Large-scale benchmarking of the OWL interoperability of Semantic Web technologies

Raúl García-Castro and Asunción Gómez-Pérez
Ontology Engineering Group, Departamento de Inteligencia Artificial.
Facultad de Informática, Universidad Politécnica de Madrid, Spain
{rgarcia,asun}@fi.upm.es

Abstract

This paper presents the UPM-FBI, a framework for benchmarking the interoperability of Semantic Web technologies; it also provides an example of how to use its automatic approach for benchmarking the interoperability of such technologies using OWL as the interchange language. The paper also introduces the OWL Lite Import Benchmark Suite and the IBSE tool, both used in the benchmarking activity, and offers an overview of the OWL interoperability results of the eight tools participating in the benchmarking: GATE, Jena, KAON2, Protégé Frames, Protégé OWL, SemTalk, SWI-Prolog, and WebODE.

1. Introduction

Even though the number of Semantic Web tools is already rather large[1], many more tools are created every year, as can be seen in Semantic Web-related conferences and workshops. The heterogeneity of Semantic Web tools exists not only because they have different functionalities (ontology development tools, ontology repositories, ontology matchers, etc.) but also because different tools use different representation formalisms, such as Frames, Description Logics, the Unified Modeling Language[2] (UML), the Ontology Definition Metamodel[3] (ODM), or the Open Biomedical Ontologies[4] (OBO) language; these representation formalisms, in turn, have different knowledge representation expressivity and different reasoning capabilities.

The RDF(S) and OWL languages, proposed by the W3C in 2004, are seen as the interchange languages to be used in the Web; in theory, and because of existence of importers and exporters from/to those languages, tools should be able of interchanging ontologies. But, to which extent does the tool heterogeneity affect this interchange? Do the Semantic Web tools interoperate?

Ontology development and its use in applications require the interchange of ontologies between different tools; however, it is well known that the current Semantic Web tools have problems in interchanging RDF(S) and OWL ontologies, either when these ontologies come from other tools or when they are downloaded from the Web. Such problems sometimes are due to the different representation formalisms used by the tools as not every tool natively supports RDF(S) and OWL; but very often, however, the problems are due to other causes such as defects in the tools.

Not to be aware of such problems causes that the interoperability between the different Semantic Web technologies be unknown, and this is so mainly because such interoperability is not evaluated, since there is no easy way of performing such evaluations.

Previously, the benchmarking of the interoperability of ontology development tools was carried out using RDF(S) as the interchange language [5]. As a result, we obtained a clear picture of the RDF(S) interoperability of the tools participating in the benchmarking, namely, Corese, Jena, KAON, Sesame, Protégé, and WebODE.

But now, the objective is to analyse the OWL interoperability of Semantic Web technologies. To this end, the OWL Interoperability Benchmarking was organised with the goals of *providing mechanisms for large-scale evaluation of the interoperability of Semantic Web technologies using OWL as the interchange language* and of *assessing and improving the current OWL interoperability of Semantic Web technologies*.

Although the OWL Interoperability Benchmarking has similar goals to those of the RDF(S) one, its approach is different. The main changes are intended to broaden the scope of the benchmarking, since we consider any type of Semantic Web technology instead of just ontology development tools, and to automate the experiment execution and the analysis of the results.

[1] 694 tools are listed in `http://www.mkbergman.com/?page_id=325` by 12/05/08

[2] `http://www.uml.org/`

[3] `http://www.omg.org/ontology/`

[4] `http://obofoundry.org/`

This paper presents a summary of the OWL Interoperability Benchmarking and an overview of the OWL interoperability results of the eight tools participating in it: one ontology-based annotation tool (GATE), three ontology repositories (Jena, KAON2, and SWI-Prolog), and four ontology development tools (Protégé Frames, Protégé OWL, SemTalk, and WebODE). A detailed analysis of the interoperability results, including results specific for each tool, can be found at [3].

