IBSE: An OWL Interoperability Evaluation Infrastructure

Raúl García-Castro, Asunción Gómez-Pérez, Jesús Prieto-González

Ontology Engineering Group, Departamento de Inteligencia Artificial.
Facultad de Informática, Universidad Politécnica de Madrid, Spain
{rgarcia,asun}@fi.upm.es,jprieto@delicias.dia.fi.upm.es

Abstract. The technology that supports the Semantic Web presents a great diversity and, whereas all the tools use different types of ontologies, not all of them share a common knowledge representation model, which may pose problems when they try to interoperate. The Knowledge Web European Network of Excellence is organizing a benchmarking of interoperability of ontology tools using OWL as interchange language with the goal of assessing and improving tool interoperability. This paper presents the development of IBSE, an evaluation infrastructure that allows executing automatically the benchmarking experiments and provides an easy way of analysing the results. Thus, including new tools into the evaluation infrastructure will be simple and straightforward.

1 Introduction

The technology that supports the Semantic Web presents a great diversity and is growing in parallel with the Semantic Web. This technology appears in different forms (ontology development tools, ontology repositories, ontology alignment tools, reasoners, etc.) and, whereas all these tools use different kinds of ontologies, not all of them share a common knowledge representation model.

This diversity in the representation formalisms of the tools causes problems when the tools try to interoperate, that is, to interchange information and use the information that has been exchanged [1]. This is so because the tools require translating their ontologies from their own knowledge models to a common one and vice versa, even when using standard APIs for managing ontologies in the common knowledge model.

OWL[1] is the language recommended by the World Wide Web Consortium for defining and instantiating ontologies; therefore, to use OWL as a language for interchanging ontologies now seems the right choice. But interoperability between Semantic Web tools using OWL is unknown, and to evaluate to what extent one tool is able to interchange ontologies with others is quite difficult as there are no means available for performing this task easily.

To this end, the Knowledge Web[2] European Network of Excellence is organizing the benchmarking of interoperability of ontology tools using OWL as interchange language with the goal of assessing and improving the interoperability of the tools.

[1] http://www.w3.org/2004/OWL/
[2] http://knowledgeweb.semanticweb.org/

To allow as much tools as possible to participate in the benchmarking and to minimise the effort devoted to this participation, we have developed IBSE, an evaluation infrastructure that automatically executes the experiments, offers a simple way of analysing the results, and permits smoothly including new tools into the infrastructure.

The results of a first execution of the experiment carried out with two tools, Jena, an ontology repository, and WebODE, and ontology development tool, are also presented.

This paper is structured as follows: Section 2 presents other interoperability evaluation initiatives. Section 3 introduces the OWL Interoperability Benchmarking and Section 4 the experiment to be performed in this benchmarking activity. Section 5 presents the set of ontologies to use as input for the experiment. Section 6 deals with the OWL ontologies used to represent the benchmarks and their results. Section 7 describes how the evaluation infrastructure has been implemented and how it can be used. Section 8 provides an example of executing the experiment with Jena and WebODE. Finally, Section 9 draws the conclusions from this work and proposes future lines of work.

2 Related Work

This section presents two other initiatives that deal with interoperability evaluations:

EON 2003 Workshop. The central topic of the Second International Workshop on Evaluation of Ontology-based Tools was the evaluation of ontology development tools interoperability [2]. 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 use of the interchange language; of the experiments carried out only two used OWL as interchange language.
- Each experiment was performed with a different procedure; hence the results obtained in that workshop did not provide general recommendations, only specific ones for each ontology development tool participating.

RDF(S) Interoperability Benchmarking. A benchmarking of the interoperability of ontology development tools using RDF(S) as interchange language[3] has been organised in Knowledge Web, before we started the benchmarking presented in this paper [3].

In the RDF(S) Interoperability Benchmarking the experimentation and analysis of the results were performed manually. This has the advantage of obtaining high detailed results, being easier to diagnose problems in the tools and so to improve them. But the manual execution and analysis of the results also makes the experimentation costly. Tools developers have often automated the execution of the benchmark suites but not always. 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.

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

3 OWL Interoperability Benchmarking

In the OWL Interoperability Benchmarking we have followed the Knowledge Web benchmarking methodology [4] for ontology tools, a methodology used before for benchmarking the interoperability of ontology development tools using RDF(S) as the interchange language [3], and for benchmarking the performance and the scalability of ontology development tools [5].

