NeOn: Lifecycle Support for Networked Ontologies

Integrated Project (IST-2005-027595)

Priority: IST-2004-2.4.7 — "Semantic-based knowledge and content systems"

D6.8.3 Testing the NeOn Toolkit interoperability

Deliverable Author: Raúl García-Castro

Deliverable Authoring Institution: Universidad Politécnica de Madrid (UPM)

This deliverable includes the analysis of the results of the third iteration of benchmarking the interoperability of the NeOn Toolkit, as well as the recommendations extracted from this analysis for improving the NeOn Toolkit.

Document Identifier:	NEON/2010/D6.8.3/v1.2	Date due:	January 31, 2010
Class Deliverable:	NEON EU-IST-2005-027595	Submission date:	January 31, 2010
Project start date	March 1, 2006	Version:	v1.2
Project duration:	4 years	State:	Final version
		Distribution:	Public

NeOn Consortium

This document is part of the NeOn research project funded by the IST Programme of the Commission of the European Communities, 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: Martin Dzbor, Enrico Motta E-mail address: {m.dzbor, e.motta}@open.ac.uk	**Universität Karlsruhe – TH (UKARL)** Institut für Angewandte Informatik und Formale Beschreibungsverfahren – AIFB D-76128 Karlsruhe Germany Contact person: Peter Haase E-mail address: pha@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.ump.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 adress: jcontreras@isoco.com	**Institut 'Jožef Stefan' (JSI)** Jamova 39 SL–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 – 665 avenue de l'Europe Montbonnot Saint Martin 38334 Saint-Ismier France Contact person: Jérôme Euzenat	**University of Sheffield (USFD)** Dept. of Computer Science Regent Court 211 Portobello street S14DP Sheffield United Kingdom Contact person: Hamish Cunningham
Universität Kolenz-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. Marino 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 00100 Rome Italy Contact person: Marta Iglesias E-mail address: marta.iglesias@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 the writing of this document:

- Universidad Politécnica de Madrid
- Ontoprise

Change Log

Version	Date	Amended by	Changes
0.1	28-12-2009	Raúl García-Castro	First draft of the document
0.2	30-12-2009	Raúl García-Castro	Inserted OWL compliance results
0.3	5-01-2010	Raúl García-Castro	Inserted OWL interoperability results, evolution of results, and recommendations
1.0	07-01-2010	Raúl García-Castro	First version
1.1	20-01-2010	Raúl García-Castro	Implemented the comments from the quality reviewer (Michael Erdmann)
1.2	08-02-2010	Raúl García-Castro	Implemented the QA comments

Executive Summary

The Knowledge Web European Network of Excellence organised two benchmarking activities with the goal of assessing and improving the interoperability of Semantic Web technology using RDF(S) and OWL (the languages recommended by the W3C) as interchange languages.

Within the NeOn project, the NeOn Toolkit has been benchmarked with the methods and benchmark suites provided for these benchmarking activities. Two rounds of the interoperability results were presented in 2007 and 2008 and, this year, we have re-evaluated the NeOn Toolkit with the latest version available at the time of writing this deliverable, which includes OWL 2 support.

The results show that the interoperability of the NeOn Toolkit is similar to that of the previous evaluated version and it is not a full OWL-interoperable tool yet because of its behaviour when dealing with ontologies that include individuals. This behaviour is inherited from the current snapshot of the OWL API, which the Neon Toolkit uses for managing ontologies. The NeOn Toolkit team is aware of them and will migrate to the official release version of the API once it is publicly made available.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The Knowledge Web[1] European Network of Excellence organised two benchmarking activities with the goal of assessing and improving the interoperability of the Semantic Web technology using an interchange language; the interchange languages used for those activities are those recommended by the W3C, namely RDF(S) and OWL.

In the NeOn project, the interoperability of the NeOn Toolkit has been evaluated with the methods and benchmark suites provided for these benchmarking activities. Therefore, and with the aim of obtaining objective evaluation results, the whole evaluation and the analysis of the results have been performed by a person who does not belong to the NeOn Toolkit developers group.

