

Process Engine Benchmarking with Betsy – Current Status and Future Directions

Matthias Geiger, Simon Harrer, and Jörg Lenhard

Distributed Systems Group, University of Bamberg, Germany
{matthias.geiger, simon.harrer, joerg.lenhard}@uni-bamberg.de

Abstract. Business process management and automation has been the focus of intense research for a long time. Today, a plethora of process languages for specifying and implementing process models have evolved. Examples for such languages are established international standards, such as the Web Services Business Process Execution Language 2.0 or, more recently, the Business Process Model and Notation 2.0. Implementations of these standards which are able to execute models, so called process engines, differ in the quality of service they provide, e.g., in performance or usability, but also in the degree to which they actually implement a given standard. Selecting the “best” engine for a particular usage scenario is hard, as none of the existing process standards features an objective certification process to assess the quality of its implementations. To fill this gap, we present our work on process engine benchmarking. We discuss what has been achieved so far and point out future directions that deserve further investigation.

Keywords: business process management, process engine, BPEL, BPMN, benchmarking

1 Introduction

The field of business process management (BPM) forms an umbrella for a variety of research areas, ranging from managerial challenges to application engineering [30]. Among these fields are the modeling and automation of processes using process-aware technologies specifically dedicated to this task. This has led to the development of a multiplicity of process languages and standards [20] that can be used for specifying process models. A subset of these languages allow to specify models that are intended for execution in specific runtime environments, called *process engines*. Typically, multiple alternative engines are available for a given process language. The user of the language can implement process models as defined in the language specification and has to select the best-fitting engine for execution. Naturally, a variety of properties can form the basis for this selection, such as pricing, performance, usability, or actual language support.

The problem in this setting is that it is hard for a potential user to meaningfully judge these properties for a given set of engines, due to the inherent complexity of such software products. In general, this selection problem is not new, and exists

in similar fashion for any sufficiently sophisticated software tooling or technology, such as application servers or ERP systems. To make such a decision, there are a plethora of methods available [28], one being the analytic hierarchy process (AHP) [24]. But to apply these methods, the properties of the different alternatives need to be known. One technique to reveal these properties is *benchmarking* [25], which in this case resolves to *process engine benchmarking*. The enabling of the benchmarking of state-of-the-art engines for widely used process standards and for a comprehensive set of quality properties is the long term goal of our work. To this end, we are developing the BPEL/BPMN engine test system (*betsy*), which implements a comprehensive benchmark for process engines¹. The development of *betsy* is in progress for more than three years already and, by now, more than a dozen engines in a variety of revisions are integrated in a fully automated and reproducible benchmarking process.

In this paper, we briefly discuss related approaches for process engine benchmarking in Sect. 2. Next, we detail the current status of *betsy* and how it has evolved since its first public release in 2012 in Sect. 3 and, in Sect. 4, how we plan to evolve *betsy* even further in the future. The paper is summed up in Sect. 5.

2 Related Work

Benchmarking of IT products is not a new phenomenon and therefore there exists already lots of related work regarding this topic (e.g., [3, 10, 17, 26]). Particularly interesting is [17] which defines general requirements to be fulfilled by benchmarks to be a valid, “good” benchmark: Overall, a benchmark should measure relevant aspects to be able to give substantial answers to the investigated research questions. Benchmarking workflow engines is a relevant topic as there are no certification authorities to check claimed compliance promises. So each vendor can claim that his product is BPMN [18] or BPEL 2.0 [21] conformant without the need to actually prove it. Moreover, also other questions are relevant for users of BPM products such as: ease of installation, portability and conformance to static analysis rules which also can be compared for different products. However, [17] lists other requirements to benchmarks which might be conflictory to the aspect *relevance* as a benchmark should also be *repeatable*, *fair*, *verifiable* and *economical*. As *betsy* focuses on standard’s conformance testing those four requirements are fulfilled: *betsy* is Open Source and fully automated which allows for *repeated* test execution. Moreover, (most) tested engines are freely available and directly integrated into our approach, which allows every interested party to execute the tests without *economic* barriers on standard developer hardware. As the standard documents define all relevant aspects to be fulfilled by the implementing engines and we are building upon the same documents, *betsy* does not give an advantage to some engines but is *fair*. Due to the openness of the standard texts and our implementation the correctness of *betsy* is open to scrutiny fostering the *verifiability*.

¹ The tool itself is available at <https://github.com/uniba-dsg/betsy>.

