Event-based Process Simulation of Operating Resources in Assembly Workflows

Philipp Staudt, Sönke Knoch, and Daniel Porta

German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
{firstName.lastName}@dfki.de

Abstract. Business Process Simulation is a useful and widely adopted technique that provides process designers and data analysts with the necessary information to estimate the performance impact of business decisions before they are actually deployed. Manufacturers can benefit from such simulations especially when a strong changeability of the assembly workflows and systems is required. In order to integrate the simulation into existing manufacturing plants, simulations need to consider not only full simulations but also mixed simulations which include virtual and real operating resources. In this work, we present a new simulation framework which allows the process designer to model mixed simulations by defining virtual operating resources. These can be coupled with real hardware allowing the simulation to interact with manufacturing components to test the integrated system for robustness at runtime.

Keywords: BPMN, event-driven simulation, mixed simulation.

1 Introduction

Business process simulation (BPS) is a technique to quantitatively analyze the process in terms of performance measures, such as time, cost, quality and flexibility [2]. BPS generates a large number of process instances executed based on stochastic information about task durations and resource information. Instead of creating the simulation model manually, logs from real process executions may be used to discover those models from historic data using existing process mining techniques [5]. Different simulators exist integrated with the Business Process Model and Notation (BPMN) differing regarding their simulation capabilities [3]. Since the extensibility of such tools is often difficult, [4] suggest an open and extensible process simulator based on a plug-in structure.

Developments aiming at smarter factories require solutions for the concept of changeability. Changing the manufacturing system, e.g. by adding new devices or exchanging existing ones, cost time if the device start-up fails leading to downtimes and further debugging. Thus, a flexible and extensible simulation framework is necessary to allow a virtual start-up in the existing manufacturing system and to test its compatibility with the running system ex-ante.

F. Casati et al. (Eds.): Proceedings of the Dissertation Award and Demonstration, Industrial Track at BPM 2018, CEUR-WS.org, 2018.

We propose a framework allowing mixed simulations running virtual and real resources together. The framework consists of a Web-based dashboard and the Basis Simulation System (BaSS) which mediates between the BPM engine and the virtual as well as the real resources. The framework allows an easy integration of new resources and the adding and exchange of resources at runtime. Different states of resources are coupled through event-based communication with a BPMN process which can be adapted easily by process designers to control the assembly flow simulated entirely or partly.

In the following, we present the framework's architecture and the simulation interface in Section 2. The usage of the system, its graphical user interface and an exemplary assembly process which are part of this demo, are illustrated in Section 3. In Section 4, we draw a brief conclusion.

2 The Framework and the Basis Simulation System

The framework's architecture can be seen in Figure 1. The core of the framework is the BaSS. It contains the BPM Engine Handler, a distributed Data Store and an undefined number of virtual operating resources. Simple events emitted by the BPM engine are processed by the BPM Engine Handler and enriched with assembly-specific data from the data store to form complex events. These complex events are propagated to the respective virtual resources via a common communication interface. The receiving virtual resources can be coupled with real resources. In this case, they act as adapters which transmit the events to the real resources. If there is no real resource, the virtual resource simulates the real one based on the implemented simulation for this resource. This mechanism allows mixed simulations of factory resources. In addition, the Data Store offers a REST-Application Programming Interface (API) for monitoring and configuration purposes and thus, provides valuable data input for visualizing and logging simulated processes aiming at evaluating simulations. A Web-based Dashboard has been implemented to visualize live and historic process data.

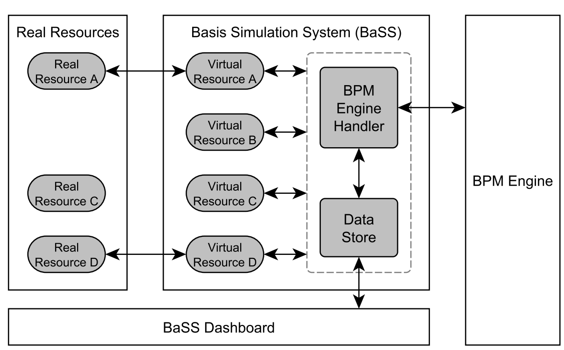

Fig. 1. Overview of the BaSS architecture.