A first round of the interoperability results was presented in 2007 in the NeOn deliverable D6.8.1 [GC07], including the results of version 1.0 B823 of the NeOn Toolkit, which run in the Frame-Logic mode with no native OWL support. A second round of the interoperability results was presented in 2008 in the NeOn deliverable D6.8.2 [GC08], including the results of version 1.2.0 B739 of the NeOn Toolkit, which included native OWL support. This year, we have re-evaluated the NeOn Toolkit with the latest version available at the time of writing this deliverable (2.3.0 B204), which includes native OWL 2 support since it is based on the (Manchester) OWL API, which is the reference implementation of the OWL 2 recommendations.

This deliverable includes the analysis of the results of benchmarking the interoperability of the NeOn Toolkit with other Semantic Web tools (GATE, Jena, KAON2, Protégé Frames, Protégé OWL, SemTalk, SWI-Prolog, WebODE, and itself) using OWL as interchange language; it also includes the recommendations extracted from the analysis performed for improving the NeOn Toolkit.

Unlike in previous deliverables, this one does not cover interoperability using RDF(S) because the last version of the NeOn Toolkit, while it still supports importing RDF(S) ontologies, no longer allows exporting ontologies to pure RDF(S) since such ontologies are translated into OWL when imported.

Since we have evaluated the interoperability of the NeOn Toolkit three times, in this deliverable we also present the evolution over time of the interoperability results of the NeOn Toolkit. In order to clearly identify the improvement (or loss) of interoperability gained after updating the NeOn Toolkit, we chose to perform the experiments maintaining the versions of the tools used in the previous experiments.

We expected not to find any execution failure in the tool and to correctly interchange with the other tools the common parts of their knowledge models. It must be noted that interoperability also depends on the other tools participating in the interchanges.

This deliverable is structured as follows. Chapter 2 gives a description of the interoperability benchmarking activities and of the experiment defined in these activities. Chapter 3 presents the analysis performed on the NeOn Toolkit OWL interoperability results. Chapter 4 offers some recommendations to improve the interoperability of the NeOn Toolkit and, finally, chapter 5 draws some conclusions from the results.

[1] http://knowledgeweb.semanticweb.org/

Chapter 2

Interoperability benchmarking

As commented above, the Knowledge Web[1] European Network of Excellence organised two benchmarking activities, the RDF(S) Interoperability Benchmarking [GCGPS07] and the OWL Interoperability Benchmarking [GCGP08], that had two main goals[2]:

- **To assess and improve the interoperability of Semantic Web technologies** using RDF(S) and OWL as interchange languages. This would permit learning about the current interoperability of the tools and maximising the knowledge that these tools can interchange while minimising the information addition or loss.
- **To identify the fragment of knowledge that the Semantic Web technologies can share** using RDF(S) and OWL as interchange languages. As this fragment becomes larger, more expressive ontologies can be interchanged among these technologies.

These two benchmarking activities followed the Knowledge Web benchmarking methodology [GCMW+04] for Semantic Web technologies and provided the following resources for automatically evaluating the interoperability of Semantic Web technologies:

- A manual and an automatic experimentation approach for benchmarking interoperability. For benchmarking the interoperability of the NeOn Toolkit we have followed the automatic experimentation approach described in section 2.1.
- Several ontology datasets for evaluating the import, export and interoperability capabilities of the tools that contain ontologies with simple combinations of the RDF(S) and OWL knowledge models. The ontology dataset used for benchmarking the RDF(S) interoperability of the NeOn Toolkit has been the RDF(S) Import Benchmark Suite[3] and the ontology dataset used for benchmarking the OWL interoperability has been OWL Lite Import Benchmark Suite[4].
- The IBSE (Interoperability Benchmark Suite Executor) tool[5], which is the interoperability evaluation infrastructure that automates the execution of the experiments and provides HTML summarised views of the obtained results.

