NeOn: Lifecycle Support for Networked Ontologies

Integrated Project (IST-2005-027595)

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

D6.7.3 Update and next release of the core NeOn Toolkit with new OWL API realisation

Deliverable Co-ordinator: **Michael Erdmann**

Deliverable Co-ordinating Institution: **Ontoprise GmbH (ONTO)**

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. Version V2.3 is now based on the OWL API to better support ongoing updates to the OWL language specification defined by the W3C OWL working group.

Document Identifier:	NEON/2010/D6.7.3/v1.0	Date due:	October 31st, 2009
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	January 31st, 2010
Project start date:	March 1, 2006	Version:	v1.0
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:

Open University (OU) – Coordinator Knowledge Media Institute – KMi Berrill Building, Walton Hall Milton Keynes, MK7 6AA United Kingdom Contact person: Enrico Motta E-mail address: e.motta@open.ac.uk	Universität Karlsruhe – TH (UKARL) Institut für Angewandte Informatik und Formale Beschreibungsverfahren – AIFB Englerstrasse 11 D-76128 Karlsruhe, Germany Contact person: Andreas Harth E-mail address: aha@aifb.uni-karlsruhe.de
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	Software AG (SAG) Uhlandstrasse 12 64297 Darmstadt Germany Contact person: Walter Waterfeld E-mail address: walter.waterfeld@softwareag.com
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	Institut 'Jožef Stefan' (JSI) Jamova 39 SI-1000 Ljubljana Slovenia Contact person: Marko Grobelnik E-mail address: marko.grobelnik@ijs.si
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	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
Universität Koblenz-Landau (UKO-LD) Universitätsstrasse 1 56070 Koblenz Germany Contact person: Steffen Staab E-mail address: staab@uni-koblenz.de	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
Ontoprise GmbH. (ONTO) Amalienbadstr. 36 (Raumfabrik 29) 76227 Karlsruhe Germany Contact person: Jürgen Angele E-mail address: angele@ontoprise.de	Food and Agriculture Organization of the United Nations (FAO) Viale delle Terme di Caracalla 1 00100 Rome Italy Contact person: Caterina Caracciolo E-mail address: Caterina.Caracciolo@fao.org
Atos Origin S.A. (ATOS) Calle de Albarracín, 25 28037 Madrid, Spain Contact person: Tomás Pariente Lobo E-mail address: tomas.parientelobo@atosorigin.com	Laboratorios KIN, S.A. (KIN) C/Ciudad de Granada, 123 08018 Barcelona, Spain Contact person: Antonio López E-mail address: alopez@kin.es

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 to its composition:

Ontoprise, SAG, UKARL, OU

Change Log

Version	Date	Amended by	Changes
1	2009-09-01	Michael Erdmann	Initial version and structure of deliverable document
2	2009-09-10	Michael Erdmann	Migration of content of previous document
3	2009-12-10	Michael Erdmann	Initialization of OWL2 feature table
4	2009-12-15	Michael Erdmann	OWL2 feature table
5	2009-12-16	Michael Erdmann	Download and Licence model
6	2009-12-16	Michael Erdmann	Intro, Conclusion, Exec Summary
7	2009-12-16	Michael Erdmann	Ready for review
8	2010-01-13	Michael Erdmann	Amendments after review comments

Executive Summary

This deliverable consists of two parts, (i) the software representing the third iteration of the NeOn Toolkit (V2.3) and (ii) this accompanying document. The NeOn Toolkit V2.3 now contains complete modelling facilities for OWL ontologies. Its internal datamodel has been re-implemented on the basis of the OWL API to better support the OWL language and especially the ongoing developments of OWL2.

The NeOn Toolkit v2.3 was released in December 2009 and is available in the form of source-code for developers and as executable programs for users. We support Windows, MacOS and Linux machines. The basic version now is completely open-source and is released under the Eclipse Public License.

Table of Contents

1 Introduction

This deliverable presents the third iteration of the NeOn Toolkit. Its predecessors D6.6.1 “Realisation & early evaluation of basic NeOn tools in NeOn toolkit V1” [NeOn D6.6.1] and D6.7.2 “Update of the core NeOn toolkit” [NeOn D6.7.2] presented the version v1.0 and the advancements of the NeOn Toolkit release v1.2, respectively.

