SHAPEworks: A BPMS Extension for Complex Process Management

Saimir Bala[1], Giray Havur[1], Simon Sperl[2], Simon Steyskal[1 2], Alois Haselböck[2], Jan Mendling[1], and Axel Polleres[1]

[1]Vienna University of Economics and Business, Austria
{name.surname}@wu.ac.at
[2]Siemens AG, Austria {name.surname}@siemens.com

Abstract. Complex engineering projects, such as the deployment of a railway infrastructure or the installation of an interlocking system, involve human safety and make use of heterogeneous data sources, as well as customized engineering tools. These processes are currently carried out in an ad-hoc fashion, relying on the experience of experts who need to plan, control, and monitor the execution of processes for delivering value to the customers. This setting makes an automated overarching-process a crucial step towards supporting engineers and project managers to deal with safety-critical constraints and the plethora of details entailed by the process. This paper demonstrates a tool that combines methods from automatic reasoning, ontologies and process mining, implemented on top of a real Business Process Management System (BPMS).

Keywords: Process Management, Resource Management, Ontologies, Process Mining, BPMS, Compliance

1 Introduction and Significance to BPM

Complex engineering projects involve a wide variety of tools, data-sources and specific domain knowledge. This is the case, e.g., for the installation of a railway interlocking system. Furthermore, these projects must respond to strict constraints imposed by safety rules and regulations in the operating domain. Managing and monitoring these projects in real life presents a number of challenges, especially in companies with a high focus on optimizing their operations through Business Process Management (BPM).

Three perspectives are of crucial importance in our setting. First, resources need to be allocated optimally not only to optimize delivery times of the final product, but also to comply with the safety rules in the domain, e.g., assign resources to tasks according to their expertise level. Second, the engineers' work must be traced in order to allow internal or external auditors to verify it at a later stage. This work can be reflected by generated artifacts, such as emails, word processor documents, and Version Control System (VCS) logs, which result from the engineering tools used by the engineers to accomplish their tasks. Third, the data from different software tools and engineering workflows must be accessible in order to ease reporting and documentation of what work has been done. To respond to these challenges, an overarching engineering process is fundamental, along with a framework that supports its execution [3].

We have devised a framework for addressing the above mentioned perspectives through a single automated solution. We demonstrate how automated reasoning can be used for optimally scheduling resources and tasks. The resulting schedule is flexible and can be customized by imposing soft constraints (e.g., optimization statements) or hard constraints (e.g., compliance rules) on resources. We use ontologies to represent organizational and other process data, e.g.,, rules and regulations. History logs are frequently monitored and mined in order to update existing knowledge of the process with new insights. Furthermore, we support document generation, which helps towards reducing human errors and increasing compliance to documentation requirements.

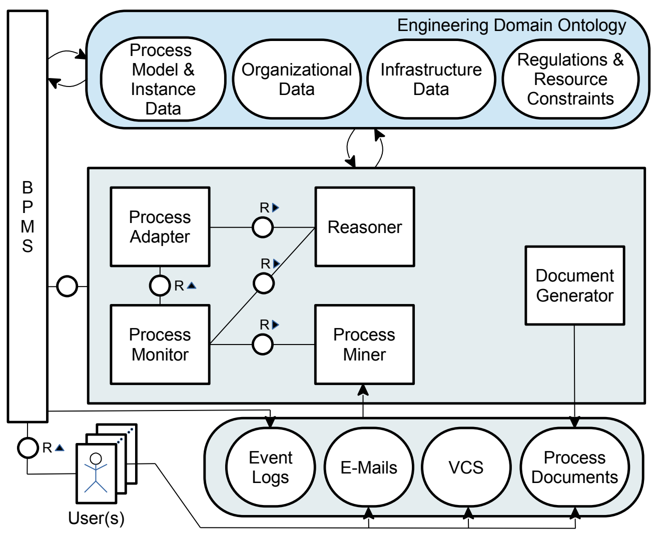


Fig. 1: Framework for process management in complex engineering projects

Fig. 1 illustrates the developed framework, using the Fundamental Modeling Concepts (FMC) notation.[2] The main components are briefly described below.

BMPS. We have extended the Camunda BPM engine, an extensible business process engine which offers API access. The engine executes the defined engineering process, and we interact with it by capturing the events about task completion, process start, process end, etc, and by using historical information that is stored in its logs.