2.1 Experiment performed

The experiment performed consisted of measuring the interoperability of the tools through the interchange of ontologies from one tool to another. From these measurements, we can extract the interoperability between

[1] http://knowledgeweb.semanticweb.org/
[2] http://knowledgeweb.semanticweb.org/benchmarking_interoperability/
[3] http://knowledgeweb.semanticweb.org/benchmarking_interoperability/rdfs/rdfs_import_benchmark_suite.html
[4] http://knowledgeweb.semanticweb.org/benchmarking_interoperability/owl/import.html
[5] http://knowledgeweb.semanticweb.org/benchmarking_interoperability/ibse

the tools, the causes of problems, and improvement recommendations.

Of the different ways that Semantic Web tools have to interoperate, we only consider interoperability when the tools interchange ontologies by using an interchange language. Therefore, the functionalities affecting the results are the importers and exporters of the tools to the interchange language. Besides, with no human intervention, we can only access tools through application programming interfaces (APIs), and thus the operations performed to access them must be supported by most of the Semantic Web tools. Therefore, the only operations to be performed by a tool should be the following: to import one ontology from a file (to read one file with an ontology and to store this ontology in the tool knowledge model), and to export one ontology into a file (to write an ontology stored in the tool knowledge model into a file).

During the experiment, a common group of benchmarks is executed and each benchmark describes one input ontology that has to be interchanged between a single tool and the others.

Each benchmark execution comprises two sequential steps, shown in Figure 2.1. Starting with a file that contains an ontology, the first step (*Step 1*) consists of importing the file with the ontology into the original tool and then exporting such ontology into a file using the interchange language. The second step (*Step 2*) consists in importing the file with the ontology (exported by the original tool) into the destination tool and then exporting such ontology into another file.

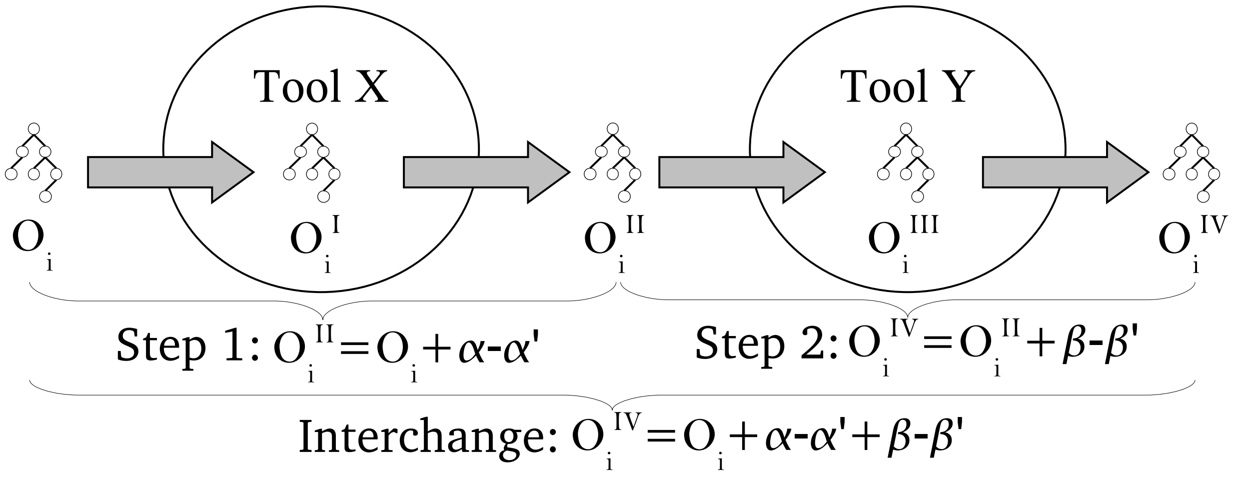

Figure 2.1: The two steps of a benchmark execution.

In these steps, there is no common way for the tools to check the results of importing the ontologies, we just have the results of combining the import and export operations (the files exported by the tools), so we consider these two operations as an atomic operation. It must be noted, therefore, that if a problem arises in one of these steps, we cannot know whether the problem was originated when importing or when exporting the ontology since we do not know the state of the ontology inside each tool.

