

Benchmark Suites for Improving the RDF(S) Importers and Exporters of Ontology Development Tools

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. Interoperability is the ability of two or more systems to interchange information and to use the information that has been interchanged. Nowadays, interoperability between ontology development tools is low. Therefore, to assess and improve this interoperability, we propose to perform a benchmarking of the interoperability of ontology development tools using RDF(S) as the interchange language. This paper presents, on the one hand, the interoperability benchmarking that is currently in progress in Knowledge Web¹ and, on the other, the benchmark suites defined and used in this benchmarking.

1 Introduction

The number of users of ontology development tools is ever increasing. These users come from academia or from industry, and might have or not deep knowledge on ontology engineering. Each ontology development tool provides a different set of functionalities and the user that develops an ontology prefers one ontology development tool over the others. Hence, users need to interchange ontologies from one ontology development tool to another.

Nowadays, users of ontology development tools do not know whether ontologies can be properly interchanged between two ontology development tools and, if so, which are the consequences of this interchange, such as addition or loss of knowledge. This leads to a slower uptake of ontology development tools by end users, both in the academia and the industrial world. Moreover, the experimentation carried out so far [1] has demonstrated that the degree of ontology interchange between ontology development tools is low.

One of the main goals of the Knowledge Web Network of Excellence is to support the industrial applicability of ontology technology. This involves assessing and improving several types of ontology technology: ontology development tools, alignment tools, annotation tools, querying and reasoning services, and semantic web service technology.

¹ <http://knowledgeweb.semanticweb.org/>

In order to assess and improve the interoperability of ontology tools, the first task to perform in Knowledge Web is benchmarking the interoperability of ontology development tools by evaluating their RDF(S)² importers and exporters, and this is what we present in this paper. Participation in the benchmarking is open to any organization and its current status and its results are publicly available³. At the end of the benchmarking process, we will get public results with detailed information about the current interoperability of ontology development tools. This benchmarking will also provide us with mechanisms that can be used to evaluate the RDF(S) importers and exporters of other Semantic Web technology (i.e. mapping tools, annotation tools, etc.).

According to [2], ontology development tools can interoperate in four ways: by mapping ontologies in the source tool to others in the target tool, by translating ontologies to a single pivot language, by translating ontologies to one language in a layered architecture of languages, and by a generalisation of the pivot and the layered approaches that does not require either a fixed pivot language or a fixed layering of languages. This paper only covers the pivot approach, that is, interoperability in terms of interchanging ontologies using a common interchange language. Therefore, the interoperability depends on the correct working of the importers and exporters from and to the different languages.

This paper is structured as follows. Section 2 presents other evaluation initiatives related to this work and the differences found. Section 3 describes briefly the benchmarking methodology being used; Sections 4 and 5 state how this methodology was instantiated to our own case. Section 6 defines how the benchmark suites used in the benchmarking were defined as well as how they are used. Section 7 shows how these benchmark suites have been evaluated according to the desirable properties of benchmark suites. Finally, Section 8 presents the conclusions derived from this work and future lines of work.

2 Related Work

This section introduces the terms benchmark and benchmarking, because they are frequently used in the paper, and explains the benefits of benchmarking over performing tool evaluations. It also presents other related evaluation initiatives and the differences between them and our approach.

2.1 Benchmark and Benchmarking

Benchmarking is a continuous process for improving products, services and processes by systematically evaluating and comparing them to those considered to be the best. This definition, adapted from the business management community [3], is used by some authors in the Software Engineering community [4] while others consider benchmarking as a software evaluation method [5].

² <http://www.w3.org/TR/rdf-schema/>

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

A **benchmark**, by contrast, is a test that measures the performance of a system or subsystem on a well-defined task or set of tasks [6]. However, Sim et al. [7] propose to measure also tools and techniques for comparing their performance.