The two main goals that we want to achieve are:

- **To assess and improve the interoperability of ontology development tools** using OWL as the interchange language. This would permit learning about the current interoperability of the tools and maximizing the knowledge that these tools can interchange while minimizing the information addition or loss.
- **To identify the fragment of knowledge that ontology development tools can share** using OWL as the interchange language. As this fragment becomes larger, more expressive ontologies can be interchanged among ontology development tools.

The main changes to perform with regard to the RDF(S) Interoperability Benchmarking presented above are the following:

- **To broaden the scope of the benchmarking**. While the RDF(S) Interoperability Benchmarking targeted at ontology development tools (even though ontology repositories also participated), this time we consider any Semantic Web tool able to read and write ontologies from/to OWL files (ontology repositories, ontology merging and alignment tools, reasoners, ontology-based annotation tools, etc.).
- **To automate the experiment execution and the analysis of the results**. In the OWL Interoperability Benchmarking we sacrifice a higher detail in results to avoid the experiments being conducted by humans. However, full automation of the result analysis is not possible since it requires a person to interpret them; nevertheless, the evaluation infrastructure automatically generates different visualizations and summaries of the results in different formats (such as HTML or SVG) to draw some conclusions at a glance. Of course, an in-depth analysis of these results will still be needed for extracting the cause of the problems encountered and improvement recommendations and the practices performed by developers.
- **To define benchmarks and results using ontologies**. The automation mentioned above requires that both benchmarks and results be machine-processable so we have represented them using ontologies. Instances of these ontologies will include the information needed to execute the benchmarks and all the results obtained in their execution. This also allows having different predefined benchmark suites and execution results available in the Web, which can be used by anyone to, for example, classify and select tools according to their results.
- **To use any group of ontologies as input for the experiments**. Executing benchmarks with no human effort can provide further advantages. We have implemented the evaluation infrastructure to generate benchmark descriptions from any group of ontologies in RDF(S) or OWL and to execute these benchmarks. Thus, we can easily perform different experiments with large numbers of ontologies, domain-specific ontologies, systematically-generated ontologies, etc.

4 Experiment to be Performed

The experiment to be performed consists in measuring the interoperability of the tools participating in the benchmarking through the interchange of ontologies from one tool to another. From these measurements, we will extract the interoperability between 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 interchanging ontologies using OWL. Therefore, the functionalities affecting the results are the OWL importers and exporters of the tools. Also, with no human intervention, we can only access tools through application programming interfaces (APIs) and 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: to import one ontology from a file (to read one file with an ontology and to store this ontology in the tool's knowledge model), and to export one ontology into a file (to write into a file an ontology stored in the tool knowledge model).

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

Each benchmark execution comprises two sequential steps, shown in Figure 1. Starting with a file that contains an ontology, the first step (*Step 1*) consists in importing the file with the ontology into the origin 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 origin tool) into the destination tool and then exporting such ontology into another file.

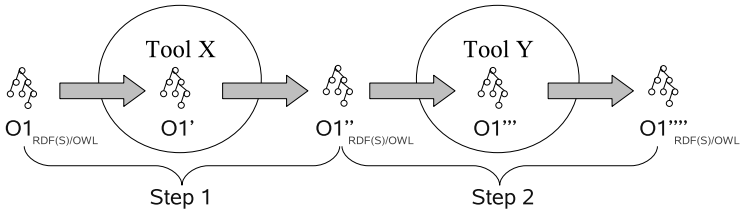

Fig. 1. The two steps of a benchmark execution

In these steps, there is not a 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) and we consider these two operations as an atomic operation. Therefore, it must be noted that if a problem arises in one of these steps, we cannot know whether the problem was caused 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*), and are the following:

- **Execution** (*OK*/*FAIL*/*C.E.*/*N.E.*) informs about 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 origin and final ontologies; 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 about 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 origin ontology with the intermediate ontology, 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.*.

Although this section provides a theoretical description of the experiment to perform in the benchmarking, carrying this experiment out requires coping with some other issues such as how to find an appropriate set of ontologies to use as input for the experimentation, how to define the way of representing the benchmarks and the results in OWL and, finally, how to develop the evaluation infrastructure. The next sections describe the approach followed in each of these issues.