Since the first release more than two years ago, the NeOn Toolkit has evolved significantly and now, better than ever, supports ontology engineers in handling networked ontologies. During the same time period the W3C OWL working group has continuously extended and redefined the update of the web ontology language, initially dubbed OWL 1.1 and now finally released as OWL2[1]. OWL2 has become a W3C recommendation in late October 2009 [Motik et al. 2009].

In order to support the new language features and also ensure compatibility with other Semantic Web software and future developments we decided to build the OWL features of the new NeOn Toolkit on-top of the (Manchester) OWL API[2]. The OWL API [Horridge, Bechhofer 2009] is available under the open source LGPL licence and has emerged as the de facto standard for implementing OWL based applications. It has an active user community and promises a high degree of standard compliance. It is the reference implementation of OWL2 recommended by the OWL working group.

In the remainder of the document we will describe the implemented OWL modelling features of the NeOn Toolkit and put this into the context of the OWL2 standard specification [Bao et al 2009]. Then we will present the licence model of the NeOn Toolkit and its components and we will describe how users can obtain executable versions of the toolkit and developers can access the source code.

[1] http://www.w3.org/2007/OWL

[2] http://owlapi.sourcrforge.net

2 The OWL Modelling Facilities of the NeOn Toolkit

2.1 Notational Conventions

The OWL2 Working group has published a quick reference guide [Bao et al. 2009] summarizing all language features[3]. In this reference guide the authors use a number of predefined namespaces and some place holders for certain kinds of entities or entity names. In the following table we replicate the used notational conventions:

Prefix Name	Expansion
rdf:	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs:	http://www.w3.org/2000/01/rdf-schema#
owl:	http://www.w3.org/2002/07/owl#
xsd:	http://www.w3.org/2001/XMLSchema#

Prefixes for standard namespaces.

Letters	Meaning
C	class expression
CN	class name
D	data range
DN	datatype name
P	object property expression
PN	object property name
R	data property
A	annotation property
a	individual
aN	individual name
_:a	anonymous individual (a blank node label)
v	literal
n	non-negative integer
f	facet
ON	ontology name
U	IRI
s	IRI or anonymous individual
t	IRI, anonymous individual, or literal
p	prefix name
_:x	blank node
(a1 … an)	RDF list

Variable names and their meaning

[3] cf. http://www.w3.org/TR/owl2-quick-reference/

2.2 Supported and Unsupported OWL2 Features

In this section we will give an overview of the supported OWL2 features (including OWL1, cf. [Golbreich, Wallace 2009] for an overview and rationale for the new, additional OWL2 features). We will use the same structuring as chosen in the OWL2 reference guide [Bao et al. 2009]. The following tables present the following information:

- The actual OWL2 feature
- Functional style syntax of the feature according to the OWL2 recommendation
- Syntax for the feature as used in the NeOn Toolkit. This is based on the Manchester Syntax for OWL (if applicable)
- Since the NeOn Toolkit currently does not support all OWL2 features on the GUI level, column four shows whether the feature is supported by the GUI of the NeOn Toolkit (yes/no). Nevertheless, even if this column contains a “no”, the NeOn Toolkit is capable of loading, manipulating and storing ontologies that use these features, because the NeOn Toolkit is using the OWL API datamodel, parsers and serializers, which is the reference-implementation of the OWL2 recommendation.

2.2.1 Class Expression

The first group of features represents the different ways to specify complex OWL classes.