The reason for benchmarking ontology tools instead of just evaluating them is to obtain several benefits from benchmarking that cannot be obtained from tool evaluations. As Figure 1 illustrates, the evaluation of a tool shows us the weaknesses of the tool or its compliance to quality requirements. If several tools are involved in the evaluation, we also obtain a comparative analysis of these tools and recommendations for users of these tools. When benchmarking several tools, besides all the benefits commented, we gain continuous improvement of the tools, recommendations for developers on the practices used when developing these tools and, from these practices, those that can be considered best practices.

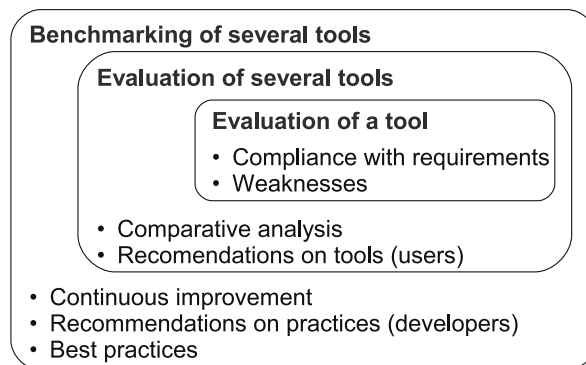


Fig. 1. Benchmarking benefits

2.2 Related Evaluations

In this section, we briefly present two evaluation initiatives related to this work. The first is a benchmark suite for evaluating RDF(S) usage, and the second is a previous evaluation of the interoperability of ontology development tools.

RDF(S) Test Cases. The RDF Test Cases⁴ were created by the W3C RDF Core Working Group. These tests check the correct usage of the tools that implement RDF knowledge bases and illustrate the resolution of different issues considered by the Working Group.

The RDF Test Cases could also be used for evaluating RDF(S) importers but, while they provide examples for, and clarification of, the normative definition of the language, our approach aims for an exhaustive evaluation of RDF(S) importers. Another difference is that we distinguish between the benchmarks that depend on the RDF(S) knowledge model and those that depend on the RDF syntax used. Moreover, we only consider valid input ontologies while the RDF Test Cases consider erroneous input ontologies and entailment benchmarks.

⁴ <http://www.w3.org/TR/rdf-testcases/>

EON 2003 Interoperability Experiments. The Second International Workshop on Evaluation of Ontology-based Tools (EON 2003) had as main topic the evaluation of the interoperability of ontology-based tools [1]. The main reasons for benchmarking the interoperability of ontology tools two years later are:

- Interoperability is still a great problem in the Semantic Web not solved yet.
- Each experiment presented in the workshop involved only a few tools.
- Some experiments evaluated export functionalities, while others evaluated import functionalities or interoperability.
- No systematic evaluation of the tools was performed since ontology tool developers were just asked to model and interchange a domain ontology. Each experiment used different test strategies, different interchange languages, and different principles for modelling ontologies. Therefore, only specific comments and recommendations were made but not general ones.

3 Benchmarking Methodology for Ontology Tools

The benchmarking methodology for ontology tools is composed of a benchmarking iteration that is repeated forever. Each iteration is composed of three phases (*Plan*, *Experiment* and *Improve*) and ends with a *Recalibration* task:

- **Plan phase.** Its main goal is to produce a document with a detailed proposal for benchmarking. It will be used as a reference document during benchmarking and should include all the relevant information about it: its goal, benefits and costs; the tool (and its functionalities) to be evaluated; the metrics to be used when evaluating these functionalities; and the people involved in the benchmarking. The last tasks of this phase are related to the search for other participant organizations and to the agreement on the benchmarking proposal (both with the organization management and with the other organizations) and on the benchmarking planning.
- **Experiment phase.** In this phase, the organizations must define and execute the evaluation experiments for each of the tools that participate in the benchmarking. The evaluation results must be compiled and analysed, determining the practices that lead to these results and identifying which of them can be considered as best practices.
- **Improve phase.** The first task of this phase is the writing of the benchmarking report, which must include: a summary of the process followed, the results and conclusions of the experimentation, recommendations for improving the tools, and the best practices found during the experimentation. The benchmarking results must be communicated to the participant organizations; afterwards, in several improvement cycles, the tool developers should improve their tools and monitor this improvement.