5 Ontology Dataset

As we mentioned above, any group of ontologies could be used as input for the experiment. For example, ontologies synthetically generated such as the Lehigh University Benchmark (LUBM) [6] or the University Ontology Benchmark (UOB) [7], a group of real ontologies in a certain domain, or the OWL Test Cases[4] developed by the W3C Web Ontology Working Group could be employed.

Nevertheless, being our goal to improve interoperability, these ontologies could complement our experiments even though they were designed with specific goals and requirements such as that of performance or correctness evaluation. In our case, we aim to evaluate interoperability with simple OWL ontologies that, although they do not cover exhaustively the OWL specification, allow highlighting problems in the tools.

Therefore, the ontologies used in the experiment are those defined for the OWL Lite Import Benchmark Suite, described in detail in [8]. This benchmark suite is intended to evaluate the OWL import capabilities of the ontology development tools by checking

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

the import of simple combinations of components of the OWL Lite knowledge model. It is composed of 82 benchmarks and is available in the Web[5].

The ontologies composing the benchmark suite are serialized using the RDF/XML syntax as this is the syntax most widely employed by tools when exporting and importing to/from OWL. Since the RDF/XML syntax allows serializing ontology components in different ways while maintaining the same semantics, the benchmark suite includes two groups 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.

6 Representation of Benchmarks and Results

This section defines the two OWL ontologies used in the OWL Interoperability Benchmarking. The *benchmarkOntology*[6] ontology defines the vocabulary that represents the benchmarks to be executed while the *resultOntology*[7] ontology defines the vocabulary that represents the results of a benchmark execution. These ontologies are lightweight, as their main goal is to be user-friendly.

Next, the classes and properties that these ontologies contain are presented. All the datatype properties have as range *xsd:string* with the exception of *timestamp*, whose range is *xsd:dateTime*.

benchmarkOntology. The *Document* class represents a document containing one ontology. A document can be further described with the following properties having *Document* as domain: *documentURL* (the URL of the document), *ontologyName* (the ontology name), *ontologyNamespace* (the ontology namespace), and *representationLanguage* (the language used to implemented the ontology).

The *Benchmark* class represents a benchmark to be executed. A benchmark can be further described with the following properties having *Benchmark* as domain: *id* (the benchmark identifier); *usesDocument* (the document that contains one ontology used as input); *interchangeLanguage* (the interchange language used); *author* (the benchmark author); and *version* (the benchmark version number).

resultOntology. The *Tool* class represents a tool that has participated as origin or destination of an interchange in a benchmark. A tool can be further described with the following properties having *Tool* as domain: *toolName* (the tool name), and *toolVersion* (the tool version number).

The *Result* class represents a result of a benchmark execution. A result can be further described with the following properties having *Result* as domain: *ofBenchmark* (the benchmark to which the result corresponds); *originTool* (the tool origin of the interchange); *destinationTool* (the tool destination of the interchange); *execution*, *executionStep1*, *executionStep2* (if the whole interchange, the first and the second

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

[6] `http://knowledgeweb.semanticwe.org/benchmarking_interoperability/owl/benchmarkOntology.owl`

[7] `http://knowledgeweb.semanticwe.org/benchmarking_interoperability/owl/resultOntology.owl`

steps are carried out without any execution problem, respectively); *interchange*, *interchangeStep1*, *interchangeStep2* (if the ontology has been interchanged correctly from the original tool to the destination tool, in the first and second steps with no addition or loss of information, respectively); *informationAdded*, *informationAddedStep1*, *informationAddedStep2* (the triples added in the whole interchange, the first and the second steps, respectively); *informationRemoved*, *informationRemovedStep1*, *informationRemovedStep2* (the triples removed in the whole interchange, the first and the second steps, respectively); and finally, *timestamp* (the date and time when the benchmark is executed).