	Functional Syntax	**NeOn Toolkit Syntax**	**Supported by GUI of NeOn Toolkit 2.3**
2.1 Class Expressions			
Predefined and named classes			
named class	C	C	yes
universal class	owl:Thing	owl:Thing	yes
empty class	owl:Nothing	owl:Nothing	yes
Boolean Connectives and Enumeration of Individuals			
intersection	ObjectIntersectionOf(C1…Cn)	C1 and … Cn	yes
union	ObjectUnionOf(C1…Cn)	C1 or … Cn	yes
complement	ObjectComplementOf(C)	not C	yes
enumeration	ObjectOneOf(a1 … an)	{a1, … an}	yes
Object Property Restrictions			
universal	ObjectAllValuesFrom (P C)	P only C	yes
existential	ObjectSomeValuesFrom(P C)	P some C	yes
individual value	ObjectHasValue(P a)	P value a	yes
local reflexivity	ObjectHasSelf(P)	P Self	yes
exact cardinality	ObjectExactCardinality(n P)	P exactly n owl:Thing	yes

qualified exact cardinality	ObjectExactCardinality(n P C)	P exactly n C	yes
maximum cardinality	ObjectMaxCardinality(n P)	P max n owl:Thing	yes
qualified maximum cardinality	ObjectMaxCardinality(n P C)	P max n C	yes
minimum cardinality	ObjectMinCardinality(n P)	P min n owl:Thing	yes
qualified minimum cardinality	ObjectMinCardinality(n P C)	P min n C	yes
Data Property Restrictions			
universal	DataAllValuesFrom (P D)	P only D	yes
existential	DataSomeValuesFrom(P D)	P some D	yes
literal value	DataHasValue(P a)	P value a	yes
exact cardinality	DataExactCardinality(n P)	P exactly n	yes
qualified exact cardinality	DataExactCardinality(n P D)	P exactly n D	yes
maximum cardinality	DataMaxCardinality(n P)	P max n	yes
qualified maximum cardinality	DataMaxCardinality(n P D)	P max n D	yes
minimum cardinality	DataMinCardinality(n P)	P min n	yes
qualified minimum cardinality	DataMinCardinality(n P D)	P min n D	yes
Restrictions Using n-ary Data Range			
n-ary universal	DataAllValuesFrom (R1 … Rn Dn)		no
n-ary existential	DataSomeValuesFrom (R1 … Rn Dn)		no

As can be seen, the NeOn Toolkit supports nearly all complex class expressions, with the exception of n-ary data ranges. It also supports the new OWL2 expressions like qualified number restrictions, local reflexivity (Self) or the new built-in class owl:Nothing.

2.2.2 Properties

In the second group we will present all features representing property expressions.

	Functional Syntax	**NeOn Toolkit Syntax**	**Supported by GUI of NeOn Toolkit 2.3**
2.2 Properties			
Object Property Expressions			
named object property	PN	PN	yes
universal object property	owl:topObjectProperty	owl:topObjectProperty	yes
empty object property	owl:bottomObjectProperty	owl:bottomObjectProperty	yes
inverse property	ObjectInverseOf(PN)		no
Data Property Expressions			
named Data property	R	R	yes
universal Data property	owl:topDataProperty	owl:topDataProperty	yes
empty Data property	owl:bottomDataProperty	owl:bottomDataProperty	yes

The NeOn Toolkit supports all property expressions, with the exception of inverse object properties. Users can specify inverse properties using axioms but currently cannot use them in class expressions. Nevertheless, loading and storing of ontologies that make use of *ObjectInverseOf* expressions, is supported by the NeOn Toolkit via the OWL API datamodel, parsers and serializers. All new top and bottom properties introduced by OWL2 are supported by the NeOn Toolkit.

2.2.3 Individuals and Literals

The third group describes basic individuals or datatype values (literals).

	Functional Syntax	**NeOn Toolkit Syntax**	**Supported by GUI of NeOn Toolkit 2.3**
2.3 Indivduals and Literals			
named individual	aN	aN	yes
anonymous individual	_:a		no
literal (dataype value)	"abc"^^DN	"abc"^^DN	yes

With the exception of the anonymous individuals we support all individuals and literals within the text fields of the NeOn Toolkit. The underlying datamodel does support anonymous individuals, thus the NeOn Toolkit can handle such ontologies.

2.2.4 Data Ranges

Group number four describes (complex) datatypes, which are called data ranges in OWL2.