After a benchmark execution, the results obtained from the ontology described in the benchmark are three different states, namely, the original ontology, the intermediate ontology exported by the first tool, and the final ontology exported by the second tool. From these results we define the evaluation criteria for a benchmark execution. These evaluation criteria will be considered in *Step 1*, *Step 2*, and in the whole interchange (*Step 1* + *Step 2*); they are the following:

- **Execution** (*OK*/*FAIL*/*C.E.*/*N.E.*) informs of the correct execution of a step or the whole interchange. Its value is *OK* if the step or the whole interchange is carried out with no execution problem; *FAIL* if the step or the whole interchange is carried out with some execution problem; *C.E.* (Comparer Error) if the comparer launches an exception when comparing the original and the final ontology; and *N.E.* (Not Executed) if the second step is not executed because the execution on the first step failed.

- **Information added or lost** informs of the information added to or lost from the ontology in terms of triples in each step or in the whole interchange. We can know the triples added or lost in *Step 1*, in *Step 2*, and in the whole interchange by comparing the original ontology with the intermediate one, then the intermediate ontology with the final one, and the original with the final ontology, respectively.

- **Interchange** (*SAME*/*DIFFERENT*/*NO*) informs whether the ontology has been interchanged correctly with no addition or loss of information. From the previous basic measurements we can define *Interchange* as a derived measurement that is *SAME* if *Execution* is *OK* and *Information added* and *Information lost* are void; *DIFFERENT* if *Execution* is *OK* but *Information added* or *Information lost* are not void; and *NO* if *Execution* is *FAIL*, *N.E.* or *C.E.*.

Chapter 3

OWL interoperability evaluation

In this chapter we present the analysis of the interoperability of the NeOn Toolkit with other Semantic Web technologies using OWL as interchange language. The tools evaluated and their corresponding versions can be seen in Table 3.1.

Tool	Version	Developer
GATE	4.0	Sheffield University
Jena	2.3	HP
KAON2	2006-09-22	Karlsruhe University
NeOn Toolkit	2.3.0 build 204	The NeOn project
Protégé	3.3 build 395	Stanford University
Protégé-OWL	3.3 build 395	Manchester University
SemTalk	2.3	Semtation
SWI-Prolog	5.6.35	University of Amsterdam
WebODE	2.0 build 140	Universidad Politécnica de Madrid

Table 3.1: Tools participating in the OWL interoperability evaluation.

This analysis is carried out in two consecutive steps:

1. We describe the behaviour of the NeOn Toolkit in the combined operation of importing one OWL ontology and exporting it again (a step of the experiment as defined in Section 2.1).
2. With the information about the behaviour of the NeOn Toolkit during a step of the experiment, we provide the analysis of the interoperability of the NeOn Toolkit with all the tools participating in the benchmarking (including itself).

Additionally, within the analysis we provide references to the ontology or ontologies that originated the comment with their names between parentheses, i.e. *(ISA01-ISA03)*.

3.1 OWL compliance results

For analysing the behaviour of the NeOn Toolkit in one single step of the experiment (a combined import and export operation), we have considered the results of the NeOn Toolkit when it is the origin of the interchange (*Step 1*), irrespective of the tool that is the destination of the interchange. This step has as input one original ontology that is imported by the NeOn Toolkit and then exported into a resultant ontology.
The results of a step execution in the NeOn Toolkit can be classified in four types:

- *The resultant ontology includes more information than the original one*. This happens in the 15 cases where the original ontology contains named individuals (ISH01-03, ISI01-05, ISK01-03, ISL07-08, ISL11-12).

  In these cases, the NeOn Toolkit exports named individuals as instances of the *owl:NamedIndividual* class, which is not included in the OWL specification (it is part of the OWL 2 language), and, therefore, converts the ontologies into OWL Full.

- *The resultant ontology includes less information than the original one*. In this case, information is also inserted in the former ontology. This occurs in the 3 cases where anonymous individuals with a datatype property appear in the ontology (ISJ03, ISL13-14).

