Case Study and Practical Assessment of BPMN with Camunda

Vincent Bode, Marcel von Maltitz*, Holger Kinkelin*
*Chair of Network Architectures and Services, Department of Informatics
Technical University of Munich, Germany
Email: vincent.bode@tum.de, vonmaltitz@net.in.tum.de, kinkelin@net.in.tum.de

Abstract—**Processes are a sequence of activities designed to reach a targeted outcome. In this general form, processes are an essential component of many types of systems. Process modeling is a technique that formalizes processes by documenting them using abstract notation. It can be used to improve a running system by optimizing the modeled process. Using a case study process of scheduling an appointment with a professor, we elicit some requirements and discuss the benefits and drawbacks of using Business Process Model and Notation (BPMN), a widespread standardized notation for modeling processes [5], for implementing this process. We observe that the formal BPMN process is a viable alternative to less formalized solutions, that it has potential for automation and a reduction of errors, but it can incur higher maintenance effort than an informal version. The applicability of BPMN therefore varies across use cases.**

Index Terms—**process modeling, bpmn, camunda**

1. Introduction

Processes are “a series of actions or operations conducing to an end” [1] found in many systems. Whether they are defined explicitly or implicitly, achieving a target goal is always driven by a process. They can be categorized according to specific traits such as their run-time, execution frequency or whether the process is mainly human or machine based. Ensuring that frequent and time-intensive processes are implemented efficiently and effectively can have a positive impact on the system’s performance. This is the field of process modeling, where the interactions between participants, tasks and communication events are analyzed and optimized with the goal of improving certain aspects of the system, such as processing speed, fault-tolerance or reliability.

The rest of the paper is divided into the following sections. Section 2 motivates process modeling. The case study is introduced in Section 3 and some requirements are stated. Section 4 introduces BPMN and Camunda used for reimplementing the process in a formalized manner. A review of the benefits and challenges encountered during the implementation is presented in Section 5. Finally, the solution is evaluated in Section 6.

2. Reasons for Process Modeling

Implicit processes are loosely defined and can consist of nothing more than a state and a target outcome, while explicit processes are fixed in documentation or formalized interaction. Making processes explicit has several advantages. For one, it allows an analyst to gain insight to complex workflows, allowing them to analyze “the sequence of steps involved in moving from the beginning to the end of a working process” [1] and making the process observable. This allows the analyst to perform studies that measure the workflow’s performance and compare it to different variations, allowing them to improve the process incrementally. A second advantage is that the process model is also a way of documenting how a process works, allowing for standardization, repeatability, simpler knowledge transfer and increased transparency. One disadvantage is that a process model is also an artifact that needs to be kept up to date. To learn about the impact of formalizing processes, we wish to examine a specific problem involving human and machine participants.

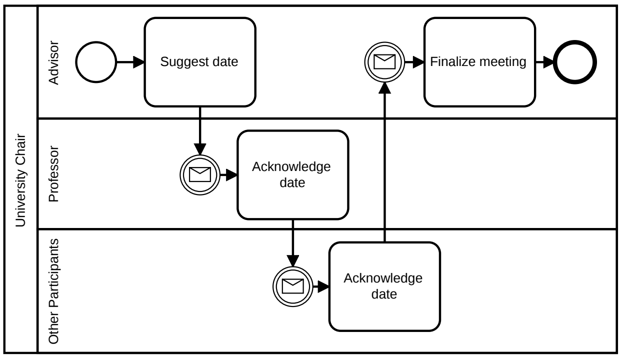

Figure 1. A schematic of the current scheduling of a meeting process. It shows the best-case scenario, where each participant is available at the suggested date and all communication is error free.

3. Case study

Throughout this work, a case study will be used to evaluate the practical effects of using process modeling on a process to improve its performance. We will examine the scheduling of an appointment with a professor and multiple other participants at a university chair. A schematic of this process may be seen in Figure 1. Table 1 shows multiple issues that affect the performance of the current process. To support our modeling and implementation, a workflow management system will be used. Such a system can administer formalized processes and track the state and history of any process running inside of it. The case study focuses on how the current problems can be solved by introducing such a system and which new ones may arise.

doi: 10.2313/NET-2019-06-1_01

Issue	Consequence
Inspectability	At any given point in the process, it is hard for a participating party to judge the state of the process and whether available information is up to date due to the lack of a centralized source of information.
Multiple iterations	When schedules are packed, the chance of a suggested date being available decreases. Many requests may be necessary before a valid appointment is found, increasing the chance of human error for each additional request.
Mailbox issues	The flexibility offered by email also allows for many possibilities of small, hard to trace mistakes with larger consequence. For example, a recipient might be forgotten by accident, an error that is hard to discover but can have large impact.