Apart from those general works, there are some approaches regarding process engine benchmarking, which are more closely related to our work. In [2] BPEL 2.0 engines are assessed regarding their performance using the SOABench testbed. Another approach dedicated to performance benchmarking of workflow engines is the BenchFlow² project which focuses on benchmarking BPMN 2.0 engines [7, 22, 27]. Their latest work [7] evaluates the performance of two anonymized open source BPMN 2.0 engines within a container-based environment. By using container-based environments, the authors follow the recommended approach to achieve reproducible research and benchmarks [3]. Directly reusing the concepts and artifacts generated by those two approaches is not useful for the scope of our tests, as measuring performance needs a far more complex infrastructure apart from the actual engines under test to generate sensible workloads and to ensure the validity of the results [17]. Our tool *betsy* should be able to reproduce the results without economical and technological barriers, i.e., it should be executable on standard developer machines without any complex installation and configuration steps. However, as both approaches are automatically executing tests on workflow engines at least the usage of virtualization techniques such as virtual machines (e.g., with Oracle VirtualBox³) or using containers (e.g., with Docker⁴) to store and restore working engine installations is also relevant for our work.

A third notable approach [5] presents a method to evaluate BPM systems (BPMS) with the aim of selecting the best fitting BPMS for a list of requirements. In a series of case studies, the authors evaluate a large list of open source and proprietary BPMS implementing three different process languages (e.g., the XML Process Definition Language (XPDL) 2.2 [32], BPEL 2.0 [21] and BPMN 2.0 [18]). In contrast to our work, this evaluation is on a more abstract level and the actual engine evaluation is not automated.

3 The Current State of *betsy*

Betsy 2.1.0, the most recent version, has been published on September, the 29th in 2015⁵. The tool is freely available and licensed under the LGPL v3. Currently, it is capable of benchmarking three BPMN engines in thirteen different versions with 135 tests and seven BPEL engines in 16 different versions, two of them also in an in-memory configuration, with 1110 tests.

The current state of *betsy* can be described according to four dimensions: 1) process languages, 2) process engine capabilities, 3) process engine types and 4) process engine environments. The dimension *process language* is reflected in the *betsy* acronym. Although the acronym never changed, its meaning has evolved from *BPEL engine test system* to *BPEL/BPMN engine test system*, since *betsy* is able to evaluate process engines implementing the process languages BPEL [21] or BPMN [18]. The dimension *process engine capabilities* describes

² See <http://www.iaas.uni-stuttgart.de/forschung/projects/benchflow.php>

³ See <https://www.virtualbox.org/>

⁴ See <https://www.docker.com/>.

⁵ See <https://github.com/uniba-dsg/betsy/releases> for all releases.

Table 1. Status quo of *betsy* according to its four dimensions

Dimensions	Dimension Characteristic	Publications
Process Languages	BPEL 2.0	[11, 12, 14–16, 19]
	BPMN 2.0	[8]
Process Engine Capabilities	Feature Conformance	[8, 11, 12]
	Static Analysis Conformance	[15]
	Expressiveness	[12]
	Robustness	[14]
	Installability	[19]
Process Engine Types	Open Source Engines	[8, 11, 12, 14–16, 19]
	Proprietary Engines	[12]
Process Engine Environments	Bare Metal Environment	[8, 11, 12, 14, 15, 19]
	Virtual Environment	[16]

the features of the process engine that are tested. At first, *betsy* was used to evaluate feature conformance, but over time it was extended to assess the static analysis conformance, expressiveness, robustness, and installability of process engines. The third dimension *process engine types* investigates which type of process engine is put under scrutiny, being either an open source or a proprietary process engine. The last dimension *process engine environments* refers to the ability to benchmark the process engines in a bare metal environment or a virtual environment, such as in a virtual machine or a container.

4 Future Directions

To support a more meaningful selection of process engines, we aim to extend *betsy* to a process engine benchmarking platform, making it faster, more flexible, powerful, and extensible. Our plans are detailed along the four dimensions.

Dimension process language: The field of process standards is vast [20] and in constant evolution. The relevancy of a process engine benchmarking system depends on the relevancy of the language it supports. Currently, *betsy* supports BPEL 2.0 [21] and BPMN 2.0 [18]. Arguably, these two languages are sufficient at the moment, since there is no competing standard that equally targets process engine execution. XPD L [32] is also a process standard that allows for the specification of executable process models, but it is primarily meant as an interchange format. Although it is used as execution format in some engines, it is expected to be replaced for this purpose by BPMN 2.0 [4]. Therefore, there is no reason to include XPD L in benchmarking directly. Furthermore, academic approaches to process languages, such as Yet Another Workflow Language (YAWL) [29], do exist. However, YAWL is neither standardized, nor do competing implementations of YAWL, apart from the reference implementation, exist. As a result, there is no selection or comparison problem and no reason to consider the language.

