

# CPEE - Cloud Process Execution Engine

Juergen Mangler\*, Stefanie Rinderle-Ma

University of Vienna, Faculty of Computer Science,  
Research Group Workflow Systems and Technology, Vienna, Austria,  
{juergen.mangler, stefanie.rinderle-ma}@univie.ac.at

**Abstract.** Cloud-Enabled BPM Platforms are becoming important corner-stones in allowing IT leaders to extend their cloud strategy to cover business processes and their execution. While the hopes are high, software providers approach the hype with rather traditional tools, that expose well-known BPM solutions as SaaS and combine them with ever more refined analysis and monitoring tools.

This demonstration will highlight experiences and considerations how to design a cloud based BPM platform that can truly serve as a launchpad for future innovation. The Cloud Process Execution Engine (CPEE) is an open source versatile cloud based BPM platform, that has been in development for over six years, and has been used in several national and international projects.

## 1 Introduction

Cloud computing has become one of the most exciting IT trends nowadays. As stated by Gartner [2], "[v]endors will [...] evolve their platforms to be cloud-capable". An immediately resulting question for the Business Process Management (BPM) community is: how can we move business processes to the cloud or more precisely, how can we offer process management technology as a service?

Subsumed by the term bpmPaaS (Business Process Management Platforms as a Service), providers like IBM<sup>1</sup>, Oracle<sup>2</sup> and Appian<sup>3</sup> typically tackle this question quite traditionally:

- The backend of the BPM solution (engine, process repository, activity repository) as well as the data (databases) these tasks access, are deployed to virtual machines in the cloud (IaaS).
- The frontends (Process Editor, Simulation and Optimization Tools, Worklists, Monitoring) are provided as Web Applications (SaaS).
- Third party products (e.g. CRMs from ORACLE, SAP) are supported through a magnitude of special predefined tasks (adapters), that can access and modify data from these products.

---

\* Copyright © 2014 for this paper by its authors. Copying permitted for private and academic purposes.

<sup>1</sup> <http://www-03.ibm.com/software/products/en/business-process-manager-cloud>

<sup>2</sup> <http://www.oracle.com/us/technologies/bpm/overview/index.html>

<sup>3</sup> <http://www.appian.com/bpm-software/cloud-bpm.jsp>

The architecture and principles of how to build BPM Solutions so far appear to be quite untouched by the hype, and basically are still based on Workflow Management Coalition (WfMC) considerations [3]. Innovations like the advent of better modeling languages (BPMN), adaptive processes and evolution [1], unified process constraint management [8], or inter-instance/process synchronization [9,5,6] are being slowly built into the rather monolithic cores of these products. Sometimes (e.g. in the case of BPMN) they even spawn new process engines<sup>4</sup> that try to directly support a particular feature.



<http://cpee.org>

With the Cloud Process Execution, we try to transparently evolve the architecture and flexibility of BPM by analyzing how the following questions affect established BPM principles:

1. How to support multiple or custom modeling languages with custom execution semantics, or special purpose modeling tools (e.g. for doctors to describe patient treatments)?
2. How to support script tasks in multiple or custom languages (e.g. javascript, java, go, swift)?
3. How to build an execution environment that supports horizontal scalability (millions of instances on distributed nodes)?
4. How to adapt monitoring to different situations: from processes that orchestrate the monitoring of sensor networks to environments with specific security requirements.
5. How to support process execution shaping: the ability for custom logic to delay tasks (e.g. for synchronization purposes) or repair running process instances (i.e. self-healing beyond exceptions by modifying the model)?
6. How to migrate process instances at runtime between nodes/engines?
7. How to enforce process constraints that cover aspects of the process structure, runtime (e.g. separation/binding of duty), or the Worklist (assignment rules).
8. How to create custom Worklists, and integrate them in various SaaS settings (e.g. customer vs. vendor point of view).
9. How to implement custom optimization and simulation capabilities, that e.g. forecast process properties based on partners business intelligence solutions.

---

<sup>4</sup> e.g. <http://www.activiti.org/>

Allowing to customize all these aspects of BPM is well beyond the state of the art of existing single (commercial or academic) process management solutions. With the Cloud Process Execution Engine we tried to carefully design a robust toolkit, to build cloud based BPM solutions. The goal was to solve the questions above as well as questions that may arise in the future, through a minimal event-based RESTful web interface. We fully explore the concept of message based systems, to allow (per instance) for custom monitors, process models in different languages<sup>5</sup>, and runtime execution analysis that can intervene and shape the execution.

While the above mentioned aspects are very important for researchers to be able to quickly explore new ideas and concepts without the burden of customizing and extending existing complex BPM infrastructure, the same properties also proved to be very usable for rapid innovation in applied / business settings.

## 2 Maturity and Significance to the BPM Field

The CPEE is a mature piece of software that has been or is actively used (in conjunction with companies) in (among others) the following projects:

ADVENTURE FP7-ICT project (Grant agreement no. 285220), with a multi-national consortium including several SME's and research partners. The project deals with virtual factories in the manufacturing domain.

Adaptive Care Planning project, funded by the Austrian Research Promotion Agency (FFG), in conjunction with a Care Center of elder people (Pflegezentrum Pottenbrunn GmbH). The project deals with introducing highly custom and adaptive patient processes into the medical domain.

