Process Engine Benchmarking with Betsy in the Context of ISO/IEC Quality Standards

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 BPEL 2.0 or, more recently, BPMN 2.0. Implementations of these standards which are able to execute models, so called process engines, differ in their quality of service, 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 use case 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 the current achievements in process engine benchmarking and point out future directions.

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 [23]. 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 [19] 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. One technique to support a meaningful decision for an end user is *benchmarking* [22], which in this case resolves to *process engine benchmarking*. Enabling 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 engines[1]. 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.

This paper is a revised version of [6] and extends it in several directions: a) a mapping of the process engine capabilities *betsy* can benchmark to software quality characteristics defined by the product quality model of ISO/IEC 25010:2011 [15] in Sect. 3.2, b) an outline of usage scenarios of *betsy* for various stakeholders according to ISO/IEC 25051:2014 [17] in Sect. 3.3, and, c) a discussion of industry initiatives for process engine benchmarking in Sect. 2.

The remainder of the paper is structured as follows: We briefly discuss requirements for benchmarking and 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. The paper is summed up in Sect. 5.

2 Related Work

Benchmarking of IT products is not a new phenomenon and various studies on this topic can be found, e.g., [2, 14, 22]. Requirements for "good", i.e., valid, benchmarks can be found in [14]: Overall, a benchmark should measure *relevant aspects* to be able to provide substantial answers to the investigated research questions. Benchmarking process engines is a relevant topic as there are no certification authorities to check claimed compliance promises. As a result, each vendor can claim that his product conforms to a standard without the

[1] The tool itself is available at `https://github.com/uniba-dsg/betsy`.

need to actually prove it. Moreover, further criteria are relevant for users of BPM products, such as ease of installation, portability, or conformance to static analysis rules. However, [14] lists other requirements for benchmarks which might conflict with the aspect of *relevance*. A benchmark should also be *repeatable*, *fair*, *verifiable* and *economical*. For *betsy*, those four requirements are fulfilled: *betsy* is Open Source and fully automated, which allows for *repeated* test execution. Moreover, most of the tested engines are freely available and directly integrated, which enables every interested party to execute the tests without *economic* barriers. As the benchmark in betsy is inferred from standard documents, which define relevant aspects, and not from concrete software tooling, it does not give an advantage to some engines but is *fair*. Due to the openness of the standard texts and our freely available implementation the correctness of betsy is open to scrutiny, fostering *verifiability*.

Apart from these general aspects, there are two recent academic approaches regarding process engine benchmarking, which are more closely related to our work. One is the benchflow project [5, 21]. Their latest work [5] 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 [2]. However, directly reusing the concepts and artifacts generated by this approach is not useful for the scope of our tests, as measuring performance requires far more complex infrastructure apart from the actual engines under test to generate sensible workloads and to ensure the validity of the results [14]. The second approach [3] presents a method to evaluate BPM systems with the aim of selecting the best fitting one for a list of requirements. In a series of case studies, the authors evaluate a large list of open source and proprietary systems implementing three different process languages. In contrast to our work, this evaluation is on a more abstract level and the engine evaluation is not automated.

Benchmarking approaches for process engines also exist in industry. For instance, the BPMN engine vendor camunda provides a performance testing framework for their own process engine[2]. This framework is based on the benchmarking project of the activiti engine[3]. In contrast to our work, their focus is to assess the relative effects of code changes on the performance characteristics of the engine. Similar to betsy, their benchmark is fully automated.

3 The Current State of betsy

The tool *betsy* 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. In this section, we outline the capabilities of betsy according to four dimensions in Sect. 3.1, classify the engine capabilities *betsy* is able to benchmark in Sect. 3.2, and conclude with an outline of use cases of *betsy* in Sect. 3.3.

3.1 The Four Dimensions of betsy

Table 1: The Four Dimensions of *betsy*

Dimensions	Characteristic	Publications
Process Languages	BPEL 2.0	[9–13, 18]
	BPMN 2.0	[7, 8]
Process Engine Capabilities	Feature Conformance	[7–10]
	Static Analysis Conf.	[12]
	Expressiveness	[8, 10]
	Robustness	[11]
	Installability	[18]
Process Engine Types	Open Source Engines	[7–13, 18]
	Proprietary Engines	[10]
Process Engine Environments	Bare Metal Environment	[7–12, 18]
	Virtual Environment	[13]

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. An overview is given in Table 1. 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 [20] or BPMN [16]. The dimension *process engine capabilities* describes 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 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 process engines in a bare metal environment or a virtual environment, such as in a virtual machine or a container.