Dimension process engine capability: For BPEL 2.0 engines, *betsy* already covers a large variety of engine capabilities [11, 12, 14–16, 19]. With the emergence of BPMN 2.0, we have started to benchmark the feature conformance of

BPMN 2.0 engines as well [8]. Our current goal is to fill in open gaps by benchmarking BPMN 2.0 engines for the same set of capabilities which we benchmarked for BPEL 2.0 engines, including static analysis conformance, expressiveness, installability, and robustness. The challenge here is how can the BPEL 2.0 benchmarks be ported to BPMN 2.0, effectively reusing the benchmarks to some extent. An interesting aspect is the static analysis conformance, i.e., do perform the engines static analysis of models as defined in the specification. Whereas the BPEL standard [21] is directly listing relevant static analysis checks this is not the case for BPMN. As shown in preliminary work [9] this raises issues for BPMN modeling tools which are also to be expected for BPMN engines.

In addition, it would be desirable to increase the set of already covered engine capabilities by also benchmarking performance. Performance has always been an important criterion for software selection and evaluation [31]. In a preliminary work, we evaluated existing benchmarking approaches of BPEL 2.0 engines [23] and revealed that most of them test a very small number of engines, use a limited workload model and only focus on mostly one or two metrics. Moreover, as stated in [23] for BPEL, additional challenges arise as the process engines do not support the same set of features. The same holds true for BPMN engines as well. Hence, either the benchmark's workload can only be executed on a few engines or it must be reduced to using only the features that all engines support. Apart from extending *betsy*, our current results can be used to improve the related work presented in Sect. 2: The conformance results of *betsy* can be used to determine a sensible workload leading to a benchmark which produces fair and reproducible results for all or at least the most important engines. What is more, existing test suites, e.g., of the control-flow pattern, can be used as workloads for micro-performance benchmarks. Thus, this area calls for further investigation.

Dimension process engine type: The market of process engines can currently be separated into proprietary and open source engines. In academic research, the usage of open source tooling is much more common, due to a more permissive access that does not involve costs. As a result, most analyses of process engines focus primarily on open source engines, e.g., [8, 11, 14–16, 19]. In contrast, work that explicitly compares these two types of process engines is rare, e.g., [12]. This is problematic, since, to the best of our knowledge, there is no indication that the usage of process engines is dominated by open source solutions. Instead, there are plenty of proprietary engines available, including products by large multi-national enterprises with a huge customer base world-wide. A blind spot regarding the evaluation of proprietary engines in research is problematic, as, potentially, the quality of such engines might be vastly different. An omission of these tools could result in wrong and unfounded conclusions that are not generalizable. This danger is especially valid for practical studies or case studies that depend on particular engines. It is our intention to extend *betsy* to support the benchmarking of more proprietary engines. This is most important for BPMN engines, where no proprietary implementations are supported so far. The biggest obstacle in

this endeavor is the licensing strategy of many vendors. Pseudonymization of research results, as used in [12], is a way to relieve restrictions, given academic licenses are available, but this is not always the case. By working together with the vendors, we see a possibility to publish the results nonetheless. What also makes benchmarking proprietary engines complicated is that most proprietary tools are not simple BPMN engines but full-fledged BPM suites. This heavily affects both the installation and startup procedures which are complex and take a long time. We already provide an approach to use virtual machines with snapshots to easily restore a started process engine within a virtual machine [16]. Currently, this is quite cumbersome to use. Therefore, we are aiming to replace this with Docker and its light-weight containers as they are working to include a similar snapshot functionality as well.

Each engine, being it open source or proprietary, has to fulfill certain criteria so that it can be tested by *betsy*. For BPEL 2.0, we already created an API to handle engines uniformly in [13], making it easier to add new engines or new versions of existing engines. In the future, we plan to extend this API to include BPMN 2.0 engines as well. This is especially important for the proprietary engines as they do have more complex APIs, resulting in a higher entry barrier to actually benchmark them.