	Functional Syntax	**NeOn Toolkit Syntax**	**Supported by GUI of NeOn Toolkit 2.3**
2.4 Data Ranges			
Data Range Expressions			
named datatype	DN	DN	Yes
data range complement	DataComplementOf (D)		No
data range intersection	DataIntersectionOf (D1…Dn)		No
data range union	DataUnionOf (D1…Dn)		No
literal enumeration	DataOneOf (v1 … vn)	{v1, … vn}	Yes
datatype restriction	DatatypeRestriction (DN f1 v1 … fn vn)		No

With respect to complex data ranges the Neon Toolkit only supports named data types and enumerations. The new OWL2 expressions to construct new data ranges, e.g. via facets, is not yet supported by the GUI.

2.2.5 Class Axioms

The next group starts a long list of axioms which can be asserted in OWL2. In the NeOn Toolkit, axioms are accessible through different GUI mechanisms instead of being represented in pure textual form. Because of this, from now on, we leave out the NeOn Toolkit syntax column.

Class axioms make statements about the relationship between OWL classes.

	Functional Syntax	**Supported by GUI of NeOn Toolkit 2.3**
2.5 Axioms		
Class Expression Axioms		
subclass	SubClassOf(C1 C2)	Yes
equivalent classes	EquivalentClasses (C1 … Cn)	Yes
disjoint classes	DisjointClasses(C1 C2)	Yes
pairwise disjoint classes	DisjointClasses (C1 … Cn)	Yes
disjoint union	DisjointUnionOf (CN C1 … Cn)	No

Except for the new OWL2 feature of specifying a class as a disjoint union of its subclasses, the NeOn Toolkit supports all class axioms. They can be entered via the entity property view for classes in separate sections or tabs (see below).

2.2.6 Object Property Axioms

In the following group we list a rather extensive set of different axioms that specify the semantics of object properties OWL2.

	Functional Syntax	**Supported by GUI of NeOn Toolkit 2.3**
Object Property Axioms		
subproperty	SubObjectPropertyOf (P1 P2)	Yes
property chain inclusion	SubObjectPropertyOf (ObjectPropertyChain (P1 … Pn) P)	Yes
property domain	ObjectPropertyDomain (P C)	Yes
property range	ObjectPropertyRange (P C)	Yes
equivalent properties	EquivalentObjectProperties (P1 … Pn)	Yes
disjoint properties	DisjointObjectProperties (P1 P2)	No
pairwise disjoint properties	DisjointObjectProperties (P1 … Pn)	No
inverse properties	InverseObjectProperties (P1 P2)	Yes
functional property	FunctionalObjectProperty (P)	Yes
inverse functional property	InverseFunctionalObjectProperty (P)	Yes
reflexive property	ReflexiveObjectProperty (P)	Yes
irreflexive property	IrreflexiveObjectProperty (P)	Yes
symmetric property	SymmetricObjectProperty (P)	Yes
asymmetric property	AsymmetricObjectProperty (P)	Yes
transitive property	TransitiveObjectProperty (P)	Yes

With the sole except of disjoint properties, the NeOn Toolkit supports all object property axioms, including the new OWL2 features such as property chains, reflexive, irreflexive or asymmetric properties.

2.2.7 Data Property Axioms

The next group shows the data property axioms, which allow us to specify the semantics of data properties OWL2.

	Functional Syntax	**Supported by GUI of NeOn Toolkit 2.3**
Data Property Axioms		
subproperty	SubDataPropertyOf(R1 R2)	Yes
property domain	DataPropertyDomain(R C)	Yes
property range	DataPropertyRange(R D)	Yes
equivalent properties	EquivalentDataProperties (R1 … Rn)	Yes
disjoint properties	DisjointDataProperties (R1 R2)	No
pairwise disjoint properties	DisjointDataProperties (R1 … Rn)	No
functional property	FunctionalDataProperty(R)	Yes

Similarly to object properties above, there is only one OWL2 feature that users of the NeOn Toolkit cannot immediately access via the GUI, namely disjoint data properties.

2.2.8 Datatype Definitions

OWL2 allows us now to define new datatypes by giving data range specifications a new name.

	Functional Syntax	**Supported by GUI of NeOn Toolkit 2.3**
Datatype Definitions		
datatype definition	DatatypeDefinition(DN D)	No

This feature is not yet supported by the NeOn Toolkit because, without being able to specify complex data ranges, this would be meaningless.