  In these cases, the anonymous individuals lose their type.

- *Execution fails in the import and export operation*. This happens in the 2 cases where the original ontology contains anonymous individuals with an object property (ISJ01-02).

- *The original and the resultant ontologies are equivalent*. This happens in the rest of the cases.

In summary, we can say that the NeOn Toolkit is compliant with OWL Lite with the exception of ontologies that include individuals. Nevertheless, we must take into account that the OWL Lite Import Benchmark Suite does not exhaustively cover the OWL Lite knowledge model and that other problems can be detected with new combinations of components not included in the benchmark suite.

3.2 OWL interoperability results

For analysing the interoperability of the NeOn Toolkit with the other tools participating in the benchmarking (including itself), we have considered the results of the interoperability of the NeOn Toolkit when it is the origin and the destination of the interchange with all the other tools.

Table 3.2[1] gives an overview of the interoperability between the tools and shows the percentage of benchmarks in which the original (O_i) and the resultant (O_i^{IV}) ontologies in an interchange are the same. For each cell, the row indicates the tool origin of the interchange, whereas the column indicates the destination tool.

		DESTINATION								
		JE	**PO**	**SP**	**NT**	**KA**	**GA**	**ST**	**WE**	**PF**
ORIGIN	**JE**	100	100	100	**76**	58	70	0	31	4
	PO	100	100	95	**77**	58	78	0	31	4
	SP	100	100	100	**76**	58	91	46	31	4
	NT	**76**	**76**	**76**	**76**	**59**	**74**	**34**	**26**	**26**
	KA	58	58	58	**52**	58	67	45	19	13
	GA	92	92	75	**30**	56	60	0	29	25
	ST	41	41	46	**5**	36	39	40	34	0
	WE	31	31	0	**26**	19	29	20	31	20
	PF	4	4	0	**5**	0	3	0	4	4

Table 3.2: Percentage of identical interchanged ontologies in the OWL evaluation.

Table 3.3 shows the results from the interoperability of the NeOn Toolkit with the other tools. In this table, the results of the interoperability between two tools (i.e., T1 and T2) have been grouped in categories, as in the previous section, and they include the interchange from one tool to another (from T1 to T2) and vice versa (from T2 to T1). The results in the table are restrictive, i.e., when a single benchmark in a category has a problem in one of the directions of the interchange, then the whole category has this problem.

[1] Tool names have been shortened in the tables: GA=GATE, JE=Jena, K2=KAON2, NT=NeOn Toolkit, PF=Protégé Frames, PO=Protégé OWL, ST=SemTalk, SP=SWI-Prolog, and WE=WebODE.