Dimension process engine environment: For reproducible research and reproducible benchmarks alike, it is paramount that results are correct and their computation is repeatable [3]. Currently, we use a fresh engine installation for every test, ensuring test isolation and an absence of side-effects. Furthermore, *betsy* is fully automated and therefore provides repeatable results. Again, the usage of container technology is promising to achieve an even higher degree of isolation fixing the benchmark environment, which makes it easier to repeat the benchmark.

What is more, we showed in [16], that virtualization helps to circumvent the install and startup times of the engines, reducing the time to compute the benchmark results drastically, thus, leading to a significantly lower turnaround time [1]. This helps to integrate our benchmark into contemporary continuous integration infrastructures, which can be used by the engine vendors to improve the quality of their implementations. To reduce the execution time even further, we suggest cutting down unnecessary waiting time by calibrating timeouts required during testing to better match the actual system performance. Also parallel and distributed test execution forms a promising area of future work.

5 Conclusion

In this paper, we have presented a roadmap for process engine benchmarking using the *betsy* system. We delineated important dimensions for engine benchmarking and outlined what has been achieved so far in these dimensions with *betsy*. This identifies gaps in current work and outlines potential areas for future work in the area of process engine benchmarking, including a) to put more focus on testing

proprietary engines, b) porting benchmarks for BPEL to BPMN engines, and c) speeding up process engine benchmarks through parallelization and virtualization technologies. By filling these gaps in the future, we hope to support process engine users in a meaningful decision when selecting an engine. To help users with such decisions, we are planning to publish all benchmark results as an interactive website. Furthermore, our work could help process engine vendors to enhance the quality of their products, e.g., by integrating the conformance test features of Betsy into their continuous integration processes. This should reduce the occurrence of test regressions we were able to reveal in our results. Because of this, we aim to get engine vendors on board, fostering and validating our results.

References

1. R. D. Banker, S. M. Datar, and C. F. Kemerer. Factors affecting software maintenance productivity: An exploratory study. In *Proc. of the 8th Intl. Conf. on Information Systems*, pages 160–175. University of Pittsburgh Pittsburgh, PA, 1987.
2. D. Bianculli, W. Binder, and M. L. Drago. Automated Performance Assessment for Service-oriented Middleware: A Case Study on BPEL Engines. In *Proc. of the 19th Intl. Conf. on World Wide Web*, pages 141–150, New York, USA, 2010. ACM.
3. C. Boettiger. An introduction to Docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1):71–79, 2015.
4. M. Chinosi and A. Trombetta. BPMN: An introduction of the standard. *Computer Standards & Interfaces*, 34(1):124–134, January 2012.
5. A. Delgado, D. Calegari, P. Milanese, R. Falcon, and E. García. A Systematic Approach for Evaluating BPM Systems: Case Studies on Open Source and Proprietary Tools. In *Open Source Systems: Adoption and Impact*, pages 81–90. Springer, 2015.
6. G. Denaro, A. Polini, and W. Emmerich. Early Performance Testing of Distributed Software Applications. In *Proc. of the 4th Intl. Workshop on Software and Performance*, pages 94–103, New York, NY, USA, 2004. ACM.
7. V. Ferme, A. Ivanchikj, and C. Pautasso. A Framework for Benchmarking BPMN 2.0 Workflow Management Systems. In *13th Intl. Conf. on Business Process Management (BPM 2015)*, Innsbruck, Austria, August 2015. Springer.
8. M. Geiger, S. Harrer, J. Lenhard, M. Casar, A. Vorndran, and G. Wirtz. BPMN Conformance in Open Source Engines. In *IEEE Intl. Symp. on Service-Oriented System Engineering*, San Francisco Bay, CA, USA, March 30 – April 3 2015. IEEE.
9. M. Geiger and G. Wirtz. BPMN 2.0 Serialization - Standard Compliance Issues and Evaluation of Modeling Tools. In *5th Intl. Workshop on Enterprise Modelling and Information Systems Architectures*, St. Gallen, Switzerland, September 2013.
10. J. Gray. *Benchmark Handbook: For Database and Transaction Processing Systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.
11. S. Harrer, J. Lenhard, and G. Wirtz. BPEL Conformance in Open Source Engines. In *IEEE SOCA, Taipei, Taiwan*, pages 237–244. IEEE, 17–19 December 2012.
12. S. Harrer, J. Lenhard, and G. Wirtz. Open Source versus Proprietary Software in Service-Oriented: The Case of BPEL Engines. In *Proc. of the 11th Intl. Conf. on Service-Oriented Computing (ICSOC'13)*, pages 99–113, Berlin, Germany, 2013.
13. S. Harrer, J. Lenhard, G. Wirtz, and T. van Lessen. Towards Uniform BPEL Engine Management in the Cloud. In *Proceedings des CloudCycle14 Workshops auf der 44. Jahrestagung der Gesellschaft für Informatik e.V. GI*, September 2014.