This paper is structured as follows: Section 2 presents the UPM Framework for Benchmarking Interoperability to be used in interoperability evaluation activities, including the one presented in this paper. Section 3 introduces the OWL Interoperability Benchmarking, and Section 4 describes the experiment performed in this benchmarking activity. Section 5 concerns the set of ontologies to use as input for the experiment, namely, the OWL Lite Import Benchmark Suite. Section 6 deals with IBSE, the automatic evaluation infrastructure and how it can be used. Section 7 provides the analysis of the OWL interoperability of the Semantic Web tools participating in the benchmarking. Section 8 presents other interoperability evaluation initiatives and, finally, Section 9 draws the conclusions from this work and proposes future lines of work.

2. The UPM Framework for Benchmarking Interoperability

The UPM Framework for Benchmarking Interoperability[5] (UPM-FBI) includes all the resources needed for benchmarking the interoperability of Semantic Web technologies using RDF(S) and OWL as interchange languages.

As Figure 1 shows, the UPM-FBI provides four benchmark suites that contain the ontologies to be used in interoperability evaluations and two approaches for performing interoperability experiments (one manual and another automatic), each of them including different tools that support the experiment execution and the result analysis.

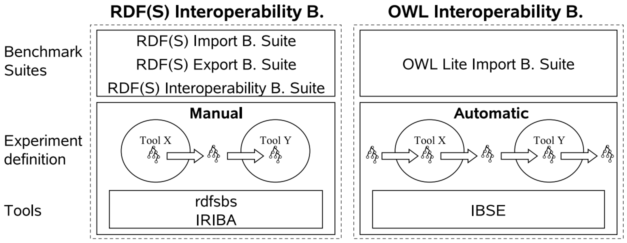

Figure 1: The UPM Framework for Benchmarking Interoperability.

In the RDF(S) Interoperability Benchmarking, experiments were performed by accessing the tools manually, using the RDF(S) Import, Export and Interoperability Benchmark Suites. Two different tools support this approach, namely, the *rdfsbs* tool, which automates part of the experiments execution, and the *IRIBA*[6] web application, which provides an easy way of analysing the results.

The next sections deal with the OWL Interoperability Benchmarking, which uses the automatic approach; in such approach, the OWL Lite Import Benchmark Suite is used and the experiments and the results analysis are automated by means of the IBSE tool.

3. The OWL Interoperability Benchmarking

In the OWL Interoperability Benchmarking, we have followed the Knowledge Web benchmarking methodology [6], a methodology used before in the RDF(S) Interoperability Benchmarking [5] and also employed for benchmarking the performance and the scalability of ontology development tools [4].

The most common way for Semantic Web technologies to interoperate is the indirect interchange of ontologies by storing them in a shared resource, which is the way considered here. A direct interchange of ontologies would require to develop interchange mechanisms for each pair of tools, which would be very costly.

In our case, the representation formalism used to interchange ontologies is OWL, whereas the shared resource is a local filesystem in which ontologies are stored in text files serialized with the RDF/XML syntax, since this is the syntax most used by Semantic Web technologies.

Therefore, the two main goals that we want to achieve in the benchmarking are (1) to provide mechanisms for large-scale evaluation of the interoperability of Semantic Web technologies using OWL as the interchange language, and (2) to assess and improve the OWL interoperability of Semantic Web technologies.

Although the goals here are similar to those of the RDF(S) Interoperability Benchmarking, this time our approach is quite different, thanks in part to the lessons learnt while carrying out the previous benchmarking activity. The main changes performed are the following:

- *Broadening the scope of the benchmarking* by contemplating any Semantic Web tool able to read and write ontologies from/to OWL files.

- *Diminishing the cost of the benchmarking* by automating the experiments. The cost of organising the benchmarking is unavoidable because it involves defining the experiments from scratch, since no previous ones exist.

[5] http://knowledgeweb.semanticweb.org/benchmarking_interoperability/

[6] http://knowledgeweb.semanticweb.org/iriba/

- *Automating the experiments*. Full automation of the result analysis is not possible since this requires a person to interpret them; nevertheless, the automatic generation of different visualizations and summaries of the results in different formats (such as HTML or SVG) allows us to draw some conclusions at a glance.

- *Including new tools easily*, because the effort to be spent in the benchmarking is a main criteria for an organisation when deciding whether to participate in the benchmarking.