2.2.9 Assertions

In this group we collect all A-Box axioms making assertions about individuals.

	Functional Syntax	**Supported by GUI of NeOn Toolkit 2.3**
Assertions		
individual equality	SameIndividual(a1 … an)	Yes
individual inequality	DifferentIndividuals(a1 a2)	Yes
pairwise individual inequality	DifferentIndividuals (a1 … an)	Yes
class assertion	ClassAssertion(C a)	Yes
positive object property assertion	ObjectPropertyAssertion (PN a1 a2)	Yes
positive data property assertion	DataPropertyAssertion (R a v)	Yes
negative object property assertion	NegativeObjectPropertyAssertion (P a1 a2)	No
negative data property assertion	NegativeDataPropertyAssertion (R a v)	No

All essential types of assertions are supported by the NeOn Toolkit. Only negative assertions are not yet implemented in the user interface.

2.2.10 Keys

OWL2 now allows for specifying key properties for classes, which formulates a rule that all individuals of that class with the same values for all key properties are identical.

	Functional Syntax	**Supported by GUI of NeOn Toolkit 2.3**
Keys		
Key	HasKey(C (P1 … Pm) (R1 … Rn))	No

This feature is not implemented in the NeOn Toolkit yet.

2.2.11 Declarations

Entities of all kinds can be declared to exist in OWL2.

	Functional Syntax	Supported by GUI of NeOn Toolkit 2.3
2.6 Declarations		
class	Declaration(Class(CN))	Yes
datatype	Declaration(Datatype(DN))	Yes
object property	Declaration(ObjectProperty(PN))	Yes
data property	Declaration(DataProperty(R))	Yes
annotation property	Declaration (AnnotationProperty(A))	Yes
named individual	Declaration (NamedIndividual(aN))	Yes

Users of the NeOn Toolkit can introduce new instances of all entity types defined in OWL 2.

2.2.12 Annotations

Annotations represent metadata about axioms or entities.

	Functional Syntax	Supported by GUI of NeOn Toolkit 2.3
2.7 Annotations		
Annotations		
annotation assertion	AnnotationAssertion (A s t)	Yes
annotation of an axiom	AXIOM(Annotation (A t) …)	No
annotation of another annotation	Annotation(Annotation (A t) … A1 t1)	No

The OWL2 specification supports extensive use of annotations. Users of the NeOn Toolkit can read, create and modify annotations for all entity types and ontologies. Axiom annotations and annotations of annotations are not yet supported.

2.2.13 Annotation Properties

OWL2 specifies a number of pre-existing annotation properties.

	Functional Syntax	**Supported by GUI of NeOn Toolkit 2.3**
Annotation Properties		
named annotation property	A	Yes
human-readable name	rdfs:label	Yes
human-readable comment	rdfs:comment	Yes
additional information	rdfs:seeAlso	Yes
defining agent	rdfs:isDefinedBy	Yes
version information	owl:versionInfo	Yes
deprecation	owl:deprecated	Yes
backwards compatibility	owl:backwardCompatibleWith	Yes
incompatibility	owl:incompatibleWith	Yes
Prior version	owl:priorVersion	Yes

All predefined annotation properties are supported by the NeOn Toolkit. In addition, user defined annotation properties can be introduced and used.

2.2.14 Annotation Axiom

OWL2 introduces new features for modelling properties of annotation properties.

	Functional Syntax	**Supported by GUI of NeOn Toolkit 2.3**
Annotation Axioms		
annotation subproperties	SubAnnotationPropertyOf(A1 A2)	Yes
annotation property domain	AnnotationPropertyDomain(A U)	No
annotation property range	AnnotationPropertyRange(A U)	No

2.2.15 Ontologies

In the final group we find ontology-level information

	Functional Syntax	**Supported by GUI of NeOn Toolkit 2.3**
2.8 Ontologies		
OWL ontology (importing)	Ontology([ON [U]] Import(ON1)... Annotation(A t) ...)	Yes
prefix declaration	Prefix(p=U)	Yes

In the NeOn Toolkit we support ontology imports and the management of prefixes.