14. S. Harrer, F. Nizamic, G. Wirtz, and A. Lazovik. Towards a Robustness Evaluation Framework for BPEL Engines. In *IEEE Intl. Conf. on Service-Oriented Computing and Applications*, pages 199–206, Matsue, Japan, 17–19 November 2014. IEEE.
15. S. Harrer, C. Preißinger, and G. Wirtz. BPEL Conformance in Open Source Engines: The Case of Static Analysis. In *IEEE Intl. Conf. on Service-Oriented Computing and Applications*, pages 33–40, Matsue, Japan, 17–19 November 2014. IEEE.
16. S. Harrer, C. Röck, and G. Wirtz. Automated and Isolated Tests for Complex Middleware Products: The Case of BPEL Engines. In *Proc. of the 7th IEEE Intl. Conf. on Software Testing, Verification and Validation Workshops (ICSTW'14)*, pages 390–398, Cleveland, Ohio, USA, April 2014. Testing Tools Track.
17. K. Huppler. *The Art of Building a Good Benchmark*. In *Performance Evaluation and Benchmarking*. Springer Berlin Heidelberg, 2009.
18. ISO/IEC. *ISO/IEC 19510:2013 – Information technology - Object Management Group Business Process Model and Notation*, November 2013. v2.0.2.
19. J. Lenhard, S. Harrer, and G. Wirtz. Measuring the Installability of Service Orchestration Using the SQuaRE Method. In *IEEE Intl. Conf. on Service-Oriented Computing and Applications (SOCA)*, Kauai, HI, USA, December 16-18 2013. IEEE.
20. H. Mili, G. Tremblay, G. B. Jaoude, E. Lefebvre, L. Elabed, and G. E. Boussaidi. Business Process Modeling Languages: Sorting Through the Alphabet Soup. *ACM Comput. Surv.*, 43(1):4:1–4:56, December 2010.
21. OASIS. *Web Services Business Process Execution Language*, April 2007. v2.0.
22. C. Pautasso, V. Ferme, D. Roller, F. Leymann, and M. Skouradaki. Towards Workflow Benchmarking: Open Research Challenges. In *Conf. on Database Systems for Business, Technology, and Web*, pages 1–20, Hamburg, Germany, March 2015.
23. C. Röck, S. Harrer, and G. Wirtz. Performance Benchmarking of BPEL Engines: A Comparison Framework, Status Quo Evaluation and Challenges. In *Proc. of the 26th Intl. Conf. on Software Engineering and Knowledge Engineering (SEKE'14)*, pages 31–34, Vancouver, Canada, July 2014.
24. T. L. Saaty. How to make a decision: The Analytic Hierarchy Process. *European Journal of Operational Research*, 48(1):9–26, 1990.
25. S. E. Sim, S. Easterbrook, and R. C. Holt. Using Benchmarking to Advance Research: A Challenge to Software Engineering. In *25th International Conference on Software Engineering*, Portland, Oregon, USA, May 2003.
26. S. E. Sim, S. Easterbrook, and R. C. Holt. Using benchmarking to advance research: A challenge to software engineering. In *Proceedings of the 25th International Conference on Software Engineering*, pages 74–83. IEEE Computer Society, 2003.
27. M. Skouradaki, D. H. Roller, F. Leymann, V. Ferme, and C. Pautasso. On the Road to Benchmarking BPMN 2.0 Workflow Engines. In *Proc. of the 6th ACM/SPEC Intl. Conf. on Performance Engineering*, pages 301–304. ACM, 2015.
28. E. Triantaphyllou. *Multi-criteria decision making methods: a comparative study*, volume 44. Springer Science & Business Media, 2013.
29. W. van der Aalst and A. ter Hofstede. YAWL: yet another workflow language. *Information Systems*, 30(4):245–275, June 2005.
30. W. M. P. van der Aalst. Business Process Management: A Comprehensive Survey. *ISRN Software Engineering*, pages 1–37, 2013.
31. E. J. Weyuker and F. I. Vokolos. Experience with Performance Testing of Software Systems: Issues, an Approach, and Case Study. *IEEE Trans. Softw. Eng.*, 26(12):1147–1156, December 2000.
32. WfMC. *XML Process Definition Language*, August 2012. v2.2.