Reasoner. The reasoner module is in charge of both computing resource allocations using Answer Set Programming (ASP) and validating Shapes Constraint Language (SHACL) [5] constraints for potential violations of domain constraints. Details on the resource allocation approach can be found in [4].

[2] http://www.fmc-modeling.org/

Process Monitor. This component listens to task completion or task starting events, in order to actively check for process non-compliant behavior and to signal potential anomalies. The process monitor uses results from the Miner to raise alerts in case the process could not be executed according to the schedule.

Miner. The Miner deals with historical data in order to infer useful information about the process. In our use case, the focus is on gathering useful statistics about resources and process activities.

Document Generator. The Document Generator is able to create or fill in textual documents from data that has been generated by users that complete their tasks. The data includes user comments to tasks and process variables.

Process Adapter This component is in charge of computing adaptations when slight deviations occur. When the adaptation is not possible, it triggers an alert to the Reasoner and the process must be stopped and re-planned.

We use an ontology to represent the engineering domain and the organizational (i.e., resource-related) knowledge, business processes, and regulations and policies. Process relevant data are stored in RDF, which allows also to use SHACL to check compliance constraints. The ontology can be used as a shared repository by the above mentioned components to share data. Our framework also considers data from event logs, emails and VCS, which can be further used for mining traditional processes, text, and project-oriented processes [1].

2 Use Case and Tool Description

We have extended the Camunda[3] BPMS with the components described in Sect. 1. Our solution stems from a real industry scenario. The bigger context of the tool is the setting described in [2]. Here we show how our tool can be used to support a typical engineering process.

2.1 Industry Scenario

A typical process in the railway domain aims at the release of a new engineering system for a railway customer. The process starts when a new agreement with a client has been signed by the project management team. A new repository is created for the customer data and, at the same time, possible additional data are requested from the client. The next step is the actual *System engineering* activity, where the new system is built. In turn, the lab management team sets up the laboratory for executing possible required tests. If test results are not satisfactory, the system must be re-engineered. Otherwise, a report is generated and the work is handed over again to the project management team, which then delivers a new release to the customer. Fig. 2 illustrates such a process, modeled in Business Process Modeling Notation (BPMN).

[3] https://camunda.org/

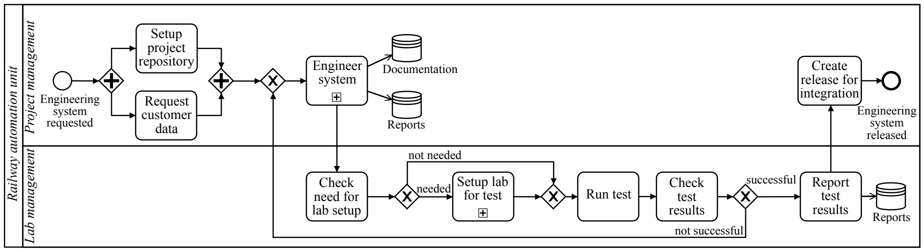
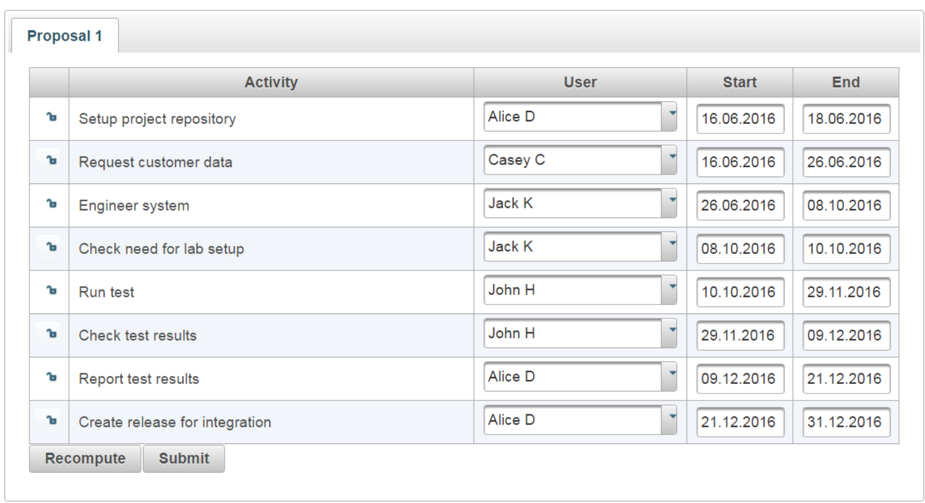


Fig. 2: An engineering process in the railway domain

2.2 Usage of the Tool

In this section we show a use case of our tool in the above mentioned scenario. Initially the user logs in to the Camunda BPMS where they can see the task list. The users with administrator privileges can start a new engineering process. When the process is started and before the first activity is executed, the reasoner computes a schedule. Afterwards, the user interacts with the results of the reasoner by confirming or modifying the schedule. A Graphical User Interface (GUI), shown in Fig. 3, has been implemented for this purpose. It shows the assignments of resources to activities. This is assisted by lock icons on the GUI, providing constraints on the resources who must be assigned to a particular task. For example, the project manager may enforce the *Check test results* activity to a particular resource.