In the Semantic Web, the interoperability problem is highly related to the ontology translation problem, which occurs when common ontologies are shared and reused over multiple representation systems [7]. In this paper, interoperability is treated in terms of knowledge reuse and should not be confused with interoperability by means of integration of resources, being the latter related to the ontology alignment problem [2].

In our scenario, interoperability depends on two different tool functionalities, one functionality that reads an ontology stored in the tool and writes it into an OWL file (OWL exporter from now on), and another that reads an OWL file with an ontology and stores this ontology into the tool (OWL importer from now on). Therefore, our experiments provided data not only on the interoperability but also on the OWL importers and exporters of the tools.

To obtain detailed information on tool interoperability using OWL as interchange language, we need to know a) the components of the knowledge model of a tool that can be interchanged with others; b) the secondary effects of interchanging these components, such as insertion or loss of information; c) the subset of the tool knowledge models that the tools can use to correctly interoperate; and d) the problems that arise when interchanging ontologies between two tools and the causes of these problems.

Participation in the benchmarking is open to any Semantic Web tool capable of importing and exporting OWL. A public call for participation was issued and many tool developers were directly contacted to participate in it.

Eight tools took part in the benchmarking: one ontology-based annotation tool (GATE[7]), three ontology repositories (Jena[8], KAON2[9], and SWI-Prolog[10]), and four ontology development tools (Protégé Frames[11], Protégé OWL[12], SemTalk[13], and WebODE[14]).

4. Experiment Performed

Although participation is open to any Semantic Web tool, the experiment requires that the tools participating be able to import and export OWL ontologies, as we need an automatic and uniform way of accessing Semantic Web tools that is supported by most of them.

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

Each benchmark execution comprises two sequential steps (Figure 2). Starting with a file that contains an OWL ontology (O_i), the first step (*Step 1*) consists in importing the file storing the ontology into the origin tool and then exporting the ontology into an OWL file (O_i^{II}). The second step (*Step 2*) consists in importing the file storing the ontology exported by the origin tool (O_i^{II}) into the destination tool and then exporting the ontology into another file (O_i^{IV}).

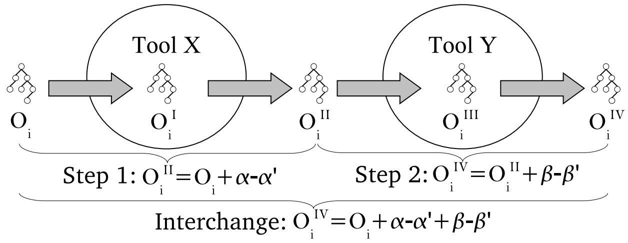
Tool X
Tool Y
O_i
O_i^I
O_i^{II}
O_i^{III}
O_i^{IV}
Step 1: $O_i^{II}=O_i+\alpha-\alpha'$
Step 2: $O_i^{IV}=O_i^{II}+\beta-\beta'$
Interchange: $O_i^{IV}=O_i+\alpha-\alpha'+\beta-\beta'$

Figure 2: The two steps of a benchmark execution.

In these steps, there is not a common way for the tools to check how good the importers (by comparing O_i with O_i^I and O_i^{II} with O_i^{III}) and exporters (by comparing O_i^I with O_i^{II} and O_i^{III} with O_i^{IV}) are. We only have the results of combining the import and export operations (the files exported by the tools), so these two operations are considered as an atomic operation. It must be noted here that if a problem arises in one of these steps, we cannot know whether it was originated when importing or when exporting the ontology, because we are totally unaware of the state of the ontology inside each tool.