While the three phases mentioned above are devoted to the improvement of the tools, the goal of the *Recalibration* task is to improve the benchmarking process itself using the lessons learnt while performing it.

Sections 4 and 5 present how we use this methodology for benchmarking the interoperability of ontology development tools.

4 Plan Phase

Goals Identification. The authors of this paper took the role of benchmarking initiators and prepared the benchmarking, carrying out the first tasks of the benchmarking process.

The goals for benchmarking the interoperability of ontology development tools are related to the benefits pursued through it, and these are:

- To evaluate and improve the interoperability of ontology development tools.
- To obtain recommendations on the interoperability of these tools for users.
- To obtain a deep understanding of the practices used to develop the importers and exporters of these tools.
- To extract from these practices those that can be considered best practices when developing importers and exporters.
- To create consensual processes for evaluating the interoperability of ontology development tools.

Tool and Metrics Identification. The authors of this paper decided to participate in the benchmarking with WebODE [8], since this is the ontology engineering platform being developed by this research group.

Of the different evaluation criteria that can be considered when evaluating ontology development tools, i.e., performance, scalability, interoperability, robustness, etc.; we contemplated only interoperability. An approach for benchmarking the performance and scalability of ontology development tools can be found in [9]. Of the different ways of dealing with interoperability, we have centered our focus on the interoperability of ontology development tools using an interchange language. In our first approach, the language used was RDF(S), in its RDF/XML syntax.

However, we cannot assess interoperability using an interchange language without assessing first the import and export of ontologies to that language. Therefore, the functionalities relevant to the benchmarking are the RDF(S) importers and exporters of the ontology development tools, while the evaluation criteria that will be used for evaluating these tools are:

- The components of the knowledge model of an ontology development tool that can be interchanged with another tool using RDF(S) as interchange language.
- The information added or lost when interchanging these components.

Participant Identification. As WebODE is being developed by the Ontology Engineering Group, it seemed quite straightforward to identify and contact the members of the organization involved with WebODE's RDF(S) importers and exporters and then to select, from this very group, the members of the benchmarking team.

Proposal Writing. The benchmarking proposal, which is now being used as a reference along the benchmarking, did not take the form of a paper document,

but of a web page⁵, which is publicly available and includes all the relevant information about the benchmarking: motivation, goals, benefits and costs, tools and people involved, planning, related events, and a complete description of the experimentation and the benchmark suites.

Management Involvement. When analysing the benchmarking proposal, the managers of the Ontology Engineering Group agreed on the continuity of the benchmarking and on the allocation of future resources.

Partner Selection. To find other best-in-class organizations willing to participate in the benchmarking, the following actions were taken:

- To research different ontology development tools, both freely available and commercial ones, that could export and import to and from RDF(S) and to contact the organizations that develop them.
- To announce the interoperability benchmarking and to call for participation through the main mailing lists of the Semantic Web area and on those lists specific to ontology development tools.

When writing this paper, five tools are participating in the benchmarking, of these four are ontology development tools: KAON⁶, OntoStudio⁷, Protégé⁸ using its RDF backend, and WebODE⁹; and one is a RDF engine: Corese¹⁰. In most cases, benchmarking is performed by the developers of the tools.

Planning and Resource Allocation. A plan for the full duration of the benchmarking was not defined since it was decided to plan the benchmarking phase by phase. Then, each of the organizations assigned enough people to perform the benchmarking.

5 Experiment Phase

Experiment Definition. Evaluating the interoperability of ontology development tools using RDF(S) for ontology interchange requires that the importers and exporters from/to RDF(S) of these tools work accurately in order to interchange ontologies correctly. Therefore, the planning for the experimentation included three consecutive stages, shown in Figure 2:

- **Agreement stage.** The quality of the benchmark suites to be used is essential for the results of the benchmarking. Therefore, the first step in the experimentation is to agree on the definition of these benchmark suites, which will be common for all the tools. Section 6 deals with the definition and use of these benchmark suites.