All components within the BaSS are implemented as OSGi services in Java. OSGi is a modular system and a service platform allowing the modular design of the framework in so-called bundles. It enables an easy exchange of bundles, among others representing resources, during runtime which avoids a re-deployment of the framework and thus interruptions of the assembly process.

Communication between the individual components is realized by using the Message Queue Telemetry Transport Protocol (MQTT) protocol. It is a light-weight and well adopted communication protocol in the Internet of Things. As such, it provides an effective way of communicating thanks to the broker-based publish / subscribe architecture and the Quality of Service (QoS) levels offered. The loose coupling of the event-based architecture supports the modular architecture of our framework. High scalability is provided by using MQTT clustering.

A virtual resource is an abstraction of a real operating resource whose API it implements. In addition, each real as well as virtual resource has a uniform life-cycle based on Packaging Machine Language (PackML), cf. [1], which is an industry technical standard developed for the control of packaging machines providing a generic state model also applicable to other resources. If a resource is simulated, it must mimic the behavior of the real one. This is realized by the implementation of the finite state automaton using an OSGi bundle which is identical to the behavior of the real resource. The implementation is the entry point for simulation input in form of stochastic data influencing the simulated behavior of a resource in terms of duration and error-proneness. If not considered for simulation, the virtual resource acts as an adapter and forwards events from the BPM engine to the real hardware.

3 User Interface and Assembly Process

The BaSS and a BPM-Engine Handler, which is based on the Camunda BPM Engine, together with the visualization dashboard shown in Figure 2 is available from our cloud server [1]. After starting the application, the following two Web interfaces are available to the user:

Configuration Admin Apache Felix Console for OSGi bundle configuration. It is used to set-up the BaSS services. It comes with pre-configured bundles for the MQTT communication and the necessary operating resources.

Live Dashboard Used for process management and real time visualization of the process flow. It is connected to the underlying BPM-Engine and can initialize and destroy processes. Important information like the current process execution or task execution times can be watched easily. In addition, audit logs are available showing the process history of the selected process.

For the demo to work, the pre-configured operating resources and the communication must be started. This is done in the configuration admin interface which forms the control center for starting and stopping resources represented by

[1] `https://cloud.dfki.de/owncloud/index.php/s/sZpJZC9HKiqemBS`

the respective bundle implementations. To enable publish / subscribe communication, a suitable MQTT broker must be installed on the system, e.g. ActiveMQ.

Afterwards, the implemented processes can be started. There are three processes included in this demo (see Figure 3):

Initialization Initializes all operating resources according to their PackML state. So, each resource must be reset and started for being able to execute concrete commands.

Main Assembly A press process is simulated in the demo. First, a button needs to be pushed before the press is able to perform the press action.

Finalization After the main assembly process, the operating resources can be completed and stopped according their PackML state model.

4 Conclusion

By providing an easy-to-use framework for the mixed simulation of operating resources in factory plants, operating resources can be tested individually in a real manufacturing context. This may be crucial to avoid wrong expectations on the process which is only simulated entirely. Due to the PackML life-cycle, virtual resources can easily be defined and coupled with real resources. Manufacturers may also provide virtual resources which act as simulated operating resources implementing the state machine of the concrete operating resource. Thereby, simulated and real components share the same API and can be easily integrated in processes models.