After a benchmark execution, we have three ontologies to compare, namely, the original ontology (O_i), the intermediate ontology exported by the first tool (O_i^{II}), and the final ontology exported by the second tool (O_i^{IV}). From these results, we define the following evaluation criteria for a benchmark execution:

- **Execution** (*OK*/*FAIL*/*C.E.*/*N.E.*) informs of the correct execution of a step or of 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

[7] version 4.0 http://gate.ac.uk/

[8] version 2.3 http://jena.sourceforge.net/

[9] version 2006-09-22 http://kaon2.semanticweb.org/

[10] version 5.6.35 http://www.swi-prolog.org/packages/semweb.html

[11] version 3.3 build 395 http://protege.stanford.edu/

[12] version 3.3 build 395 http://protege.stanford.edu/overview/protege-owl.html

[13] version 2.3 http://www.semtalk.com/

[14] version 2.0 build 240 http://webode.dia.fi.upm.es/

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 ontologies; and *N.E.* (Not Executed) if the second step is not executed because the first step execution failed.

- **Information added or lost** informs of the information that is 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.*.

For evaluating the interoperability of the tools, we used the OWL Lite Import Benchmark Suite, described in the next section, which is common for all the tools and contains ontologies with simple combinations of OWL components.

5. The OWL Lite Import Benchmark Suite

The ontologies used in the experiment are those defined for the OWL Lite Import Benchmark Suite and described in detail in [1]. This benchmark suite was intended to evaluate the OWL import capabilities of Semantic Web tools by checking the import of ontologies with simple combinations of components of the OWL Lite knowledge model. It is composed of 82 benchmarks and is available in the Web[15].

Each benchmark of the benchmark suite is described by a unique *identifier*, a *description* in natural language, a *formal description* in Description Logics notation of the ontology, a *graphical representation* of the ontology, and a *file* with the ontology in the RDF/XML syntax.

Since the RDF/XML syntax allows serializing ontology components in different ways while maintaining the same semantics, the benchmark suite includes two kinds of benchmarks: one to check the import of the different combinations of the OWL Lite vocabulary terms, and another to check the import of OWL ontologies with the different variants of the RDF/XML syntax. The first two columns of Table 2 show the groups of the OWL Lite Import Benchmark Suite and the number of benchmarks in each group.

The OWL Lite Import Benchmark Suite is here used to evaluate the interoperability of Semantic Web tools. Nevertheless, any group of ontologies could be used as input for the experiment. For example, we could employ a group of real ontologies in a certain domain, ontologies synthetically generated such as the Lehigh University Benchmark (LUBM) [8] or the University Ontology Benchmark (UOB) [9], or the OWL Test Cases[16] (developed by the W3C Web Ontology Working Group).

These ontologies were designed with specific goals and requirements such as performance or correctness evaluation. Being our goal to improve interoperability, these ontologies could complement our experiments but, in our case, we aim to evaluate interoperability with simple OWL ontologies that, even though they may not cover exhaustively the OWL specification, are simple and allow isolating problem causes and highlighting problems in the tools.

6. The IBSE Tool

IBSE (Interoperability Benchmark Suite Executor) is the evaluation infrastructure that automates the execution of the experiments of the OWL Interoperability Benchmarking. IBSE offers a simple way of executing the experiments between any selected group of tools and of analysing the results, and permits including new tools smoothly.

The IBSE tool has been implemented with Java; its source code and binaries are publicly available and can be downloaded from its web page[17]. The only requirements for executing IBSE are to have both a Java Runtime Environment and the IBSE binaries. However, to perform the experiments either with SemTalk or with WebODE, these tools must be previously installed.

The main requirements taken into account in the development of the IBSE tool surge from the benchmarking requirements described in Section 3, and are the following: a) to perform the experiments with any tool able to import and export OWL files; b) to automate the experiment execution and the analysis of the results; c) to define benchmarks and results through ontologies, as the automation mentioned above requires benchmarks and results to be machine-processable; d) to use any group of ontologies as input for the experiments; and e) to separate benchmark execution and report generation.

The IBSE tool employs two OWL ontologies: the *benchmarkOntology*[18] one and the *resultOntology*[19] one, which

[15] http://knowledgeweb.semanticweb.org/benchmarking_interoperability/owl/import.html

[16] http://www.w3.org/TR/owl-test/

[17] http://knowledgeweb.semanticweb.org/benchmarking_interoperability/ibse/

[18] http://knowledgeweb.semanticweb.org/benchmarking_interoperability/owl/benchmarkOntology.owl

[19] http://knowledgeweb.semanticweb.org/benchmarking_

define the vocabulary for representing the benchmarks and the results of a benchmark execution, respectively.