Categories	Benchmarks	GA-NT	JE-NT	K2-NT	NT-NT	PF-NT	PO-NT	ST-NT	SP-NT	WE-NT
Class hierarchies										
Named class hierarchies without cycles	ISA01-ISA04	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF	SAME	SAME
Named class hierarchies with cycles	ISA05-ISA06	DIFF	SAME	SAME	SAME	DIFF	SAME	DIFF	SAME	DIFF
Classes subclass of a value constraint in an object property	ISA07-ISA08	DIFF	SAME	SAME	SAME	DIFF	SAME	DIFF	SAME	DIFF
Classes subclass of a cardinality constraint in an object property	ISA09-ISA12	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF
Classes subclass of a cardinality constraint in a datatype property	ISA13-ISA16	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF
Classes subclass of a class intersection	ISA17	DIFF	SAME	SAME	SAME	DIFF	SAME	DIFF	SAME	DIFF
Class equivalences										
Equivalent named classes	ISB01	SAME	SAME	SAME	SAME	DIFF	SAME	DIFF	SAME	DIFF
Classes equivalent to a value constraint in an object property	ISB02-ISB03	DIFF	SAME	SAME	SAME	DIFF	SAME	DIFF	SAME	DIFF
Classes equivalent to a cardinality constraint in an object property	ISB04-ISB07	SAME	SAME	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF
Classes equivalent to a cardinality constraint in a datatype property	ISB08-ISB11	SAME	SAME	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF
Classes equivalent to a class intersection	ISB12	SAME	SAME	SAME	SAME	DIFF	SAME	DIFF	SAME	DIFF
Classes defined with set operators										
Classes intersection of other classes	ISC01-ISC02	SAME	SAME	SAME	SAME	DIFF	SAME	DIFF	SAME	DIFF
Property hierarchies										
Object property hierarchies	ISD01-ISD02	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF
Datatype property hierarchies	ISD03-ISD04	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF
Properties with domain and range										
Object properties without domain or range	ISE01-ISE02	DIFF	SAME	SAME	SAME	DIFF	SAME	DIFF	SAME	SAME
Object properties with domain and range	ISE03-ISE04	DIFF	SAME	SAME	SAME	DIFF	SAME	DIFF	SAME	SAME
Object properties with multiple domains or ranges	ISE05-ISE06	DIFF	SAME	SAME	SAME	DIFF	SAME	DIFF	SAME	DIFF
Datatype properties without domain or range	ISE07-ISE08	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF
Datatype properties with domain and range	ISE09	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF
Datatype properties with multiple domains	ISE10	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF
Relations between properties										
Equivalent object and datatype properties	ISF01-ISF02	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF
Inverse object properties	ISF03	DIFF	SAME	SAME	SAME	DIFF	SAME	DIFF	SAME	DIFF
Global cardinality constraints and logical property characteristics										
Transitive object properties	ISG01	DIFF	SAME	SAME	SAME	DIFF	SAME	DIFF	SAME	SAME
Symmetric object properties	ISG02	DIFF	SAME	SAME	SAME	DIFF	SAME	DIFF	SAME	SAME
Functional object and datatype properties	ISG03-ISG04	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF	SAME	DIFF
Inverse functional object properties	ISG05	DIFF	SAME	SAME	SAME	DIFF	SAME	DIFF	SAME	DIFF
Single individuals										
Instances	ISH01, ISH03	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF
Instances of multiple classes	ISH02	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF
Named individuals and properties										
Named individuals and object properties	ISI01-ISI03	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF
Named individuals and datatype properties	ISI04-ISI05	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF
Anonymous individuals and properties										
Anonymous individuals and object properties	ISJ01-ISJ02	N.E.	N.E.	N.E.	N.E.	N.E.	N.E.	N.E.	N.E.	N.E.
Anonymous individuals and datatype properties	ISJ03	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	N.E.	DIFF	DIFF
Individual identity										
Equivalent individuals	ISK01	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF
Different individuals	ISK02-ISK03	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF	DIFF

Table 3.3: Summary of the results of the OWL interoperability of the NeOn Toolkit.

The results for a category can be the following:

- **SAME.** When all the ontologies interchanged between two tools are the same (all the benchmarks in the category have an *INTEROPERABILITY* result of *SAME*).
- **DIFF.** When at least one ontology interchanged between two tools is different and there were no execution errors (any benchmark in the category has an *INTEROPERABILITY* result of *DIFFERENT* and no benchmark exists with an *EXECUTION* result of *N.E.*).
- **N.E.** When at least one ontology could not be interchanged between two tools because of an execution error (any benchmark in the category has an *EXECUTION* result of *N.E.* - Non Executed).

In Table 3.3 we can see that the interoperability results of the **NeOn Toolkit with itself, Jena, Protégé OWL and SWI-Prolog** depend on the behaviour of the NeOn Toolkit described in the previous section. Therefore, the NeOn Toolkit can interchange with these tools all the combinations of components of the OWL Lite knowledge model with the exception of ontologies that include individuals.

The interoperability results of the **NeOn Toolkit with GATE** depend on the behaviour of the NeOn Toolkit but also on the behaviour of GATE. This means that the NeOn Toolkit can only interchange with GATE ontologies that include equivalent named classes, classes equivalent to cardinality constraints, and classes equivalent to class intersections.