TABLE 1. ISSUES WITH THE CURRENT SCHEDULING PROCESS

3.1. Requirements

The new process should fulfill the following requirements to improve on problems in the current implementation.

1) **Reduced consumption of resources** is one possible goal when exchanging processes for improved versions. One resource to minimize is the participant's time, which can be measured by the average process execution time. The solution should therefore not increase this average.
2) **Runtime flexibility** is key. The improved process should prevent error prone or otherwise unwanted methods of reaching the goal while retaining desirable approaches in order to efficiently reach the target. A process that is too restrictive can face user acceptance issues, while a process that allows for too many options may be hard to maintain.
3) **Transparency and accountability** is necessary to ensure that any given task has a responsible participant. The current mail-based process is tracked by the participant's mailbox, the new process should also offer a way to view task responsibilities.
4) **Eliminating repetitive or tedious tasks** achieves increased user satisfaction and reliability. These are tasks where automation is usually viable, leading to further reduced error rates and operator strain. The feasibility of automation using BPMN machine tasks should be considered for each activity.
5) **Maintainability** is an important factor, as processes are not static and will need to be adapted over time. A solution should account for this and implement the necessary paradigms to ensure it can be maintained.

Fulfilling these requirements will improve the workflow's performance, increasing efficiency and effectivity.

4. BPMN and Camunda

The Business Process Model and Notation (BPMN) is a standardized notation for process modeling at a high level. BPMN was built to model the interaction between human and machine tasks in diagrams. We will use a BPMN toolset to implement the new solution [5]. A typical setup consists of an analyst, a modeling tool, a process engine as well as external actors and systems which interact with the process (Figure 2). One advantage of BPMN being a standardized, XML-based notation is that the format is vendor-independent. This allows components such as the modeling tool and the process engine to be chosen individually. One of these providers for a modeling tool as well as a process engine is Camunda. It was chosen due to its open source implementation as well as its suitability for fulfilling the requirements.

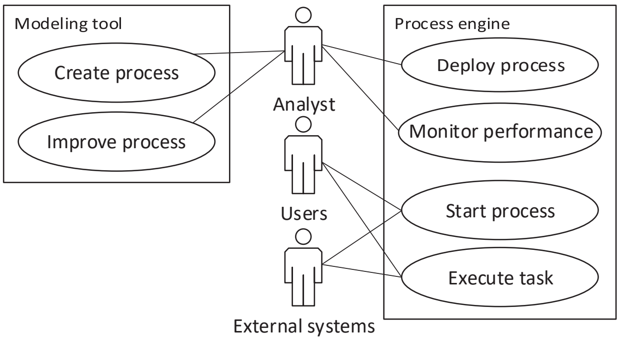

Figure 2. UML use case diagram of a generic BPMN system. An analyst maintains the process in the modeling tool and monitors it in the process engine while users and external systems interact with it.

Three parts make up the process engine. The "Cockpit" is used for monitoring by the analysts, the task list for completing human tasks, and the administration for configuring the system. The Camunda process engine provides both user and group task lists containing activities, each of which can be assigned automatically or manually. Each activity can contain a form that allows humans to interact with data in the process. For machine tasks, the process engine can execute Java code, call REST APIs, execute scripts or simpler Java Unified Expression Language (JUEL) expressions.

The Camunda modeler is another useful part of the ecosystem, as it serves as an abstraction layer between the analyst and the XML Notation, which can be verbose. Editing a visual representation of the process will come more naturally to the analyst, especially since they might not have a technical background. The modeler also supports some Camunda specific extensions, such as forms and scripts, to simplify configuration.

5. Implementation

During the implementation, we were able to make use of some benefits provided by the BPMN and Camunda platform. However, there were also some pitfalls leading to implementation challenges.

5.1. Advantages

BPMN and Camunda are a powerful toolset for process modeling. During the implementation, the following features were found to be particularly useful.

5.1.1. Forms. A useful extension by Camunda to the BPMN specification is the ability to add forms. It allows the analyst to append HTML forms to tasks, providing a simple way to interact with the user. Variables used inside the BPMN process can be directly made available to the user for viewing or modification. There is no coding required to set up these forms since all the configuration works inside the modeler and there are no external systems required for providing user interaction. If the analyst decides they need something more sophisticated, it is possible to embed custom HTML forms to further improve end user experience. All the user interaction in the scheduling process, such as asking the user to specify and approve dates, was implemented using these forms.

5.1.2. Script Execution. Another useful feature offered by Camunda is that it can run scripts embedded in the BPMN diagram directly on the process engine. This allows uncomplicated interfacing with other systems and the developer is free to script parts of the system with the tools of their choice. This is a plus for automation as well, since small repetitive tasks that would otherwise be executed manually can now be scripted with a little time investment. The scheduling process used this feature for implementing logic that would have been hard to read in a BPMN diagram, such as detecting when there is no more viable date at which all parties can attend. It was also used for communicating with the email service to send notifications to the individual users.