A normal execution of IBSE comprises three consecutive steps that can also be executed independently. These steps are the following:

1. *To generate machine-readable benchmark descriptions from a group of ontologies*. In this step, from a group of ontologies located in a URI, one RDF file with one benchmark for each ontology is generated.

2. *To execute the benchmarks*. In this step, considering all the different combinations of ontology interchanges between the tools, each benchmark described in the RDF file is executed and its results are stored in another RDF file.

   To execute a benchmark between an origin tool and a destination one, as described in Section 4, first the file storing the ontology is imported into the origin tool and then exported into an intermediate file and, second, this intermediate file is imported into the destination tool and then exported into the final file.

   Once we have the original, intermediate and final files with their corresponding ontologies, we can extract the results by comparing these ontologies, as shown in Section 4. This comparison and its output depend on an external ontology comparer. The current implementation makes use of the OWL comparer of the *KAON2 OWL Tools*[20], but other comparers can also be inserted by implementing a Java interface.

3. *To generate HTML files with different visualizations of the results*. In this step, different HTML files are generated with different visualizations, summaries and statistics of the results.

6.1. Inserting a new tool

Inserting a new tool in IBSE is easy; this can be done by either implementing a Java interface in IBSE or building a program that imports an ontology from a file and exports the imported ontology into another file.

Most of the tools have implemented the Java interface since they provide Java methods for performing the import and export operations. With non-Java tools (SemTalk and SWI-Prolog), these operations are performed by executing precompiled binaries.

6.2. Evaluating the ontology comparer

As the software used for ontology comparison could have execution problems, we have evaluated it in two steps:

First, the interoperability experiment was carried out with the tools that have OWL as knowledge model, since these tools should interchange all the ontologies correctly as no ontology translation is required for the interchange. In this step, the cases in which the interchanged ontology was different than the original one were analysed.

And second, the interoperability experiment was carried out with all the tools. In this step, the cases in which the comparison of two ontologies caused an execution error in the comparer were analysed.

After carrying out the previous steps, we found several problems in the KAON2 OWL Tools ontology comparer (see [3] for details). Some of the problems were solved by adapting the output of the comparer inside IBSE, whereas in the other cases the behaviour of the ontology comparer was documented and taken into account when analysing the interoperability results. Although we did not make an exhaustive evaluation of the comparer, after analysing all the benchmarking results, we found no more errors.

7. OWL Interoperability Results

In this section we present the analysis of the OWL interoperability of the eight tools that participated in the benchmarking. The IBSE tool was adapted to include these tools and the authors executed automatically the experiments with the tool and analysed the results. A detailed analysis of the interoperability results, including results specific for each tool, can be found at [3]. The HTML and RDF files generated by the IBSE tool are available in the Web[21].

Because of the large number of benchmark executions (for 9 tools we have 81 possible interoperability scenarios, each composed of 82 benchmark executions, which results in 6642 benchmark executions), for each of the tools we have carried out the analysis in two consecutive steps (described in detail below):

1. To analyse the behaviour of the tool when managing OWL ontologies in the combined import and export operation, as it will affect the tool interoperability.

2. To analyse the interoperability of the tool with all the tools participating in the benchmarking (including the tool itself).

7.1. Analysis of the import and export operation

Here we describe how the tools behave in the combined operation of importing one OWL ontology and exporting it again (such operation is a step of the experiment, as defined

interoperability/owl/resultOntology.owl

[20] version 0.27 http://owltools.ontoware.org/

[21] http://knowledgeweb.semanticweb.org/benchmarking_interoperability/owl/2007-08-12_Results/

in Section 4). To analyse this behaviour, we have considered the results of the tool 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 an original ontology that is imported by the tool (O_i) and then exported into a resultant ontology (O_i^{II}). This analysis is been performed by comparing the original and the resultant ontologies.

