE’s bin folders.

Developers must obtain copies of the basic Eclipse development environment version 3.3.2 and the Graphical Editing Framework (GEF). The Eclipse IDE can be found at

<http://www.eclipse.org/downloads/>.

The GEF provides a framework for developing graphical editors and visualizations. It can be downloaded from

<http://download.eclipse.org/tools/gef/downloads/>.

Alternatively the feature update mechanism of the Eclipse IDE can be used. It is important that the version of GEF is compatible with the Eclipse platform version 3.3. It is recommended to use GEF version 3.3.2.

For setting up the IDE, install Eclipse 3.3 and copy the GEF files into the proper folders ([plugins] and [features]) in the installation folder of Eclipse 3.3. All other required libraries are contained in the [lib]-folders of the plugins shipped and need not be installed separately.

The software deliverable contains three more files:

- ontoconfig.prp
- log4j.properties
- neon.key.xml

These files contain required configuration information. They must be copied into the root folder of your Eclipse installation. In Eclipse speak it is `${eclipse_home}`.

## 5.2 The Executable Versions of the NeOn Toolkit

For the community of users we provide a multitude of different executables for the NeOn Toolkit. All can be found in the download section of the NeOn Toolkit portal:

<http://www.neon-toolkit.org/>

We provide self-extracting installers for Windows that installs the toolkit with a single mouse-click, and compressed zip files containing the toolkit for Windows, Linux and MacOS.

As stated above, there are two different configurations of the executable versions of the NeOn Toolkit. The basic configuration (cf. Section 4.1) is released under the open-source license and can be freely used by anyone. It supports only a subset of the functionality offered by the extended configuration (cf. Section 4.2). Since the extended configuration contains some commercial plugins developed by Ontoprise it is released under a proprietary evaluation license which expires after a period of three months.



## 6 Future Steps for the Core NeOn Toolkit

The current build of the NeOn Toolkit is based on OntoStudio 2.1 which was released in September. Early next year we expect another bug fix update to the NeOn Toolkit, which then will be the last version that is based on the OntoStudio 2.1 release branch.

Ontoprise currently maintains the 2.1 branch only but invests most of its *new* developments on the forthcoming version 2.3 of OntoStudio. This next major release of OntoStudio is planned for May 2009.

During the first months of 2009 the Ontoprise will migrate the OWL editor, which currently only exists on the 2.1 release branch, to the new version. This will prepare the needed base functionality for the NeOn community as a new basic release of the NeOn Toolkit shortly after OntoStudio 2.3 is released. This release will be numbered NeOn Toolkit 1.3.

For this new release a couple of API modifications, primarily on the Eclipse-based GUI level are expected that the developer community including the NeOn partners will need some time and effort to migrate the implemented features and plugins to the new platform.

The build process will greatly improve for the version 1.3. We plan to implement a so-called headless Eclipse build, which synchronized the development and build environment greatly. In this process we will also package the plugins of the basic and extended NeOn Toolkit into *Eclipse features*, which simplifies the task of installing, updating, and also publishing the components individually. We expect a feature structure similar to the one used to present the extended plugins in Section 4.2.

## 7 Conclusion

This deliverable mainly consists of the third iteration of the NeOn Toolkit (V1.2). It now contains complete modelling facilities for OWL-DL ontologies. The NeOn Toolkit is released in three different versions: a source-code version for developers, a basic and an extended configuration for users. The different versions are released under different software licenses, which range from the open-source Eclipse Public License, over a free-ware license to a proprietary evaluation license for commercial plugins.

This version of the toolkit provides a solid basis for the plugins developed by the scientific partners in the NeOn project. These plugins were mainly geared towards OWL modelling like e.g. the Radon plug-in that checks ontologies for consistency and provides suggestions for repairing them.

In this document we presented an overview of the new OWL modelling features, and the different versions of the toolkit with associated license agreements. For the future we foresee that we gather feedback from the NeOn user community and the NeOn project partners with respect to usage of the OWL modelling feature to locate and fix bugs, to enhance the usability and to add missing features; overall, to make it more mature.

## 8 Reference

- Kifer et al. 1995 Michael Kifer, Georg Lausen, James Wu: Logical Foundations of Object-Oriented and Frame-Based Languages. Journal ACM 42(4): 741-843 (1995)
- NeOn D6.2.1 Walter Waterfeld, Moritz Weiten, Peter Haase: Deliverable D6.2.1: Specification of NeOn reference architecture and NeOn APIs. NeOn Project Deliverable, 2007.
- NeOn D6.4.1 W. Waterfeld, R. Palma, P. Haase: D6.4.1 Realisation & early evaluation of NeOn service-oriented registry repository, NeOn Project Deliverable 2007
- NeOn D6.6.1 Michael Erdmann, Dirk Wenke: D6.6.1 Realisation & early evaluation of basic NeOn tools in NeOn Toolkit V1. NeOn Deliverable, August 2007.
- NeOn D6.6.2 Michael Erdmann, M. Becker, W. Hihn, R. Wroblenski: D6.6.2 NeOn Toolkit Documentation, NeOn Project Deliverable 2008.
- NeOn D6.7.1 Michael Erdmann, Thomas Hemp: Deliverable D6.7.1: Beta release of the Core NeOn Online Documentation. NeOn Project Deliverable, 2007.
- NeOn D6.10.1 Peter Haase et al. NeOn Deliverable D6.10.1 Realization of core engineering components for the NeOn Toolkit. February 2008
- Ontoprise2007a Ontoprise GmbH: OntoBroker User Guide V5.0. Ontoprise GmbH, Karlsruhe 2007.  
[http://www.Ontoprise.com/content/e799/e893/e938/e954/e956/Use\\_rGuide\\_OntoBroker\\_5.0\\_eng.pdf](http://www.Ontoprise.com/content/e799/e893/e938/e954/e956/Use_rGuide_OntoBroker_5.0_eng.pdf)

