Defining benchmark suites with reusability in mind

Raúl García-Castro

Ontology Engineering Group, Departamento de Inteligencia Artificial.
Facultad de Informática, Universidad Politécnica de Madrid, Spain
rgarcia@fi.upm.es

Abstract. The Semantic Web technology needs to be thoroughly evaluated for providing objective results and attaining a massive technology improvement, being current evaluation and benchmarking activities over Semantic Web technology insufficient and a barrier to the maturity of this technology.
This paper is intended to provide some guidelines to develop benchmark suites with usability in mind. To this end, it presents what a benchmark suite is and what its desirable properties are and provides a questionnaire for evaluating the usability of a benchmark suite.

1 Introduction

The Semantic Web technology needs to be thoroughly evaluated for providing objective results and attaining a massive technology improvement, being current evaluation and benchmarking activities over Semantic Web technology insufficient and a barrier to the maturity of this technology.

Evaluating and benchmarking technology is costly as people do not know how to do it, there are not standard or consensuated methods to follow, and it is difficult to reuse results and lessons learnt from others.

One main need in the Semantic Web is to produce methods and tools for evaluating the technology at great scale in an easy and inexpensive way. This requires researchers to increase the quality of their evaluations, to define evaluations focusing on their usability, and to aim for collective improvements in their technology by means of benchmarking it.

This paper is intended to provide some guidelines to develop benchmark suites with usability in mind. To this end, it presents what a benchmark suite is and what its desirable properties are, and provides a questionnaire for evaluating the usability of a benchmark suite.

2 Benchmark Suites

A benchmark suite is a collection of benchmarks, being a benchmark *a test or set of tests used to compare the performance of alternative tools or techniques* [1].

A benchmark definition must include the following:

- The **context** of the benchmark, namely, which tools and which of their characteristics are measured with it.

- The **requirements** for running the benchmark, namely, the tools (hardware or software), data, or people needed.
- The **input variables** that affect the execution of the benchmark, and the values that they will take.
- The **procedure** to execute the benchmark and to obtain its results.
- The **evaluation criteria** used to interpret these results.

A benchmark suite definition must include the definition of all its benchmarks. Usually, all these benchmarks share some of their characteristics, such as the context or the requirements. In that case, these characteristics are defined at the benchmark suite level, and not individually for each benchmark.

2.1 Desirable properties of a benchmark suite

The following properties, extracted from the work of different authors [1–4], can help either to develop new benchmark suites or to assess the quality of different benchmark suites before using them.

Although a good benchmark suite should have most of these properties, each evaluation will require that some of them be considered before others.

It must also be considered that achieving a high degree of all these properties in a benchmark suite is not possible since the increment of some has a negative influence over others.

Accessibility A benchmark suite must be accessible to anyone interested. This involves providing the necessary software to execute the benchmark suite, its documentation, and its source code in order to increase transparency. The results obtained when executing the benchmark suite should be made public so that anyone can apply the benchmark suite and compare his/her results with the ones available.

Affordability Using a benchmark suite entails a number of costs, commonly human, software, and hardware resources. The costs of using a benchmark suite must be lower than those of defining, implementing, and carrying out any other experiments that fulfil the same goal. Some ways of reducing the resources consumed in the execution of a benchmark suite are: automating the execution of the benchmark suite, providing components for data collection and analysis, or facilitating its use for different heterogeneous systems.

Simplicity The benchmark suite must be simple and interpretable. It must be documented so anyone who wants to use it must be able to understand how it works and the results that it yields. If the benchmark suite is not transparent enough, its results will be questioned and it could be interpreted incorrectly. To ease the process, the elements of the benchmark suite should have a common structure and use and common inputs and outputs. Measurements should have the same meanings across the benchmark suite.

Representativity The actions that perform the benchmarks composing the benchmark suite must be representative of the actions that are usually performed on the system.

Portability The benchmark suite should be executed on a variety of environments as wide as possible, and should be applicable to as many systems as possible. It should

also be specified at a high enough level of abstraction to ensure that it is portable to different tools and techniques and that it is not biased against other technologies.

Scalability The benchmark suite should be parameterised to allow scaling the benchmarks with varying input rates. It should also scale to work with tools or techniques at different levels of maturity. It should be applicable to research prototypes and commercial products.

Robustness The benchmark suite must consider unpredictable environment behaviours and should not be sensitive to factors not relevant to the study. When running the same benchmark suite several times on a given system under the same conditions, the results obtained should not change considerably.