Table 1 presents the results of executing a step for each tool[22]. It shows the number of benchmarks in every category in which these results can be classified:

- *The original and the resultant ontologies are the same*. The only tools that always produce the same ontologies are Jena, Protégé OWL and SWI-Prolog. Frame-based tools (Protégé Frames and WebODE) rarely produce the same ontologies; this is so because they usually insert and remove information when importing and exporting.
- *The resultant ontology includes more information than the original one*. This only happens with Protégé Frames and WebODE, as they insert *rdfs:label* properties into classes and properties with their names.
- *The resultant ontology includes less information than the original one*. In this case, information is sometimes inserted into the resultant ontology.
- *The execution fails in the import and export operation*. The tools do not have execution problems.
- *The execution fails when the ontologies are compared*. There are several cases in which the execution of the comparer fails when it compares two ontologies, as we observed in the evaluation of the comparer. This failure does not let us to know whether the tool behaves correctly or not, but it pinpoints cases that should be analysed in detail. Nevertheless, these figures are an indicator of the low robustness of the comparer used.

Table 1: Results in *Step 1* (for 82 benchmarks).

	GA	JE	K2	PF	PO	ST	SP	WE
Same	79	82	63	4	82	39	82	14
More				4				11
Less	2		11	56		33		57
Tool f.								
Comp. f.	1		8	18		10		

Table 2 is a breakdown of the row "*Same*" in Table 1, according to the combination of components present in the ontology; it shows the number of benchmarks in each group and the percentage of benchmarks whose original (O_i) and resultant (O_i^{II}) ontologies are the same in *Step 1*. It can be observed that some tools work better with some component combinations than with others.

7.2. Analysis of the interoperability

With the previous information about the behaviour of the tools in the *Step 1* of the experiment, we provide the analysis of their interoperability with all the tools participating in the benchmarking. In this analysis we consider all the tools because when in *Step 1* a tool produces an ontology different from the original one, this tool may be working correctly, as intended by its developers (e.g., the resultant ontology is semantically equivalent to the original one, or the tool just inserts annotation properties).

In order to analyse the interoperability between two tools (i.e., T1 and T2), we have considered the interchange from one tool to another (from T1 to T2) and vice versa (from T2 to T1).

Table 3 provides an overview of the **interoperability between the tools**; it 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 and the column indicates the tool destination of the interchange.

Table 3: Percentage of identical ontologies after the interchange.

ORI.	**DESTINATION**							
	JE	**PO**	**SP**	**K2**	**GA**	**ST**	**WE**	**PF**
JE	100	100	100	78	85	16	17	5
PO	100	100	95	78	89	16	17	5
SP	100	100	100	78	55	45	17	5
K2	78	78	78	78	40	39	6	0
GA	96	52	79	74	46	13	15	13
ST	45	46	46	27	24	46	17	0
WE	17	18	0	6	16	17	17	12
PF	5	5	0	0	4	5	0	13

At a glance, we can observe that the interoperability between the tools is low, even in interchanges between a tool and itself.

Is it also clear from the results that interoperability using OWL as interchange language depends on the knowledge model of the tools, hence the more similar the knowledge model of a tool is to OWL the more interoperable the tool is. Nevertheless, the way of serializing the ontologies in the RDF/XML syntax also has a high influence on the results.

The correct working of a tool importers and exporters does not ensure interoperability. Not all the tools that produced the same ontologies in the first step also pro-

[22] The tool names have been abbreviated in the tables: GA=GATE, JE=Jena, K2=KAON2, PF=Protégé Frames, PO=Protégé OWL, ST=SemTalk, SP=SWI-Prolog, and WE=WebODE.

Table 2: Percentage of identical ontologies per group in *Step 1*.