## 9 Appendix – License Texts

### 9.1 EPL

#### Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

#### 1. DEFINITIONS

"Contribution" means:

- a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and
- b) in the case of each subsequent Contributor:

- i) changes to the Program, and

- ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf.

Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

#### 2. GRANT OF RIGHTS

- a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

- b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

- c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.
- d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

### 3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
- i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
  - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
  - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
  - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

### 4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the

extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

## **5. NO WARRANTY**

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

## **6. DISCLAIMER OF LIABILITY**

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## **7. GENERAL**

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably

practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

## 9.2 Freeware License

### Freeware License Agreement

Ontoprise GmbH, An der RaumFabrik 29, 76227 Karlsruhe, Germany (“Ontoprise”) hereby grants you a non-exclusive, royalty-free and non-transferable license to use this software in binary form only and to use the related documentation in connection with your use of this software.

You shall not (1) sell, rent, lease or sublicense rights in this software, (2) remove or alter any trademark, logo, copyright or other proprietary notices, legends, symbols or labels, (3) use this software for purposes of application service providing (ASP) or comparable uses. You shall not use the software in a manner which is disparaging to Ontoprise. You shall not reverse engineer, decompile or disassemble the software except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

As a condition to using this software in connection with any other software, you shall display on the screen of such other software a prominent notice on the proprietorship of the rights. This notice shall read as follows:

Copyright (c) 2008 Ontoprise GmbH. All Rights Reserved.

The software being licensed in accordance with the present freeware license agreement might contain components that are being licensed under an “open source license”. Ontoprise is not the licensor of such software and such software is not being licensed under the present freeware license agreement. The licensor in such a case is the respective owner of the respective rights for the licensing as open source software.

[licenses-3rdparty] contains a list of components of the licensed software which are licensed under an open source license and the texts of such licenses.

This software is provided "as is", without a warranty of any kind. All express or implied conditions, representations and warranties are hereby excluded. Ontoprise will not be liable for any damages suffered by you as a result of using the software. In no event will Ontoprise be liable for any lost revenue, profit or data, or for indirect, special, consequential, incidental or punitive damages.

This freeware license agreement is subject to German law, excluding is conflicts of law provisions. The courts in Karlsruhe, Germany shall have exclusive jurisdiction for all disputes arising out of or in connection with this freeware license agreement.

Copyright (c) 2008 Ontoprise GmbH. All Rights Reserved.



## 9.3 Evaluation License

### Evaluation License

Ontoprise GmbH, An der RaumFabrik 29, 76227 Karlsruhe, Germany (“Ontoprise”) hereby grants you a non-exclusive, royalty-free and non-transferable evaluation license to use this software in binary form only and to use the related documentation in connection with your use of this software.

This evaluation license is granted for a period of three (3) months after the earlier of i) your first download of the software or ii) the time you obtain a copy of this software in any other way. After this three (3) month period, the software will cease to function properly. If you wish to continue to use the software, you must contact Ontoprise to obtain a full license of the software.

You shall not (1) sell, rent, lease or sublicense rights in this software, (2) remove or alter any trademark, logo, copyright or other proprietary notices, legends, symbols or labels, (3) use this software for purposes of application service providing (ASP) or comparable uses. You shall not use the software in a manner which is disparaging to Ontoprise. You shall not reverse engineer, decompile or disassemble the software except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

As a condition to using this software in connection with any other software, you shall display on the screen of such other software a prominent notice on the proprietorship of the rights. This notice shall read as follows:

Copyright (c) 2008 Ontoprise GmbH. All Rights Reserved.

The software being licensed in accordance with the present evaluation license agreement might contain components that are being licensed under an “open source license”. Ontoprise is not the licensor of such software and such software is not being licensed under the present license agreement. The licensor in such a case is the respective owner of the respective rights for the licensing as open source software.

[licenses-3rdparty] contains a list of components of the licensed software which are licensed under an open source license and the texts of such licenses.

This software is provided "as is", without a warranty of any kind. All express or implied conditions, representations and warranties are hereby excluded. Ontoprise will not be liable for any damages suffered by you as a result of using the software. In no event will Ontoprise be liable for any lost revenue, profit or data, or for indirect, special, consequential, incidental or punitive damages.

This evaluation license agreement is subject to German law, excluding is conflicts of law provisions. The courts in Karlsruhe, Germany shall have exclusive jurisdiction for all disputes arising out of or in connection with this license agreement.

Copyright (c) 2008 Ontoprise GmbH. All Rights Reserved.