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

⁶ <http://kaon.semanticweb.org/>

⁷ http://www.ontoprise.de/content/e3/e43/index_eng.html

⁸ <http://protege.stanford.edu/>

⁹ <http://webode.dia.fi.upm.es/>

¹⁰ <http://www-sop.inria.fr/acacia/soft/corese/>

- **Evaluation stage 1.** The RDF(S) importers and exporters of the ontology development tools are evaluated with the agreed versions of the benchmark suites.
- **Evaluation stage 2.** Once the RDF(S) importers and exporters have been evaluated, a second stage will cover the evaluation of the ontology interchange between ontology development tools.

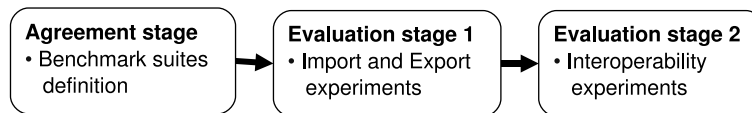


Fig. 2. Experimentation Phases

Experiment Execution. When writing this paper, the benchmarking participants are in the *Evaluation stage 2*, performing the interoperability experiments on the tools after having reached an agreement on the benchmark suites and having performed the import and export experiments. By the beginning of June 2006, the experimentation will be finished, and the results obtained will be available in the benchmarking web page.

Experiment Result Analysis. Once the results of the experimentation on each tool are available, the participants will analyse them as well as the practices that lead to these results. They will also attempt to identify among the practices found whether some of them can be considered best practices.

6 Definition of the Benchmark Suites

This section describes the three benchmark suites used in the benchmarking: the RDF(S) Import Benchmark Suite and the RDF(S) Export Benchmark Suite used in the *Evaluation stage 1*, and the RDF(S) Interoperability Benchmark Suite to be used in the *Evaluation stage 2*.

6.1 RDF(S) Import Benchmark Suite

Our goal when defining the RDF(S) Import Benchmark Suite was to perform an exhaustive evaluation of the RDF(S) import capabilities of the ontology development tools, testing the import of RDF(S) ontologies.

The benchmark suite is composed of benchmarks that import an ontology that models a simple combination of the components of the RDF(S) knowledge model (classes, properties, instances, etc.) [10]. Assessing the import of real, large or complex ontologies can be useless if we do not know if the importer can deal correctly with simple ones. Besides, it is easier to find problems in simple cases than in complex ones. These benchmarks can depend on the RDF(S) knowledge model or on the RDF(S) syntax chosen:

- **Benchmarks that depend on the knowledge model.** These benchmarks check the import of ontologies with different combinations of the RDF(S) knowledge model components. Since the checking of the different syntaxes of the selected RDF serialisation is performed in another group of benchmarks, the syntax selected is one that is easily understood by humans. These benchmarks evaluate the import of single components, all the possible combinations of two components with a property, combinations of more than two components usually appearing together in RDF(S) graphs such as the properties that have both domain and range (*rdf:Property* with *rdfs:domain* and *rdfs:range*); the statements that have subject, predicate and object (*rdf:Statement* with *rdf:subject*, *rdf:predicate* and *rdf:object*); and the definitions of lists (*rdf:List* with *rdf:first*, *rdf:rest* and *rdf:nil*).
- **Benchmarks that depend on the syntax.** These benchmarks check the import of ontologies with the different variants of the RDF/XML syntax, as stated in the RDF/XML Syntax Specification [11] since this is the most commonly used by ontology editors for importing and exporting ontologies.
 - Different syntax of URI references: absolute URI references, URI references relative to a base URI, URI references transformed from *rdf:ID* attribute values, and URI references relative to an *ENTITY* declaration.
 - Language identification attributes (*xml:lang*) in tags.
 - Abbreviations of empty nodes, multiple properties, typed nodes, string literals, blank nodes, containers, collections, and statements.