Proposal 1

	Activity	User	Start	End
	Setup project repository	Alice D	16.06.2016	18.06.2016
	Request customer data	Casey C	16.06.2016	26.06.2016
	Engineer system	Jack K	26.06.2016	08.10.2016
	Check need for lab setup	Jack K	08.10.2016	10.10.2016
	Run test	John H	10.10.2016	29.11.2016
	Check test results	John H	29.11.2016	09.12.2016
	Report test results	Alice D	09.12.2016	21.12.2016
	Create release for integration	Alice D	21.12.2016	31.12.2016

Recompute Submit

Fig. 3: Automatic resource assignment, as a result of the scheduling component

After the schedule is confirmed, the allocations take place and the resources can see the tasks appearing in their tasklist. Then, the process can be executed while the process-monitoring component continuously listens to events that occur during the execution of the process. In particular, when a task is finished, the process monitor checks whether the schedule is respected. At the same time, the mining-component updates statistical information from the Camunda logs. For example, after a task is completed, a skill score and an expertise score are updated for the allocated resource. Moreover, statistics are collected, e.g., punctuality of task completion, standard deviation, average duration, percentage of deviations, etc.

3 Maturity and Future Work

SHAPEworks is our first implementation in the context of a BPM scenario which demands for more complexity. Currently it is a prototype that serves mainly as a proof-of-concept. It shows the advantage of having an integrated solution of different approaches implemented on top of a BPMS. We plan to further develop our extension by adding more features, divided into three levels.

Level 1 (current). SHAPEworks includes: *i)* resource (re-)allocation with data and resources from Camunda; *ii)* ontology describing the organizational model; *iii)* process monitoring that triggers alerts when the process risks running late and the process execution does not respect the allocation; and *iv)* mining history of tasks and resources, and updating ontology with mined data.

Level 2. SHAPEworks includes: *v)* allocation of non-human resources, which are fully synchronized with the data from the ontology; *vi)* delay prediction by mining history logs; and *vii)* infrastructure model stored in the ontology.

Level 3. SHAPEworks includes: *viii)* flexible resource (re-)allocation, i.e., allocating resources up to the next decision point of the process, thus enabling a more dynamic schedule; *ix)* mining XOR probabilities, i.e., the likelihood of particular choices made in XOR gates; and *x)* feasibility check using XOR probabilities, i.e., given the likelihood of the path to be executed in the business process and the resources, check if the execution is possible in *n* amount of time.

The Camunda BPMS along with our custom extensions has been deployed on a server and can be used by following this link: http://camunda.ai.wu.ac.at:8080/camunda. Credentials for project managers – administrative users with higher privileges – are ‘demo’ and ‘demo’, respectively user name and password; standard users can access with username same as their name and the string ’password’ as password. A screencast of the tool can be found in http://camunda.ai.wu.ac.at/shapeworks/video.html.

Acknowledgements. This work is funded by the Austrian Research Promotion Agency (FFG) under grant 845638 (SHAPE): http://ai.wu.ac.at/shape-project/

References

1. Bala, S., Cabanillas, C., Mendling, J., Rogge-Solti, A., Polleres, A.: Mining Project-Oriented Business Processes. In: BPM. pp. 425–440 (2015)
2. Cabanillas, C., Mendling, J., Polleres, A., Haselböck, A.: Safety-critical human-and data-centric process management in engineering projects. In: 5th International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA) (2015)

3. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer (2013)
4. Havur, G., Cabanillas, C., Mendling, J., Polleres, A.: Automated resource allocation in business processes with answer set programming. In: 11th International Workshop on Business Process Intelligence (2015)
5. Knublauch, H., Ryman, A.: Shapes Constraint Language (SHACL). Working Draft (work in progress), W3C (2016), https://www.w3.org/TR/shacl/