Actor Based Business Process Modeling and Execution: a Reference Implementation Based on Ontology Models and Microservices

Matthias Geisriegler*, Maksym Kolodiy*, Stefan Stani*, and Robert Singer[†]
FH JOANNEUM – University of Applied Sciences
Dep. of Applied Computer Sciences
Alte Poststraße 147
8020 Graz, Austria
robert.singer@fh-joanneum.at

Abstract—text

I. INTRODUCTION

Current developments in business, the demand for digitalization, and technology driven new business models foster more than ever the need for mature business process management (BPM) methodologies and corresponding supporting technologies for distributed business processes, so-called choreographies. Communication is the very nature of a business process choreography—or, in other words, any choreography is a set of structured communication patterns. That means a choreography defines how work is done, taking into account all involved process participants and systems.

During recent years, tools have emerged to support the execution of business processes, so-called Business Process Management Suits (BPMS). Most of these tools are built around the *de facto* standard for business process modeling languages, namely Business Process Model and Notation (BPMN 2.x).

The standard may be suitable for modeling purposes, but does not directly support the execution of business process models [1] [2] [3]. That means, there is a gap between the conceptual model and the digitized and executable representation. For example, the BPMN standard document provides several compliance classes. Though, the class for execution offers only a rather restricted subset of the modeling classes and no execution support for process collaborations and choreographies [1].

To overcome this weakness, modeling notations based on actor models have emerged. The standard for an actor based approach for business process modeling is the so-called Subject-oriented Business Process Management (S-BPM) approach [4] which not only provides a formal but rather simple, modeling notation and also allows a direct translation into executable code. S-BPM is a mature approach, as has been proven in theory and practice [5] [4] [6] [7].

In this paper, we will discuss how to move away from monolithic software suites towards agile approaches. The central research question is, how to define a BPMS architecture which is fully based on microservice design principles.

To have a common understanding, we will shortly discuss some topics for a shared understanding of the problem domain. Then we will discuss an architecture based on microservice for the execution of process models; the discussion builds on a prototypical implementation of the presented architecture, which is also available for further research. We will not argue why in this case a software architecture based on microservices is more adequate than a monolithic one; we take this for granted and as state of the art for enterprise applications [8] [9].

II. WORKFLOW SYSTEMS

A typical conceptual architecture of an enterprise workflow system as part of a business process management system is based on a monolithic architecture, as described in [10], for example. All known workflow-engines are based on these architectural concepts; therefore, most of them are heavyweight applications with steep learning curves. Moreover, many of these applications have "problems" to realize process collaborations in a practical way. This can easily be proved by trial and is rooted in the underlying concept to conceptualize a business process as *one* finite state machine (FSM).

III. SUBJECT-ORIENTED BUSINESS PROCESS MANAGEMENT (S-BPM)

IV. S-BPM EXECUTION PLATFORM

A. Previous Work

Previously we have presented an entirely functional S-BPM workflow solution[1] using the Microsoft Windows Workflow Foundation functionality as discussed in [11] and [12]. The designed architecture has full enterprise functionality and

---

*Equal contribution
[†]Corresponding author

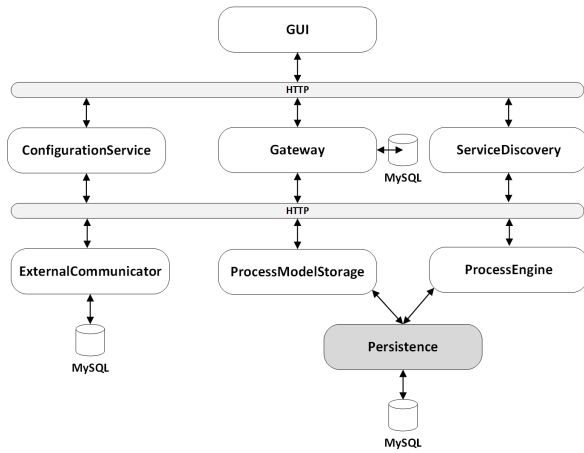[1]https://github.com/InFlowBPM/InFlow-BPMS

Fig. 1. The architecture of the prototype: a collection of microservices.

supports the execution of inter- and intra-company business processes (i.e. collaborations, choreographies).

Nevertheless, the developed prototype is still based on the concepts of traditional workflow systems—e.g. *one* web server for all process instances— and needs many infrastructure components as a functional backbone and is therefore rather costly and has a rather high complexity.

B. Prototype based on Microservices

One major design goal of our architecture is to separate the modeling and the execution platform. As a result, the modeling platform generates an OWL file that can be uploaded to the execution platform by using the management user interface (UI). The management UI provides an interface to control the execution of processes. The individual platforms are based on different sets of technology, as will be described next.

1) Graphical User interface (GUI):

2) Configuration Service:

3) Gateway:

4) Service Discovery:

5) ExternalCommunicator:

6) ProcessModelStorage:

7) ProcessEngine: The ProcessEngine is the most extensive service of the execution platform. This service provides features to start, stop processes, and handles the complete workflow of processes. The Process Engine is based on Akka, which is one of the most popular Actor Model frameworks.

In the Actor Model, all objects are independent, computational units. These units only respond to received messages and do not share a common state. Actors change their state only when they receive a stimulus in the form of a message. So, an actor is a computational entity that, in response to a message it receives, can concurrently [13]:

- send a finite number of messages to other actors
- create a finite number of new actors
- designate the behavior to be used for the next message it receives

V. MODELING PLATFORM

VI. DISCUSSION

With this prototype, we want to stimulate further research in the domain of business process modeling and execution (the prototype is released as open source software). The presented and discussed architecture (which is still under development at the time of writing) is a starting point to define a modern reference architecture for BPMS which are based on microservices.

The microservice approach supports several real requirements for enterprise software: with this work, we have a proof of concept, which demonstrates the practical usability of the microservice approach in the domain of business process management, in concrete in the development of agile workflow systems. Nevertheless, the pre-condition to building an agile BPMS is to change the way how to define process models.

REFERENCES

[1] Object Management Group, "Business Process Model and Notation," http://omg.org, 2013.
[2] B. Silver, *BPMN Method and Style*, 2nd ed. Cody-Cassidy Press, 2011.
[3] E. Börger, "Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL," *Software & Systems Modeling*, 2011.
[4] A. Fleischmann, W. Schmidt, C. Stary, S. Obermeier, and E. Börger, *Subject-Oriented Business Process Management*. Springer, 2012.
[5] A. Fleischmann, *Distributed Systems: Software design and implementation*. Springer, 1994.
[6] A. Fleischmann, S. Raß, and R. Singer, *S-BPM Illustrated*. Springer, 2013.
[7] A. Fleischmann, W. Schmidt, and C. Stary, Eds., *S-BPM in the Wild*. Springer, 2015.
[8] M. Amundsen and M. Mclarity, *Microservice Architecture*. O'Reilly, 2016.
[9] S. Newman, *Building Microservices*. O'Reilly, 2015.
[10] D. Hollingsworth, *The Workflow Reference Model*. Workflow Management Coalition, 1994.
[11] R. Singer, J. Kotremba, and S. Rass, "Modeling and Execution of Multienterprise Business Processes," in *IEEE 16th Conference on Business Informatics (CBI)*, vol. 2, 2014, pp. 68–73.
[12] R. Singer and S. Raß, "Structured Communication—Approaching S-BPM with Microsoft Technologies," in *S-BPM in the Wild*, A. Fleischmann, W. Schmidt, and C. Stary, Eds. Springer, 2015, pp. 235–255.
[13] M. K. Gupta, *Akka Essentials*. Packt Publishing, 2012.