The interoperability results of the **NeOn Toolkit with KAON2** depend on the behaviour of the NeOn Toolkit but also on the behaviour of KAON2. This means that the NeOn Toolkit can only interchange with KAON2 ontologies that include named class hierarchies with cycles, classes subclass of value constraints and of class intersections, equivalent named classes, classes equivalent to value constraints and to class intersections, classes intersection of other classes, object properties, and inverse, transitive, symmetric, and inverse functional object properties.

The interoperability results of the **NeOn Toolkit with Protégé Frames** depend on the behaviour of the NeOn Toolkit but also on the behaviour of Protégé Frames. Therefore, the NeOn Toolkit cannot interchange any combination of components with Protégé Frames.

The interoperability results of the **NeOn Toolkit with SemTalk** depend on the behaviour of the NeOn Toolkit but also on the behaviour of SemTalk. Therefore, the NeOn Toolkit cannot interchange any combination of components with SemTalk.

The interoperability results of the **NeOn Toolkit with WebODE** depend on the behaviour of the NeOn Toolkit but also on the behaviour of WebODE. Therefore, the only combinations of components that the NeOn Toolkit can interchange with WebODE are: named class hierarchies without cycles, object properties with or without domain or range, and transitive and symmetric object properties.

3.3 Evolution of OWL interoperability results

This section compares the results of the NeOn Toolkit in its last version (version 2.3.0 B204) to its previous results. We present first the evolution of the OWL compliance results and, second, the evolution of the OWL interoperability results with all the tools (including the NeOn Toolkit itself). It must be noted that results are comparable only to a certain extent because of the significant changes performed inside the NeOn Toolkit from the previous versions to the current one.

Regarding the OWL compliance of the NeOn Toolkit, table 3.4 presents the results of a step execution for the NeOn Toolkit before and after the changes; it shows the number of benchmarks in each category in which the results of a step execution can be classified. In such results we can observe that the updated version of the NeOn Toolkit performs similarly to the previous one, but it includes some execution problems.

If we compare the percentage of benchmarks in which the original and the resultant ontologies in an interchange are the same (see tables 3.6 and 3.5 for the previous results and table 3.2 for the current ones), we can see that the overall interoperability is similar to that from the previous version. Besides, in some cases interoperability with some tools has increased and in other cases it has decreased.

	NeOn Toolkit v1.0 B823	NeOn Toolkit v1.2.0 B739	NeOn Toolkit v2.3.0 B204
Same	23	66	62
More		11	15
Less	59	5	3
Tool fails			2
Comparer fails			
Not valid			
TOTAL	82	82	82

Table 3.4: Updated results in *Step 1* in the OWL evaluation.

		DESTINATION								
		JE	**PO**	**SP**	**NT**	**KA**	**GA**	**ST**	**WE**	**PF**
ORIGIN	**JE**	100	100	100	**28**	58	70	0	31	4
	PO	100	100	95	**28**	58	78	0	31	4
	SP	100	100	100	**28**	58	91	46	31	4
	NT	**28**	**28**	**28**	**28**	**15**	**28**	**15**	**15**	**0**
	KA	58	58	58	**15**	58	67	45	19	13
	GA	92	92	75	**23**	56	60	0	29	25
	ST	41	41	46	**0**	36	39	40	34	0
	WE	31	31	0	**21**	19	29	20	31	20
	PF	4	4	0	**4**	0	3	0	4	4

Table 3.5: Percentage of identical interchanged ontologies in the first OWL evaluation.

		DESTINATION								
		JE	**PO**	**SP**	**NT**	**KA**	**GA**	**ST**	**WE**	**PF**
ORIGIN	**JE**	100	100	100	**80**	58	70	0	31	4
	PO	100	100	95	**80**	58	78	0	31	4
	SP	100	100	100	**80**	58	91	46	31	4
	NT	**80**	**80**	**80**	**80**	**58**	**84**	**57**	**31**	**4**
	KA	58	58	58	**58**	58	67	45	19	13
	GA	92	92	75	**78**	56	60	0	29	25
	ST	41	41	46	**48**	36	39	40	34	0
	WE	31	31	0	**31**	19	29	20	31	20
	PF	4	4	0	**4**	0	3	0	4	4

Table 3.6: Percentage of identical interchanged ontologies in the second OWL evaluation.