In the case of evaluating the import of RDF(S) using other syntax (N3, N-Triples, etc.), only this group of benchmarks should be redefined.

As RDF(S) does not impose any restriction on the combinations of its components, the number of resulting benchmarks is huge (more than 4000) and the benchmark suite has to be pruned, as seen in [12], according to its intended use and to the kind of tools that it is supposed to evaluate: ontology development tools. Therefore, we only considered the RDF(S) components most frequently used for modelling ontologies in these tools: *rdfs:Class*, *rdf:Property*, *rdfs:Literal*, *rdf:type*, *rdfs:domain*, *rdfs:range*, *rdfs:subClassOf*, and *rdfs:subPropertyOf*. The rest of the RDF(S) components have not been dealt with.

Table 1 shows the categories of the benchmark suite, which contains 72 benchmarks, with the number of benchmarks and the components used in each category. A detailed description of such benchmarks can be found in a web page¹¹.

The definition of each benchmark in the benchmark suite, as Table 2 shows, includes the following fields:

- An identifier for tracking the different benchmarks.
- A description of the benchmark in natural language.
- A graphical representation of the ontology to be imported in the benchmark.
- A file containing the ontology to be imported in the RDF/XML syntax.

¹¹ http://knowledgeweb.semanticweb.org/benchmarking_interoperability/rdfs_import_benchmark_suite.html

Table 1. Categories of the import benchmarks

Category	No.	Components used
Class	2	<i>rdfs:Class</i>
Metaclass	5	<i>rdfs:Class, rdf:type</i>
Subclass	5	<i>rdfs:Class, rdfs:subClassOf</i>
Class and property	6	<i>rdfs:Class, rdf:Property, rdfs:Literal</i>
Property	2	<i>rdf:Property</i>
Subproperty	5	<i>rdf:Property, rdfs:subPropertyOf</i>
Property with domain and range	21	<i>rdfs:Class, rdf:Property, rdfs:Literal, rdfs:domain, rdfs:range</i>
Instance	3	<i>rdfs:Class, rdf:type</i>
Instance and property	9	<i>rdfs:Class, rdf:type, rdf:Property, rdfs:Literal</i>
Syntax and abbreviation	14	<i>rdfs:Class, rdf:type, rdf:Property, rdfs:Literal</i>

Table 2. An example of a benchmark definition

Identifier	I14
Description	Import one class that has the same property with several other classes
Graphical representation	<pre> graph LR C1((C1)) -- prop1 --> C2((C2)) C1 -- prop1 --> C3((C3)) </pre>
RDF/XML file	<pre> <rdf:RDF xmlns="http://www.w3.org/2000/01/rdf-schema#" xmlns:g1="http://www.test.org/graph14#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"> <Class rdf:about="http://www.test.org/graph14#class1"> <g1:prop1 rdf:resource="http://www.test.org/graph14#class2"/> <g1:prop1 rdf:resource="http://www.test.org/graph14#class3"/> </Class> <Class rdf:about="http://www.test.org/graph14#class2"/> <Class rdf:about="http://www.test.org/graph14#class3"/> </rdf:RDF> </pre>

The steps for executing each import benchmark are the following:

1. To model into the ontology development tool the expected ontology that results from importing the RDF(S) ontology .
2. To import the file with the RDF(S) ontology into the tool.
3. To compare the imported ontology with the expected ontology and to check whether they are equivalent.

Although these steps can be performed manually, performing them (or part of them) automatically is highly advised when dealing with many benchmarks, especially for comparing the expected and imported ontologies.

The evaluation criteria used for the benchmark suite are:

- **Modelling (YES/NO).** The tool can model the ontology components described in the benchmark.