7 Implementation of the Evaluation Infrastructure

IBSE (Interoperability Benchmark Suite Executor) is the interoperability evaluation infrastructure that automates the execution of the experiments to be performed in the OWL Interoperability Benchmarking and that provides HTML summarized views of the obtained results. This is performed in the following three consecutive steps:

1. **To generate machine-readable benchmark descriptions from a group of ontologies**. In this step, a RDF file is generated including one benchmark for each ontology from a group of ontologies located in a URI and the vocabulary of the *benchmarkOntology* ontology. This can be skipped if benchmark descriptions are already available.
2. **To execute the benchmarks**. In this step, each benchmark described in the RDF file is executed between each pair of tools, being one tool the origin of the interchange and the other the destination of the interchange. The results are stored in a RDF file using the vocabulary of the *resultOntology* ontology.
   Once we have the original, intermediate and final files with the corresponding ontologies, we extract the execution results by comparing each of these ontologies as shown in Section 4. This comparison and its output depend on an external ontology comparer. The current implementation uses the *diff* methods of a RDF(S) comparer (rdf-utils[8] version 0.3b) and of an OWL comparer (KAON2 OWL Tools[9] version 0.27). Nevertheless, the implementation permits inserting other comparers.
3. **To generate HTML files with different visualizations of the results**. In this step, three HTML files are generated. One of them shows the original, intermediate and final ontologies obtained in a benchmark execution whereas the other two summarize the *Execution*, *Interchange*, *Information added*, and *Information lost* results contained in the RDF result files. These two HTML files show, for each benchmark, the results of the final interchange and of the intermediate steps (*Step 1* and *Step 2*), as Figure 2 illustrates. The only difference between these two visualizations is that one depicts the count of triples inserted or removed to provide a quick summary of the results, the other, all the triples that have changed. We also distinguish between changes in the ontology formal model and in its documentation (its annotation properties). Regarding the triples inserted or removed, the tables show clearly

[8] http://wymiwyg.org/rdf-utils/
[9] http://owltools.ontoware.org/

Id	Final	Step 1 (Jena)	Step 2 (WebODE)
B01	**Interchange**=DIFFERENT **Execution**=OK **Inserted**: Annotations: 1, Others: 0	**Interchange**=SAME **Execution**=OK	**Interchange**=DIFFERENT **Execution**=OK **Inserted**: Annotations 1, Others: 0

Id	E01
Final	**Interchange**=DIFFERENT **Execution**=OK **Inserted**: ns=http://www.example.org/ISA01# ns:Person rdfs:label "Person^^xsd:string".
Step 1 (Jena)	**Interchange**=SAME **Execution**=OK
Step 2 (WebODE)	**Interchange**=DIFFERENT **Execution**=OK **Inserted**: ns=http://www.example.org/ISA01# ns:Person rdfs:label "Person^^xsd:string".

Fig. 2. Example of a result of a benchmark execution

the annotation properties both in the triple count by counting them separately and in the triple list, by showing them in a different colour.

The only requirements for executing the evaluation infrastructure are to have a Java Runtime Environment and the tools that participate in the experiment installed, with their corresponding implementations in the IBSE evaluation infrastructure. The benchmarking web page contains links to the source code and binaries of IBSE[10] and links to the RDF file with the description of the benchmarks to be executed in each tool[11].

The only operation that a tool has to perform to participate in the experiment, as seen in Section 4, is to import an ontology from a file and to export the imported ontology into another file. To insert a new tool in the evaluation infrastructure only one method from the *ToolManager* interface has to be implemented: *void ImportExport(String importFile, String exportFile, String ontologyName, String namespace, String language)*. This method receives as input parameters the location of the file with the ontology to be imported, the location of the file where the exported ontology has to be written, the name of the ontology, the namespace of the ontology, and the representation language of the ontologies respectively. This method has already been implemented for the Jena[12] ontology repository and for the WebODE[13] ontology development tool.

8 Execution Sample

We have performed a sample experiment using Jena and WebODE. After running the experiment, we obtained four sets of results from all the possible ways of interchanging

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

[11] http://knowledgeweb.semanticweb.org/benchmarking_interoperability/owl/OIBS.rdf

[12] http://jena.sourceforge.net/

[13] http://webode.dia.fi.upm.es/

ontologies between these tools: Jena with itself, Jena with WebODE, WebODE with Jena, and WebODE with itself.

The HTML files with the interoperability results of Jena and WebODE provide not just data about the interoperability of the tools but some hints on improving the evaluation infrastructure. Table 1 presents a summary of this first set of results including a richer classification of these results after some in-depth analyses.