Chapter 4

Recommendations for improving the NeOn Toolkit

The following recommendations are intended to improve the interoperability of the NeOn Toolkit using OWL as the interchange language. These recommendations have been extracted from the analysis presented in Chapter 3.

Though it is not compulsory to follow these recommendations, they would improve the interoperability of the NeOn Toolkit in the identified situations; it has to be noted that, in some cases, the results present the intended behaviour of the tool as programmed by its developers and the tool is working correctly.

In order to increase its interoperability using OWL as interchange language, the NeOn Toolkit:

- Should allow exporting ontologies maintaining backwards compatibility with OWL 1. Right now, the NeOn Toolkit exports named individuals as instances of the *owl:NamedIndividual* class, which may not be recognized by tools that only manage OWL 1.
- Should not lose the type of anonymous individuals when an ontology contains anonymous individuals with datatype properties.
- Should not produce an execution exception when an ontology contains anonymous individuals with object properties.

Chapter 5

Conclusions

This deliverable presents the results of the third iteration of the evaluation of the interoperability of the NeOn Toolkit with other Semantic Web tools.

The results show that the interoperability of the NeOn Toolkit is similar to that from its previous version. Nevertheless, some issues still prevent the NeOn Toolkit from being a full OWL-interoperable tool. Besides, since the current management of ontologies in the NeOn Toolkit completely relies in the OWL API, the improvement of the NeOn Toolkit interoperability is directly related to the improvement of the OWL API.

The OWL API is still undergoing heavy development and has not been officially released. Currently, the NeOn Toolkit is using a SVN snapshot from the end of 2009. The NeOn Toolkit team is aware of some issues with this internal snapshot and will migrate to the official release version of the OWL API once it is publicly made available.

These results have provided a set of recommendations to improve the interoperability of the NeOn Toolkit. However, it has to be taken into account that the interoperability problems encountered were not only caused by the NeOn Toolkit, as it has been observed that in many cases interoperability problems were created by other tools.

We have also presented the improvement in time of the OWL interoperability results of the NeOn Toolkit, showing how the improvement of the NeOn Toolkit also entails the improvement of the interoperability of this tool with the others. Nevertheless, it must be noted that this requires to consider the benchmarking of the interoperability of the NeOn Toolkit as a continuous activity.

Bibliography

[Bar07] J. Barrasa. *Modelo para la definición automática de correspondencias semánticas entre ontologías y modelos relacionales*. PhD thesis, Universidad Politécnica de Madrid. Facultad de Informática, January 2007.

[GC07] R. García-Castro. D6.8.1: Testing the NeOn Toolkit interoperability. Technical report, NeOn, September 2007.

[GC08] R. García-Castro. D6.8.2: Testing the NeOn Toolkit interoperability. Technical report, NeOn, November 2008.

[GCGP08] R. García-Castro and A. Gómez-Pérez. Large-scale benchmarking of the OWL interoperability of Semantic Web technologies. In *Proceedings of the Second IEEE International Conference on Semantic Computing (ICSC2008)*, Santa Clara, CA, USA, August 4-7 2008. IEEE Computer Society.

[GCGPS07] R. García-Castro, A. Gómez-Pérez, and Y. Sure. Benchmarking the RDF(S) interoperability of ontology tools. In *Proceedings of the Nineteenth International Conference on Software Engineering & Knowledge Engineering (SEKE'2007)*, pages 410–415, Boston, Massachusetts, USA, July 9-11 2007. Knowledge Systems Institute Graduate School.

[GCMW+04] R. García-Castro, D. Maynard, H. Wache, D. Foxvog, and R. González-Cabero. D2.1.4 specification of a methodology, general criteria and benchmark suites for benchmarking ontology tools. Technical report, Knowledge Web, December 2004.