Table 3. An example of the result of a benchmark execution

Tool	Id	Added	Lost	Model	Execute	Comments
Protégé	I50	class	-	NO	OK	Undefined resources with instances are imported as classes.
WebODE	I50	-	instance	NO	OK	Instances of undefined resources are not imported.

- **Execution (OK/FAIL).** The execution of the benchmark is normally carried out without any problem, and the tool always produces the expected result. When a failed execution occurred, the benchmarking participants were asked to provide information for obtaining the practices used when developing the RDF(S) importers. The information required was the following:
 - The reasons for failing the benchmark execution.
 - If the tool was corrected to pass a benchmark, the changes performed.
- **Information added or lost.** The information added or lost in the ontology interchange when executing the benchmark.

Table 3 shows an example with the results of executing benchmark I50 (Import one instance of a resource, without the resource definition) in two tools. While both tools cannot model an instance of an undefined resource, they produce the expected result, one inserting information and the other losing it.

6.2 RDF(S) Export Benchmark Suite

When defining the RDF(S) Export Benchmark Suite, our goal was to perform an evaluation of the RDF(S) export capabilities of ontology development tools by testing the export of ontologies modelled in these tools.

The benchmark suite for evaluating RDF(S) exporters is composed of benchmarks that export a single ontology with a simple combination of the components of the knowledge models of the tools.

The composition of the RDF(S) Export Benchmark Suite is similar to the composition of the import one. Instead of taking as input the knowledge model of RDF(S), we took as input a common core of the knowledge modelling components that is very frequently used in ontology development tools: classes and class hierarchies, object and datatype properties, instances, and literals.

Table 4 shows the categories of the benchmark suite, that contains 52 benchmarks, with the number of benchmarks and the components used in each category. A detailed description of such benchmarks can be found in a web page¹².

The definition of each benchmark in the benchmark suite, as Table 5 shows, includes the following fields:

- An identifier for tracking the different benchmarks.
- A description of the benchmark in natural language.

¹² http://knowledgeweb.semanticweb.org/benchmarking_interoperability/rdfs_export_benchmark_suite.html

- A graphical representation of the ontology to be exported by the tool.
- The instantiation of the ontology described in the benchmark for each of the participating tools, using the vocabulary and components of these tools.

The steps for executing each export benchmark are the following:

1. To define in RDF(S) the expected ontology resulting from exporting the ontology.
2. To model into the tool the ontology described in the benchmark.
3. To export the ontology modelled using the tool to RDF(S).
4. To compare the exported RDF(S) ontology with the expected RDF(S) ontology to check whether they are equivalent.

As in the case of the import benchmark suite, some automatic mean of performing these steps is highly advisable.

The evaluation criteria used for the export benchmark suite are the same as those from the import benchmark suite, that is, *Modelling*, *Execution* and *Information added or lost*. The only difference with the import criteria is that,

Table 4. Categories of the export benchmarks

Category	No.	Components used
Class	2	<i>class</i>
Metaclass	5	<i>class, instanceOf</i>
Subclass	5	<i>class, subclassOf</i>
Class and object property	4	<i>class, object property</i>
Class and datatype property	2	<i>class, datatype property, literal</i>
Object property	13	<i>object property</i>
Datatype property	9	<i>datatype property</i>
Instance	3	<i>class, instanceOf</i>
Instance and object property	7	<i>class, instanceOf, object property</i>
Instance and datatype property	2	<i>class, instanceOf, datatype property, literal</i>

Table 5. An example of a benchmark definition

Identifier	E09
Description	Export one class that is subclass of several classes
Graphical representation	
WebODE's instantiation	Export one concept that is subclass of several concepts
Protégé's instantiation	Export one class that is subclass of several classes
...	...

as there may be a benchmark that defines an ontology that cannot be modelled in a certain tool, that benchmark cannot be executed in the tool, being the *Execution* result *N.E.* (Non Executed). In the import benchmark suite, even if a tool cannot model some components of the ontology, it should be able to import correctly the rest of the components.