3.2 Classification of Engine Capabilities

The purpose of benchmarking process engine capabilities is to gather information about the quality of an engine. The software quality characteristics *betsy* targets can be identified using the widely accepted product quality model defined in the ISO/IEC 25010:2011 standard [15]. Table 2 outlines the mapping between the engine capabilities *betsy* can benchmark and the corresponding quality characteristics. For four out of the eight quality characteristics of [15], at least one of the sub-characteristics can be evaluated by *betsy* directly. This shows the versatile application of *betsy* as an evaluation platform. The focus of the current version of *betsy* lies on *functional suitability* and *portability*. By

[2] https://github.com/camunda/camunda-bpm-platform/tree/master/qa/performance-tests-engine

[3] See http://www.jorambarrez.be/blog/2012/06/28/the-activiti-performance-showdown/

benchmarking *feature conformance*, *betsy* provides data for *functional completeness*, *functional correctness*, and *replaceability*, making it the most influential benchmarked engine capability, followed by *expressiveness* which helps to determine *functional suitability* as well. Both *installability* and *static analysis conformance* have their direct counterpart in the quality model [15]. *Fault tolerance* is covered by *betsy*'s capability to evaluate *robustness*. What is more, the quality characteristic *maintenance* with its sub-characteristic *testability* is indirectly addressed, as *betsy* performs automated tests which inherently require a certain form of *testability*.

Table 2: Classification of Engine Capabilities

Quality Characteristic	Engine Capabilities
functional suitability	
functional completeness	feature conformance, expressiveness
functional correctness	feature conformance, expressiveness
usability	
user error protection	static analysis conformance
resilience	
fault tolerance	robustness
portability	
installability	installability
replaceability	feature conformance

3.3 Use Cases of betsy

Process engines are software products that can be downloaded, installed, and used directly. Thus, they correspond to *ready to use software products* (RUSP) in the sense of the ISO/IEC 25051:2014 standard [17]. This standard defines requirements for RUSP and instructions on how to evaluate such requirements. In this context, [17] specifies different stakeholders, i.e., vendors, (potential) users, certification bodies, testing laboratories, and accreditation bodies, for whom a tool like *betsy* is useful for a variety of different purposes [17, p. 1]. Certification bodies can define certification schemes, i.e., benchmarks, which are used by testing laboratories to determine whether an engine fulfills a specific certification. Both, certification bodies and testing laboratories, can be accredited through accreditation bodies. Vendors can use *betsy* to assess and improve the quality of their engines and product information, declare conformance towards an existing standard, or apply for certificates. Furthermore, potential users can check whether their requirements are met by the engine and/or its product information, and whether it is certified as required. Thus, *betsy* is a valuable tool for the evaluation of process engines as RUSP.

4 Future Directions

To support the use cases of *betsy*, we aim to extend it in various directions. Our plans are detailed along the four dimensions outlined in Sect. 3.1.

Dimension process language: The field of process standards is vast [19] and in constant evolution. The relevance of a process engine benchmarking system depends on the relevance of the language it supports. Currently, *betsy* supports BPEL 2.0 [20] and BPMN 2.0 [16]. Arguably, these two languages are sufficient at the moment, since there is no competing standard that equally targets the execution of process models on process engines.

Dimension process engine capability: For BPEL 2.0 engines, *betsy* already covers a large variety of engine capabilities [9–13, 18]. With the emergence of BPMN 2.0, we have started to benchmark the feature conformance and expressiveness of BPMN 2.0 engines as well [7, 8]. Our current goal is to fill in open gaps by benchmarking BPMN 2.0 engines for the same set of capabilities as for BPEL 2.0 engines.

In addition, it would be desirable to increase the set of covered engine capabilities along the different quality characteristics of the ISO/IEC 25010:2011 standard [15]. Especially *performance efficiency* is of interest since the performance of process execution heavily relies on the performance of the engine used [4]. Performance has always been an important criterion for software selection and evaluation [24]. The conformance results of betsy can be used to determine sensible workload models that lead to fair and reproducible benchmarks. Existing test suites, e.g., of workflow control-flow patterns, could be used as workload models for performance micro-benchmarks.

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 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. [7–9, 11–13, 18]. In contrast, work that explicitly compares these two types of process engines is rare, e.g. [10]. 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 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 [10], is a way to relieve restrictions, given academic licenses are available.