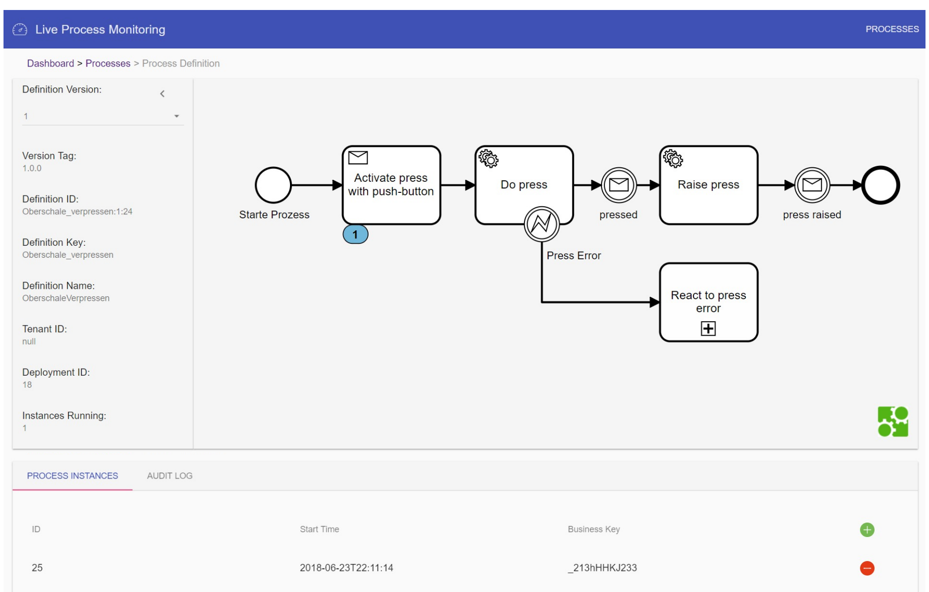

Fig. 2. BPMN Process Dashboard Showing the Simulated Process.

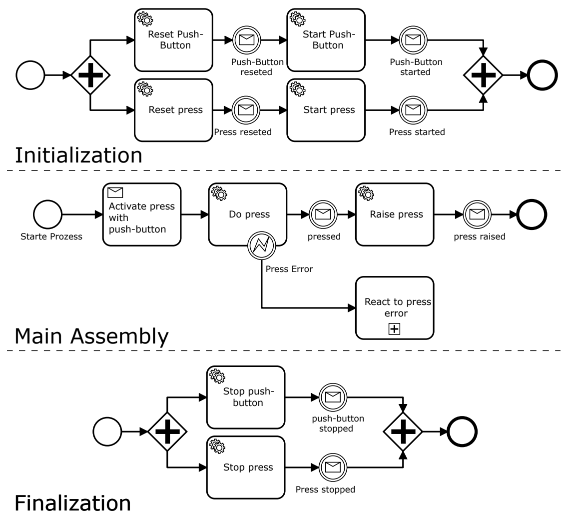

Fig. 3. Complete press process including process initialization (top), the main press process (middle) and the finalization of the operating resources (bottom).

Acknowledgments. This research was funded in part by the German Federal Ministry of Education and Research (BMBF) under grant number 01IS16022E (project BaSys4.0). The responsibility for this publication lies with the authors.

References

[1] ANSI/ISA-TR88.00.02-2015. *Machine and Unit States: An Implementation Example of ANSI/ISA-88.00.01.* 2015. URL: `https://www.isa.org/store/ansi/isa-tr880002-2015,-machine-and-unit-states-an-implementation-example-of-ansi/isa-880001-/43761047` (visited on 06/25/2018).

[2] Marlon Dumas et al. *Fundamentals of Business Process Management, Second Edition.* Springer, 2018.

[3] José Luís Pereira and António Paulo Freitas. "Simulation of BPMN Process Models: Current BPM Tools Capabilities". In: *New Advances in Information Systems and Technologies.* Ed. by Álvaro Rocha et al. Cham: Springer International Publishing, 2016, pp. 557–566.

[4] Luise Pufahl, Tsun Yin Wong, and Mathias Weske. "Design of an Extensible BPMN Process Simulator". In: *Business Process Management Workshops.* Ed. by Ernest Teniente and Matthias Weidlich. Cham: Springer International Publishing, 2018, pp. 782–795.

[5] A. Rozinat et al. "Discovering Simulation Models". In: *Inf. Syst.* 34.3 (May 2009), pp. 305–327.