6.3 RDF(S) Interoperability Benchmark Suite

Our goal when defining the RDF(S) Interoperability Benchmark Suite was to evaluate the interoperability of ontology development tools by testing the interchange of ontologies from one origin tool to a destination one, and vice versa.

We considered the interchange of a common core of the knowledge modelling components most frequently used for modelling ontologies. These are: classes and class hierarchies, object and datatype properties, instances, and literals. As these components are the same as those in the RDF(S) Export Benchmark Suite, the Interoperability Benchmark Suite is identical to the RDF(S) Export Benchmark Suite and has the same benchmark definitions as presented in Section 6.2.

The interoperability will be checked between each pair of tools. As the RDF(S) exported files of all the tools will be available from the export experiments of the *Experiment stage 1*, participants will not have to export these ontologies again, they will just have to import the exported files into their tools.

The steps for executing each interoperability benchmark are the following:

1. To define in the destination tool the expected ontology resulting from interchanging the ontology.
2. To import the RDF(S) file exported by the origin tool into the destination tool.
3. To compare the interchanged ontology with the expected ontology and to check whether they are equivalent.

The evaluation criteria used for the interoperability benchmark suite are the same as those from the export benchmark suite, that is, *Modelling*, *Execution* and *Information added or lost*.

7 Evaluation of the Benchmark Suites

We have evaluated these benchmark suites according to the main desirable properties of a benchmark suite that many different authors have stated [7, 13, 14, 15]: accessibility, affordability, simplicity, representativity, portability, scalability, robustness, and consensus.

Accessibility. The complete definition of the benchmark suites as well as all the information relevant to the benchmarking are accessible to anyone in a public web page. This page will include, when available, the results obtained when executing the benchmark suites. Thus anyone can execute them and compare their results with the ones available.

Affordability. The costs of using the benchmark suites and analysing their results are mainly human resources. In order to reduce these costs and facilitate the work we provided a clear definition of the benchmark suites and templates to fill in the results.

Simplicity. The benchmark suites are simple and interpretable because we have provided different ways of defining each benchmark, i.e. in natural language, graphically, etc. These benchmark suites and their results are also clearly documented, having a common structure and use.

Representativity. Although the different benchmarks that compose the benchmark suites are not exhaustive or represent real-world ontologies, they represent the different structures of ontologies commonly used when developing these ontologies; and a first evaluation of these simple ontologies is needed before evaluating more complex ones.

Portability. The benchmark suites have been defined at a high level of abstraction, so they are not biased towards a certain tool or tools. Therefore, they can be executed on a wide variety of environments and not just on ontology development tools.

Scalability. The benchmark suites scale to work with tools at different levels of maturity. Also, as their benchmarks are grouped according to the different ontology components that they manage, it is quite easy to increase or decrease the number of benchmarks by considering new components or by taking into account only certain components of interest, respectively.

Robustness. As the results of the benchmark suites depend on the algorithms implemented to perform the import and export of ontologies, they are not influenced by factors irrelevant to the study. Furthermore, running the benchmark suites with the same version of the tools will always produce the same results.

Consensus. The benchmark suites were developed by experts in the domain of ontology translation and interoperability and were assessed and agreed on by the benchmarking partners and by the members of Knowledge Web.

8 Conclusions and Future Work

The interoperability benchmarking described in this paper is now taking place. Although the benchmarking has not reached its *Improve* phase yet, the developers have already improved their tools by correcting the bugs detected while executing the experiments.

Although the benchmarking has not finished yet, we have already learnt some lessons from it. The main one is that benchmarking is a time consuming process both to organize and to participate on it and it is also enduring. Besides, the involvement of tool developers is a primary need, as they are the most appropriate for and capable of analysing and improving the tools.

In addition to the benchmarking participants, other intended users of the work here presented and of its future results are, on the one hand, end users who want to know the current interoperability of ontology development tools before

selecting one of these tools and, on the other hand, ontology tool developers who wish to improve the interoperability of their ontology development tools or of any other ontology technology capable of interchanging ontologies by means of their RDF(S) importers and exporters.