Dimension process engine environment: For reproducible research and reproducible benchmarks alike, it is paramount that results are correct and their computation is repeatable [2]. 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 reproducible results. To achieve an even higher degree of isolation, we aim to use Docker containers and images to capture and fix the benchmark environment, which makes it easier to repeat the benchmark. If vendors use *betsy* to improve the quality of their engines, it is paramount that the benchmark executes quickly to ensure fast feedback cycles [1]. To circumvent the long engine install and startup procedures, we restored snapshots of virtual machines in [13], reducing benchmark duration drastically. To reduce this 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 dimensions for engine benchmarking and outlined what has been achieved so far in these dimensions with *betsy*. Moreover, we mapped the capabilities evaluated by *betsy* to notable software quality characteristics and outlined use cases of the tool for various stakeholders. We discussed gaps in current work and identified potentials for future work in the area of process engine benchmarking. 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, and even the quality of the product information and documentation.

References

[1] R. D. Banker, S. M. Datar, and C. F. Kemerer. Factors affecting software maintenance productivity: An exploratory study. In *ICIS*, 1987.

[2] C. Boettiger. An introduction to Docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1):71–79, 2015.

[3] 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.

[4] G. Denaro, A. Polini, and W. Emmerich. Early Performance Testing of Distributed Software Applications. In *WOSP*, pages 94–103. ACM, 2004.

[5] V. Ferme, A. Ivanchikj, and C. Pautasso. A Framework for Benchmarking BPMN 2.0 Workflow Management Systems. In *BPM*. Springer, 2015.

[6] M. Geiger, S. Harrer, and J. Lenhard. Process Engine Benchmarking with Betsy – Current Status and Future Directions. In *ZEUS*, 2016.

[7] M. Geiger, S. Harrer, J. Lenhard, M. Casar, A. Vorndran, and G. Wirtz. BPMN Conformance in Open Source Engines. In *IEEE SOSE*, 2015.

[8] M. Geiger, S. Harrer, J. Lenhard, and G. Wirtz. On the Evolution of BPMN 2.0 Support and Implementation. In *IEEE SOSE*, 2016. to appear.

[9] S. Harrer, J. Lenhard, and G. Wirtz. BPEL Conformance in Open Source Engines. In *IEEE SOCA, Taipei, Taiwan*, pages 237–244, 17–19 Dec. 2012.

[10] S. Harrer, J. Lenhard, and G. Wirtz. Open Source versus Proprietary Software in Service-Orientation: The Case of BPEL Engines. In *ICSOC*, 2013.

[11] S. Harrer, F. Nizamic, G. Wirtz, and A. Lazovik. Towards a Robustness Evaluation Framework for BPEL Engines. In *IEEE SOCA*, 2014.

[12] S. Harrer, C. Preißinger, and G. Wirtz. BPEL Conformance in Open Source Engines: The Case of Static Analysis. In *IEEE SOCA*, 2014.

[13] S. Harrer, C. Röck, and G. Wirtz. Automated and Isolated Tests for Complex Middleware Products: The Case of BPEL Engines. In *IEEE ICSTW*, pages 390–398, 2014.

[14] K. Huppler. The Art of Building a Good Benchmark. In *Performance Evaluation and Benchmarking*. Springer Berlin Heidelberg, 2009.

[15] ISO/IEC. *ISO/IEC 25010:2011; Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*, 2011.

[16] ISO/IEC. *ISO/IEC 19510:2013 – Information technology - Object Management Group Business Process Model and Notation*, 2013. v2.0.2.

[17] ISO/IEC. *ISO/IEC 25051:2014; Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing*, 2014.

[18] J. Lenhard, S. Harrer, and G. Wirtz. Measuring the Installability of Service Orchestrations Using the SQuaRE Method. In *IEEE SOCA*, 2013.

[19] H. Mili, G. Tremblay, G. B. Jaoude, E. Lefebvre, et al. Business Process Modeling Languages: Sorting Through the Alphabet Soup. *ACM Comput. Surv.*, 43(1):4:1–4:56, December 2010.

[20] OASIS. *Web Services Business Process Execution Language*, April 2007. v2.0.

[21] C. Pautasso, V. Ferme, D. Roller, F. Leymann, and M. Skouradaki. Towards Workflow Benchmarking: Open Research Challenges. In *BTW*, 2015.

[22] S. E. Sim, S. Easterbrook, and R. C. Holt. Using Benchmarking to Advance Research: A Challange to Software Engineering. In *ICSE*, 2003.

[23] W. M. P. van der Aalst. Business Process Management: A Comprehensive Survey. *ISRN Software Engineering*, pages 1–37, 2013.

[24] 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.