2.3 Summary

In the tables above the W3C document [Bao et al. 2009] lists 105 different features of the OWL2 language. In the NeOn Toolkit GUI we support 85 of them; 14 of them have been added since the previous release v1.2.3 of the NeOn Toolkit. Since the underlying datamodel, the parsers and the serializers are formed by the Manchester OWL API, the reference implementation of OWL2, we support loading, storing and serializing OWL2 models fully.

3 The License Model of the NeOn Toolkit[4]

The NeOn Toolkit is a complex piece of software. It is built on top of the Eclipse platform and one of its goals it 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 plug-ins.

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 following advice given through consultations with 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 complete basic version of the NeOn Toolkit (with all OWL modelling facilities) is released under the open source *Eclipse Public License*.

- The extended configuration of the NeOn Toolkit containing commercial plug-ins by Ontoprise is released under an *evaluation licence*.

The third license, the so called *freeware license*, is not needed anymore because the basic NeOn Toolkit now uses the OWL API as its core datamodel and does not rely on proprietary components for this purpose.

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[5], cf. Section 7.1). This is the same license that all Eclipse plug-ins are distributed with. As we are redistributing a couple of plug-ins from the Eclipse platform with the NeOn Toolkit, the use of EPL is compatible; actually, the Eclipse foundation recommends using EPL for plug-ins.

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. It makes it particularly clear that individual plug-ins can be released under/on 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.

[4] This section is an update to the section with the same title from deliverable D6.7.2.

[5] http://www.eclipse.org/legal/epl-v10.html

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

- com.ontoprise.dependencies
- com.ontoprise.ontostudio.dependencies
- com.ontoprise.ontostudio.owl.gui
- com.ontoprise.ontostudio.owl.gui.ontologyimportsgraph
- com.ontoprise.ontostudio.owl.help
- com.ontoprise.ontostudio.owl.model
- com.ontoprise.ontostudio.owl.visualize
- dependencies
- org.neontoolkit.core
- org.neontoolkit.gui
- org.neontoolkit.io
- org.neontoolkit.jpowergraph
- org.neontoolkit.ontovisualize
- org.neontoolkit.plugin
- org.neontoolkit.refactor
- org.neontoolkit.search
- org.neontoolkit.swt
- org.semanticweb.owl

Note, that the last plug-in in the list simply contains the Manchester OWL API wrapped into a plug-in to make it available to the core plug-ins of the NeOn Toolkit and to all other plug-ins that extend the basic toolkit.

3.2 Evaluation License

When building the (extended configuration of the) NeOn Toolkit we include a couple of commercial plug-ins by ontoprise (e.g. graphically modelling rules or mappings). These plug-ins are released under an *evaluation licence* (cf. Section 7.2). This license is not an *open-source* license but closer in spirit to an end-user-license agreement (EULA). The license especially prohibits decompiling and changing the code. In a similar manner to the EPL, it 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 evaluation license contains 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 technically makes sure that the extended plug-ins 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.

4 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 for developers via the source code management services offered by Source Forge. Users can retrieve the *executable* versions via the NeOn-Toolkit.org web site.

4.1 The Source Code Version

4.1.1 Source Code Management

The source-code of the NeOn Toolkit is hosted on the Source Forge software repository at:

http://sourceforge.net/projects/neon-toolkit/

The site hosts the repository on a subversion server:

http://neon-toolkit.svn.sourceforge.net/viewvc/neon-toolkit/trunk/

This can then be accessed by any subversion client as well as from within integrated development environments, like Eclipse. The full content of this neon-toolkit SVN repository falls under the open-source EPL (cf. Section 7.1).

4.1.2 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[6] for issue-tracking. Secondly, all developers (including NeOn Partners) can implement their own functionality, e.g. scientific achievements, as plug-ins for the Toolkit. These plug-ins can be uploaded to the NeOn Toolkit portal or to open source code management systems such as Source Forge or Google Code.

Currently, we restrict direct, unsupervised contributions to the open-source plug-ins on the NeOn Toolkit SVN. This means only developers of selected members of the NeOn consortium can directly contribute to the code base. For all others, the source code uploaded on Source Forge is essentially “read-only”. Contributions are welcome but must not be committed to the repository directly and will be scrutinized by the core development team. This, of course, does not prevent development of plug-ins or third party contributions.