5.2. Challenges

While the in-built functionalities included in Camunda make it suitable for many use cases, using BPMN and Camunda to implement a simple process is less straightforward than one might expect. This is illustrated by some problems encountered below.

5.2.1. Deployment. After having modeled a process in the modeler, it is time to deploy it to the process engine. This can be achieved through the modeler's user interface, but not without quirks. One problem a user might encounter is a misleading status message when trying to deploy BPMN diagrams, stating that the deployment was successful. Searching the Camunda Cockpit, it is possible that the just deployed process is nowhere to be found. This can be due to syntax or semantic errors in the BPMN diagram. Unfortunately, the user does not receive any feedback as to what went wrong and the documentation does not offer help on how to troubleshoot this problem. Instead, the only method of finding errors related to the BPMN diagram we were able to find is to search the logs of the Apache Tomcat instance in which Camunda runs. The encountered error will be listed inside a Java exception and refers to the position of the error in the BPMN XML notation. This in turn can help the user track down the error in the visual representation of the diagram to fix it in the local diagram and attempt redeployment.

While users can certainly get used to this method of error tracing, it violates some usability guidelines such as Shneidermann's "8 Golden Rules of User Interface Design" [6]. Rule two states that feedback should be instant and precise. Rule five states that errors should be prevented where possible and that help should be offered where not. Both of these rules are not well implemented, decreasing overall usability. Another problem is that the only group of users who are likely to check any log files are programmers. Since BPMN is designed to be used by both developers and business users as a way of collaborating, this solution ignores the needs of the less technically skilled users. Additionally, it is likely that users are working in a business environment where they do not have access to the logs for security reasons, which could prevent them from getting any assistance at all.

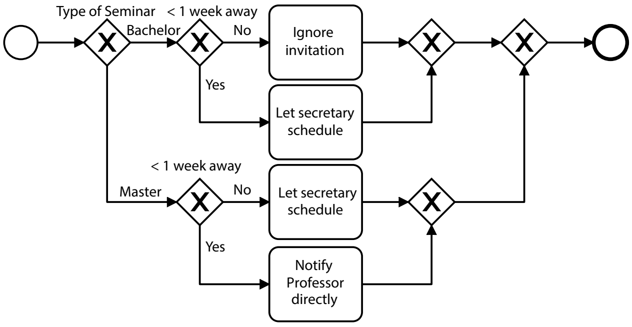

Figure 3. The BPMN equivalent of spaghetti code. Each new decision makes the number of activities double, causing the diagram to quickly increase in size. Complex decision-making should be avoided in BPMN in favor of comprehensibility, since complexity cannot be hidden by the usual object-oriented concepts such as inheritance. Instead, it is possible to externalize decisions to tools such as DMN, a notation specifically designed for decision-making [4].

5.2.2. Utilizing BPMN to the appropriate Degree. Another issue that users will encounter while using BPMN is ensuring that the level of detail which is used for process diagrams is appropriate. Too little detail will mean that changing a process to fit new requirements is inflexible, as large parts of the process are black-box activities. Too much detail, and it will become hard to interpret and modify process diagrams accordingly (see Figure 3). In this case, it would be preferable to encapsulate the decision-making in an activity and implement it somewhere else in order to hide complexity. Experience is required to be able to judge whether a given set of activities should be modeled within BPMN or whether an implementation in other tools is more appropriate.

5.2.3. Testing and Debugging. Since BPMN has similarities with programming languages, it is worth taking a look at how processes are tested and debugged within Camunda. One nice feature offered is allowing the user to view variables inside of a running process. This feature represents the basis for debugging a process, as it makes its internal state observable. Unfortunately, this is where the built-in tools for testing and debugging end. Therefore, the testing framework lacks some functionality one might expect of a programming language, such as automated testing or setting breakpoints. One particular tedious problem is when processes require a large amount of user input. Getting the process into a certain state for testing can require lots of manual interaction and this setup needs to be repeated every time a test is run. Some help is offered by the Camunda REST API, which allows the automation of human input tasks by calling appropriate API endpoints instead of entering form data manually.

6. Evaluation of Solution

In order to review the solution that was implemented, the process will now be evaluated against the previous set of requirements. A simplified view of the resulting BPMN workflow can be seen in Figure 4.