Benchmark group	**No.**	**GA**	**JE**	**K2**	**PF**	**PO**	**ST**	**SP**	**WE**
A - Class hierarchies	17	47	100	71	6	100	35	100	24
B - Class equivalences	12	50	100	75	0	100	0	100	0
C - Classes defined with set operators	2	50	100	100	0	100	100	100	0
D - Property hierarchies	4	50	100	50	50	100	75	100	0
E - Properties with domain and range	10	50	100	100	0	100	70	100	0
F - Relations between properties	3	33	100	100	0	100	33	100	0
G - Global cardinality constraints and logical property characteristics	5	60	100	100	0	100	60	100	0
H - Single individuals	3	0	100	100	0	100	100	100	67
I - Named individuals and properties	5	40	100	100	0	100	60	100	0
J - Anonymous individuals and properties	3	67	100	100	0	100	0	100	0
K - Individual identity	3	33	100	100	0	100	33	100	0
L - Syntax and abbreviation	15	53	100	47	53	100	60	100	53

duce the same ontologies after interchanging them. Interchanges between Jena and Protégé OWL and interchanges between Jena and SWI-Prolog do produce the same ontologies. But in interchanges between Protégé OWL and SWI-Prolog, when the interchange goes from Protégé OWL to SWI-Prolog some problems arise: SWI-Prolog produces ontologies with an incorrect namespace identifier (*[]*) when it imports ontologies that contain default namespaces (*xmlns="namespaceURI"*).

This leads us to a second fact, that interoperability between two tools is usually different depending on the direction of the interchange. This can be clearly seen in the above table and in the previous example.

To analyse the **interoperability of the tools regarding the combination of components present in the ontology**, we have grouped the tools into clusters. We can see that Jena, KAON2, Protégé OWL, and SWI-Prolog can interchange correctly all the combinations of components except class hierarchies, class equivalences and property hierarchies. Jena, Protégé OWL, and SWI-Prolog can interchange correctly all the combinations of components but, as there are some problems when Protégé OWL interchanges ontologies with SWI-Prolog in the *Syntax and abbreviation* benchmarks, the only two clusters of fully-interoperable tools are Jena with Protégé OWL and Jena with SWI-Prolog. Furthermore, in some cases interchanges can be performed in one direction but not in both[23].

With regard to the **robustness of the tools**, we can see that tools have no execution problems when processing the ontologies of the benchmark suite; however, some of them do have problems when processing ontologies generated by other tools. Needless to say, this lack of robustness also has a negative effect in interoperability.

8. Other Interoperability Evaluations

This section presents two other initiatives that deal with interoperability evaluations: the experiments of the Second International Workshop on Evaluation of Ontology-based Tools and the RDF(S) Interoperability Benchmarking.

The central topic of the **Second International Workshop on Evaluation of Ontology-based Tools** (EON2003) was the evaluation of ontology development tools interoperability [10]. In this workshop, the participants were asked to model ontologies with their ontology development tools and to perform different tests for evaluating tool import, export and interoperability.

In these experiments, there was no constraint regarding the interchange language to be used; of the experiments carried out only two used OWL as interchange language.

Furthermore, no systematic evaluation was performed; each experiment used different evaluation procedures and principles for modelling ontologies. Therefore, the results were not comparable and only specific comments and recommendations for each ontology development tool participating were made.

As mentioned before, the **RDF(S) Interoperability Benchmarking**, a benchmarking of the interoperability of ontology development tools using RDF(S) as the interchange language, was organised before we started the benchmarking here presented [5].

In the RDF(S) Interoperability Benchmarking, the experiments and the analysis of the results were performed manually. This had the advantage of yielding highly detailed results, which permits diagnosing problems in the tools and, consequently, improving them, but the disadvantage that it makes the experimentation costly. Some tool developers automated the execution of the experiments but not all of them. Furthermore, the results obtained may be influenced by human mistakes and they depend on the people performing the experiments and on their expertise with the tools.

[23] Tables with detailed results according to the combination of components can be found in `http://knowledgeweb.semanticweb.org/benchmarking_interoperability/owl/2007-08-12_Results/per_group.html`

9. Conclusions

This paper is intended to serve not just as a summary of the OWL Interoperability Benchmarking, but as a guide to perform benchmarking activities or interoperability evaluations over Semantic Web technologies using the UPM Framework for Benchmarking Interoperability.

The main goal of this work, the assessment of the current interoperability of eight best-in-class Semantic Web tools, has been fulfilled. Such assessment has provided us with detailed results of the behaviour of the tools not just when they interoperate with other tools, but also when they import and export OWL ontologies.