[6] http://www.neon-toolkit.org/bugzilla

4.1.3 Prerequisites for Using and Extending the Source Code

In order to build and extend the NeOn Toolkit, developers must install JDK 1.6. 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 3.5. The Eclipse IDE can be found at:

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

For former versions we required the Graphical Editing Framework (GEF). But for the current release 2.3 we do not require it anymore.

It would be helpful to install a subversion client into your Eclipse environment, e.g. Subclipse[7]. Pointing the SVN client to the above mentioned repository location and checking out the plug-ins is all that is needed to start developing plug-ins for the NeOn Toolkit.

As a central managing point for NeOn Toolkit plug-ins developed within the NeOn project, we are using the source-code management system at:

https://neon-plugins.googlecode.com/svn/trunk/

Deliverable D6.10.3 presents an overview over the multitude of different plug-ins already available. More information for developers can be found at:

´ http://neon-toolkit.org/wiki/Developer_Corner

This is the developers section of the NeOn Toolkit portal.

4.2 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/wiki/Download

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. The basic configuration (restricted to modelling OWL ontologies) is released under the open source EPL licence and can be freely used by anyone. The extended configuration contains some commercial plug-ins developed by Ontoprise and is released under a proprietary evaluation license which expires after a period of three months.

[7] http://subclipse.tigris.org/

5 Conclusion

This deliverable mainly consists of the third iteration of the NeOn Toolkit (V2.3). It now contains complete modelling facilities for OWL ontologies and fully supports loading, storing and serializing OWL-2 models. The graphical user interface of the NeOn Toolkit supports most OWL2 features, with the exception mentioned in Section 2. In the same section we presented an overview of all the OWL2 modelling features that are supported.

Since the previous release of the NeOn Toolkit approximately one year ago, we have decided to re-implement the underlying datamodel using the Manchester OWL API [Horridge, Bechhofer 2009]. Although this has been a major change and a lot of effort had to be spent on this task, in the end it was worthwhile. We can now immediately benefit from bug fixes and improvements of the reference implementation of OWL2. Although the OWL API has still not been officially released yet (we are using the trunk version of the OWL API SVN, v3.0.0), and some incompatible changes during the last months had cost us additional integration work, we think that the benefits outweigh the costs.

Given the popularity of the OWL API, which is the de facto standard for implementing OWL based applications, we also expect that uptake of the NeOn Toolkit as a target platform for developers will increase in the future.

6 Reference

Bao et al 2009 — Jie Bao, Elisa F. Kendall, Deborah L. McGuinness, Peter F. Patel-Schneider: OWL 2 Web Ontology Language Quick Reference Guide. W3C Recommendation 27 October 2009. http://www.w3.org/TR/2009/REC-owl2-quick-reference-20091027/

Golbreich, Wallace 2009 — Christine Golbreich, Evan K. Wallace (ed.): OWL 2 Web Ontology Language. New Features and Rationale. W3C Recommendation 27 October 2009. http://www.w3.org/TR/2009/REC-owl2-new-features-20091027/

Horridge, Bechhofer 2009 — Matthew Horridge, Sean Bechhofer: The OWL API: A Java API for Working with OWL 2 Ontologies. In proceedings of OWLED 2009 - OWL: Experiences and Directions. Sixth International Workshop. Chantilly, Virginia, USA 23-24 October 2009, Co-located with ISWC 2009 http://www.webont.org/owled/2009/papers/owled2009_submission_29.pdf

Motik et al 2009 — Boris Motik, Peter F. Patel-Schneider, Bijan Parsia: OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. W3C Recommendation 27 October 2009 http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/

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 D6.6.3 — NeOn Toolkit documentation v3. NeOn Project Deliverable 2009 (*forthcoming*)

NeOn D6.7.2 — Michael Erdmann: D6.7.2 Update of the core NeOn Toolkit. NeOn Project Deliverable 2008

NeOn D6.10.3 — Updated NeOn Toolkit plugins v3. NeOn Project Deliverable 2009 (*forthcoming*)

7 Appendix – License Texts

7.1 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.

7.2 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.