Consensus The benchmark suite must be developed by experts who apply their knowledge of the domain and are able to identify the key problems. It should also be assessed and agreed on by the whole community.

3 Evaluating the usability of a benchmark suite

This section provides a questionnaire for evaluating up to what extent one existing benchmark suite can be partial or totally reused in different tools and the facilities provided for this reuse.

We do not consider the way of describing the benchmark suite in the evaluation, as different benchmark suites will require different detail in their definitions. Instead, we will extract the usability information by performing some questions about the benchmark suite.

We consider that fulfilling the desirable properties of a benchmark suite defined in the previous section and an aim for continuous improvement can be an indication of the reusability of an evaluation. Therefore, from these desirable properties we have produced several questions to assess the usability of a benchmark suite.

We have defined the possible answers of each question, trying to limit these answers as much as possible. Most of the questions require a *Yes* or *No* as an answer and in just a few cases some number or comment is requested.

Table 1 shows all the questions included in the questionnaire grouped in several categories to facilitate the collection of answers. It also includes the possible answers of a question and the desirable property that is affected by each question.

Table 1: Usability evaluation questionnaire

Question	**Values**	**Property**
General		
Is the benchmark suite intended to be used one time or continuously?	Continuous /One-time	Improvement
Is improvement one of the goals of the benchmark suite?	Yes/No	Improvement
Is a common evaluation framework provided?	Yes/No	Affordability
Has any other evaluation approach been reused?	Yes/No	Consensus
Are the benchmark suite requirements clearly defined?	Yes/No	Simplicity

Table 1 – continued from previous page

Question	Values	Property
Which is the cost of using the benchmark suite?	String	Affordability
Tools/methods where the benchmark suite can be used		
In which tools/methods can it be used?	String	Portability
Does it allow the comparison of different tools?	Yes/No	Improvement
Is it portable to other tools?	Yes/No	Portability
Are the measurable characteristics of the tools clearly defined?	Yes/No	Simplicity
Which operating systems/platforms are needed to use it?	String	Portability
Input data		
Is the input data clearly defined?	Yes/No	Simplicity
Does the input data present a common structure?	Yes/No	Simplicity
Does the input data represent actions that are usually performed on the system?	Yes/No	Representativity
Is it possible to scale the input data with varying input rates?	Yes/No	Scalability
Is the input data documented?	Yes/No	Simplicity
Is the input data accessible?	Yes/No	Accessibility
Procedure for executing the benchmark suite		
Is the procedure clearly defined?	Yes/No	Simplicity
Can the procedure or part of it be executed automatically?	Yes/No	Affordability
Benchmark suite results		
Are they affected by unpredictable environment behaviours?	Yes/No	Robustness
Do they provide the development practices that lead to them?	Yes/No	Improvement
Are they the same when executing the benchmark suite several times under the same conditions?	Yes/No	Robustness
Result analysis		
Are the evaluation criteria clearly defined?	Yes/No	Simplicity
Can the results be analysed automatically or semiautomatically?	Yes/No	Affordability
Do the results provide tool improvement recommendations?	Yes/No	Improvement
Software that supports the benchmark suite		
Is there any software to support it?	Yes/No	Affordability
Is the software accessible?	Yes/No	Accessibility
Is the source code of the software accessible?	Yes/No	Accessibility
Is the source code documented?	Yes/No	Simplicity
Documentation of the benchmark suite		
Is it documented anywhere besides the paper?	Yes/No	Simplicity
Is the documentation accessible?	Yes/No	Accessibility

Table 1 – continued from previous page

Question	Values	Property
Is there a web page with information about it?	Yes/No	Accessibility
Community involvement		
How many people developed the benchmark suite?	Number	Consensus
Is the benchmark suite driven by a community?	Yes/No	Consensus
Has it been assessed and agreed on by a community?	Yes/No	Consensus
For each evaluation performed using the benchmark suite		
How many tools were evaluated?	Number	Simplicity
How many people participated in the evaluations?	Number	Simplicity
Are the evaluation results accessible?	Yes/No	Accessibility

References

1. 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
2. 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
3. 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
4. Stefani, F., Macii, D., Moschitta, A., Petri, D.: FFT Benchmarking for Digital Signal Processing Technologies. In: 17th IMEKO World Congress, Dubrovnik, Croatia (2003)