As in the case of the RDF(S) Interoperability Benchmarking, the benchmarking process has been long. And as a result, we have discovered that interoperability between the tools is very low and that real interoperability in the Semantic Web requires the involvement of tool developers.

We have also checked that the interoperability problem not only depends on the ontology translation problem but also on robustness and specification problems. In some cases interoperability problems are due to the representation formalisms managed by the tools, but in others they are due to defects in the tools or to the way of serializing ontologies, having the latter a high impact in interoperability.

This panoramic, although disappointing, can serve to promote the second of our goals: the improvement of the tools. Although this goal is out of our scope right now, because each tool is developed by independent organizations, we hope, nevertheless, that the results we provide may help to their improvement.

The benchmarking results are publicly available in the Web in HTML and in RDF. Thus, anyone can use them and compare them with their own results or reason about them. In addition, the developers of the tools that have participated in this benchmarking are already informed of the results.

The IBSE tool can be used in other scenarios, using any group of ontologies as input or using other languages as interchange. Right now the tool allows performing experiments using RDF(S) as the interchange language and *rdf-utils*[24] as the ontology comparer, other tools should implement the corresponding interface in IBSE and then use the RDF(S) Import Benchmark Suite[25] as ontology dataset.

Finally, the ontology comparer of the KAON2 OWL Tools is not the most appropriate to be used because we detected some problems in it as well as in KAON2's interoperability. Future work includes changing the ontology comparer, either by using another or by developing a new one that could use one of the tools that do not pose interoperability problems.

[24] version 0.3b http://wymiwyg.org/rdf-utils/

[25] http://knowledgeweb.semanticweb.org/benchmarking_interoperability/rdfs/rdfs_import_benchmark_suite.html

Acknowledgments

Thanks to all the people that have participated in the OWL Interoperability Benchmarking by adapting the IBSE tool for some best-in-class Semantic Web tools: Stamatia Dasiopoulou, Danica Damljanovic, Michael Erdmann, Christian Fillies, Roman Korf, Diana Maynard, Jesús Prieto-González, York Sure, Jan Wielemaker, and Philipp Zaltenbach. Without their effort, this could have not been possible. This work is partially supported by a FPI grant from the Spanish Ministry of Education (BES-2005-8024), by the IST project Knowledge Web (IST-2004-507482) and by the CICYT project Infraestructura tecnológica de servicios semánticos para la web semántica (TIN2004-02660). Thanks to Rosario Plaza for reviewing the grammar of this paper.

References

[1] S. David, R. García-Castro, and A. Gómez-Pérez. Defining a benchmark suite for evaluating the import of OWL lite ontologies. In *Proceedings of the OWL: Experiences and Directions 2006 workshop (OWL2006)*, Athens, Georgia, USA, November 10-11 2006.

[2] Euzenat, J., et al. *D2.2.3 State of the art on ontology alignment. Knowledge Web*, June 2004.

[3] R. García-Castro, S. David, and J. Prieto-González. D1.2.2.1.2 Benchmarking the interoperability of ontology development tools using OWL as interchange language. Technical report, Knowledge Web, September 2007.

[4] R. García-Castro and A. Gómez-Pérez. Guidelines for Benchmarking the Performance of Ontology Management APIs. In *Proceedings of the 4th International Semantic Web Conference (ISWC2005)*, number 3729 in LNCS, pages 277–292, Galway, Ireland, November 2005.

[5] 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, USA, July 2007.

[6] 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, Knowledge Web, December 2004.

[7] T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition 5*, (2):199–220, 1993.

[8] Y. Guo, Z. Pan, and J. Heflin. LUBM: A Benchmark for OWL Knowledge Base Systems. *Journal of Web Semantics 3(2)*, (2):158–182, 2005.

[9] L. Ma, Y. Yang, Z. Qiu, G. Xie, Y. Pan, and S. Liu. Towards a complete OWL ontology benchmark. In *Proceedings of the 3rd European Semantic Web Conference (ESWC 2006)*, volume 4011 of *LNCS*, pages 125–139, Budva, June 2006.

[10] Y. Sure and O. Corcho, editors. *Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003)*, vol. 87 of CEUR-WS, Florida, October 2003.