The RDF(S) Import Benchmark Suite can be used to evaluate the RDF(S) import capabilities of any tool, while the RDF(S) Export and Interoperability Benchmark Suites can be used to evaluate any kind of interoperability between ontology development tools since these are not dependant on the interchange language.

Once we will have the benchmarking results, we will be able to provide different kind of information about the interoperability of ontology development tools for different groups of people. For example, from an ontology modelled in a certain tool, we can obtain information about the possibility of interchanging that ontology between that tool and any other tool.

The benchmarking web page will also provide mechanisms for updating the interoperability results when the tools are improved, or for inserting the results of a new tool.

Based on the structure and the definition of the benchmark suites, other benchmark suites can be defined to consider the evaluation of interoperability using other languages for interchanging ontologies. We have also started to organise a benchmarking activity¹³, similar to this, for benchmarking the interoperability of ontology development tools using OWL¹⁴ as the interchange language.

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 all the people participating in the benchmarking: Olivier Corby, York Sure, Moritz Weiten, and Markus Zondler. Thanks to Rosario Plaza for reviewing the grammar of this paper.

References

1. 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)
2. Euzenat, J., Stuckenschmidt, H.: The ‘Family of Languages’ approach to semantic interoperability. In Omelayenko, B., Klein, M., eds.: Knowledge transformation for the semantic web. IOS press, Amsterdam (NL) (2003) 49–63
3. Spendolini, M.: The Benchmarking Book. AMACOM, New York, NY (1992)

¹³ http://knowledgeweb.semanticweb.org/benchmarking_interoperability/owl/

¹⁴ <http://www.w3.org/TR/owl-features/>

4. Wohlin, C., Aurum, A., Petersson, H., Shull, F., Ciolkowski, M.: Software inspection benchmarking - a qualitative and quantitative comparative opportunity. In: Proceedings of 8th International Software Metrics Symposium. (2002) 118–130
5. Kitchenham, B.: DESMET: A method for evaluating software engineering methods and tools. Technical Report TR96-09, Department of Computer Science, University of Keele, Staffordshire, UK (1996)
6. Sill, D.: comp.benchmarks frequently asked questions version 1.0 (1996)
7. Sim, S., Easterbrook, S., Holt, R.: Using benchmarking to advance research: A challenge to software engineering. In: Proceedings of the 25th International Conference on Software Engineering (ICSE'03), Portland, OR (2003) 74–83
8. Arpírez, J., Corcho, O., Fernández-López, M., Gómez-Pérez, A.: WebODE in a nutshell. *AI Magazine* 24 (2003) 37–47
9. 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
10. Brickley, D., Guha, R.V. (editors): RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004 (2004)
11. Beckett, D. (editor): RDF/XML Syntax Specification (Revised). W3C Recommendation 10 February 2004 (2004)
12. García-Castro, R., Gómez-Pérez, A.: A method for performing an exhaustive evaluation of RDF(S) importers. In: Proceedings of the Workshop on Scalable Semantic Web Knowledge Based Systems (SSWS2005). Number 3807 in LNCS, New York, USA, Springer-Verlag (2005) 199–206
13. Bull, J.M., Smith, L.A., Westhead, M.D., Henty, D.S., Davey, R.A.: A methodology for benchmarking java grande applications. In: Proceedings of the ACM 1999 conference on Java Grande. (1999) 81–88
14. Shirazi, B., Welch, L., Ravindran, B., Cavanaugh, C., Yanamula, B., Brucks, R., Huh, E.: Dynbench: A dynamic benchmark suite for distributed real-time systems. In: Proc. of the 11 IPPS/SPDP'99 Workshops, Springer-Verlag (1999) 1335–1349
15. Stefani, F., Macii, D., Moschitta, A., Petri, D.: FFT benchmarking for digital signal processing technologies. In: 17th IMEKO World Congress, Dubrovnik (2003)