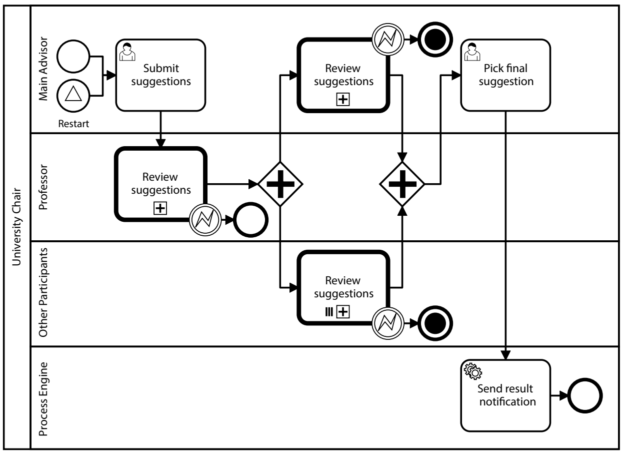

Figure 4. An extract of the final seminar scheduling BPMN process. It includes error handling and sub-processes (thick border activity) to build reusable parts of the process. Notice how the overall structure of the workflow has nevertheless stayed the same.

6.1. Review of Requirements

1) **Reduced consumption of resources** is one of the requirements specified earlier. Whether the newly implemented process is faster than the existing one depends on the probability that all participants are available for the requested appointment. A high probability causes the mail-based solution to be faster, however each retry increases the chance for error and slows down the mail process. This is where relative improvement can be observed with the BPMN-based process. Since the BPMN process is closely based upon the existing process, the number of tasks a user receives stays the same. This requirement is therefore partially fulfilled.
2) **Runtime flexibility** was maintained where appropriate. While there is a strict sequence of activities, it is possible for the advisor to cancel or restart the process at any time should the conditions change.
3) **Transparency and accountability** have been improved from the existing solution. At any given state, it is possible for participants to view the internal state of the process as well as who is responsible for the current activity. A history of previous activities is also provided. This requirement is therefore fully fulfilled.
4) **Eliminating repetitive or tedious tasks** was another previously stated goal. This was achieved by automating tasks suitable for machine processing, such as sending result notifications to participants and publishing the agreed date on the chair's website. This lessens the potential for error and increases user satisfaction.
5) **Maintainability** was a technical requirement. Some common techniques such as call activities (the BPMN equivalent of procedures and procedure calls) were used to reduce duplication.

6.2. User study

To further gain qualitative feedback on the new implementation, a small user base was asked to test the process. Some responses are shown in Table 2. It shows some things that were not considered in the design, and some potentially unwanted mechanics that were missed in testing. These provide a good basis for the analyst to continue improving the process.

Topic	Response
Communication	"The scheduler should provide a means of communicating with the others. Otherwise, I need to use email for communication in addition to this scheduler."
Disappearing tasks	"Sometimes tasks disappear from the task list without an apparent reason." (Later, it was found this happens when other people submit their review in parallel and there are no more viable suggestions)
Information flow	"It does a good job tracking what needs to be done with multiple running requests."

TABLE 2. USER'S FEEDBACK AFTER TESTING THE BPMN PROCESS.

The newly introduced process is therefore a viable alternative to the e-mail based solution. While there are other known solutions to the scheduling problem, this case study was useful to examine some benefits and drawbacks of BPMN based process modeling in relation to academic processes.

7. Conclusion

Processes are part of many systems and managing them can help improve their performance. The effects of using process modeling were examined using a case study of a process at the university chair. Some problems with the current process and requirements for an improved version were determined. A new solution was then implemented using BPMN and Camunda, which was later evaluated against the previous requirements. We showed that both the email-based solution and the BPMN-based solution each have their advantages and disadvantages. While the email-based solution is less maintenance intensive, its complexity and error rate increase with a decreased probability of all participants being available at a given request. On the other hand, while the BPMN implementation offered good accountability, it reduced the overall flexibility. With a small user study, some additional feedback was collected and some minor issues found. Overall, the new solution was deemed a viable alternative. When applied to the right processes with the right requirements, BPMN and Camunda can help achieve significant improvement in efficiency and effectivity.

References

[1] Merriam-Webster, "Dictionary", https://www.merriam-webster.com/dictionary/. Accessed 25 Sept. 2018.

[2] camunda services GmbH, "BPMN 2.0 Symbol Reference", https://camunda.com/bpmn/reference/. Accessed 02 Sept. 2018.

[3] camunda services GmbH, "Best Practices for creating BPMN 2.0 process diagrams", https://camunda.com/bpmn/examples/. Accessed 25 Sept. 2018.

[4] camunda services GmbH, "DMN Tutorial", https://camunda.com/dmn/. Accessed 25 Sept. 2018.

[5] Object Management Group (OMG), "Business Process Model and Notation (BPMN)", https://www.omg.org/spec/BPMN/2.0.2/PDF, December 2013

[6] Dr. O. N. N. Fernando and Dr. Y. He, "Human Computer Interaction", NTU School of Computer Science and Engineering, January 2018