Table 1. Interoperability results of Jena (J) and WebODE (W)

		J-J	J-W	W-J	W-W
Correct interchange	Ontology interchanged correctly	77	0	0	0
	Changes only in blank node identifiers	5	0	0	0
Incorrect interchange	Only annotation properties are inserted	0	14	12	12
	Triples are inserted or removed	0	49	51	44
Failed execution	Execution failed in Step 1	0	0	16	16
	Execution failed in Step 2	0	13	0	6
	Comparer launches an exception	0	6	3	4

As Table 1 shows, Jena interchanges correctly all the ontologies with itself, whereas WebODE has some problems when dealing with OWL ontologies, since it crashes when processing them or makes the ontology comparer crash.

The results depend on the correct working of the ontology comparer. At first sight, the results inform that Jena is not capable of interchanging five ontologies but, looking closer at the results, it can be observed that these ontologies are the same, the discrepancy lays in that the OWL Tools ontology comparer does not consider two ontologies to be the same if they do not have the same blank node identifiers.

It can be noted that when WebODE exports ontologies, it generates labels and comments with the name of the concepts and properties in the ontology, inserting new information in the ontology and not providing the same ontology as a result.

9 Conclusions

We are currently organizing the benchmarking of interoperability of ontology tools using OWL as interchange language. To support the automatic execution of the experiment and the analysis of the results we have developed IBSE, the evaluation infrastructure presented in this paper, which can be used either by the benchmarking participants or by anyone interested in evaluating the interoperability of their tools.

At the time of writing this paper, we do not have a definitive list of the tools participating in the benchmarking and, on the other hand, the evaluation over the tools has not started. Therefore, we do not have conclusive results from any tool. The evaluation infrastructure is, nevertheless, currently under development and these first results provide valuable feedback for continuing the work.

One change that should be implemented is the detection of comparer errors, although this can be quite comparer-specific, and the inclusion of these errors into the results. To facilitate the analysis we also need to enhance result visualization by providing graphical visualizations and statistics of the whole results.

In the case of tools whose internal knowledge model does not correspond to the interchange language, the analysis of the results is not straightforward and sometimes triples are inserted or removed as it was intended by their developers, but this correct functioning is difficult to evaluate or to distinguish in the current results.

Another issue that is not clear enough with the current results is which components or combinations of components are interchanged correctly and which are not. This issue could be solved by extending the description of the ontologies with this data.

The evaluation infrastructure presented in this paper also considers the current evolution of OWL. It allows evaluating OWL 1.1 implementations either by using any set of OWL 1.1 ontologies to evaluate their interoperability or by using the current ontology dataset to evaluate their backward compatibility with the OWL recommendation.

Acknowledgments

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. IEEE-STD-610: ANSI/IEEE Std 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology. IEEE (1991)
2. Sure, Y., Corcho, O., eds.: Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003). Volume 87 of CEUR-WS., Florida, USA (2003)
3. García-Castro, R., Sure, Y., Zondler, M., Corby, O., Prieto-González, J., Bontas, E.P., Nixon, L., Mochol, M.: D1.2.2.1.1 Benchmarking the interoperability of ontology development tools using RDF(S) as interchange language. Technical report, Knowledge Web (2006)
4. García-Castro, R., Maynard, D., Wache, H., Foxvog, D., González-Cabero, R.: D2.1.4 specification of a methodology, general criteria and benchmark suites for benchmarking ontology tools. Technical report, Knowledge Web (2004)
5. García-Castro, R., Gómez-Pérez, A.: Guidelines for Benchmarking the Performance of Ontology Management APIs. In: Proceedings of the 4th International Semantic Web Conference (ISWC2005). Number 3729 in LNCS, Galway, Ireland, Springer-Verlag (2005) 277–292
6. Guo, Y., Pan, Z., Heflin, J.: LUBM: A Benchmark for OWL Knowledge Base Systems. Journal of Web Semantics 3(2) (2005) 158–182
7. Ma, L., Yang, Y., Qiu, Z., Xie, G., Pan, Y., Liu, S.: Towards a complete OWL ontology benchmark. In Sure, Y., Domingue, J., eds.: Proceedings of the 3rd European Semantic Web Conference (ESWC 2006). Volume 4011 of LNCS., Budva, Montenegro (2006) 125–139
8. David, S., García-Castro, R., Gómez-Pérez, A.: 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 (2006)