C3Pro project, co-funded by the Austrian Science Fund FWF (lead agency, project number I743) and the German Research Foundation DFG. The project deals with change propagation in process choreographies.

Furthermore the CPEE is heavily used in teaching master courses at the University of Vienna. The sustainability of the core concepts have been evaluated [7,10] and proved through an independent third party engine implementation [4].

## 3 Screencasts, Documentation

The Cloud Process Execution Engine, including a working demo for creating simple processes that access REST and SOAP services, as well as a multitude of screencasts can be found at:

---

<sup>5</sup> e.g. BPEL, BPMN2, custom languages; this includes the exploration of different execution semantics, e.g. for pure modelling languages

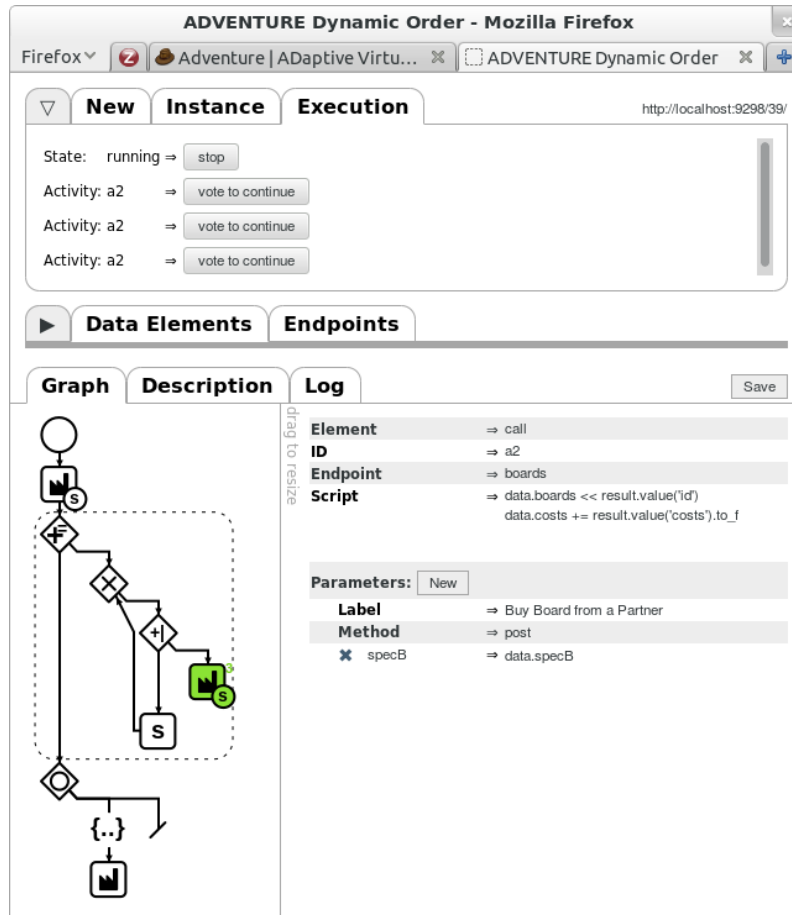


Fig. 1: CPEE Cockpit

<http://cpee.org>

For a screenshot of the purely javascript based adaptation and monitoring cockpit UI see Fig. 1.

Based on the on-line usable prototype a multitude of concepts can be tested:

- Rule-based inter-instance synchronization, enacted by a rule engine that listens to events emitted during execution, calculates optimal queues and transparently delays the execution of single tasks.
- Custom Worklists.
- Support for different process models / languages.
- Support for different protocols, like Advanced Message Queuing Protocol (AMQP, the basis for advanced Enterprise Service Buses), Extensible Messaging and Presence Protocol (XMPP) and HTTP, for accessing all aspects

- of the process engine itself (creating new instances, monitoring, calling services through tasks).
- Monitoring the process engine through client side javascript applications.
  - Runtime adaptation of process instances.

## Acknowledgements

This work was partially supported by the Commission of the European Union within the ADVENTURE FP7-ICT project (Grant agreement no. 285220).

## References

1. Dadam, P., Reichert, M., Rinderle, S., Jurisch, M., Acker, H., Göser, K., Kreher, U., Lauer, M.: Towards truly flexible and adaptive Process-Aware information systems. In: Int'l Conference UNISCON. p. 72–83 (2008)
2. Gartner: Cloud computing (2010), [http://www.gartner.com/it/products/research/cloud\\_computing/cloud\\_computing.jsp](http://www.gartner.com/it/products/research/cloud_computing/cloud_computing.jsp), [Last accessed: 17.03.2010]
3. Hollingsworth, D.: Workflow management coalition: The workflow reference model. Document Number TC00-1003, Document Status - Issue 1.1, Workflow Management Coalition (1995), <http://www.wfmc.org/Download-document/TC00-103-Interface-4-Workflow-InteroperabilityThe-Workflow-Reference-Model.html>
4. Kissinger, B.: Implementation of a modern workflow engine using erlang (2012), <http://www.impactsoftwarelabs.com/wee/Implementation%20of%20a%20Modern%20Workflow%20Engine.pdf>
5. Leitner, M., Mangler, J., Rinderle-Ma, S.: Definition and enactment of instance-spanning process constraints. In: Wang, X.S., Cruz, I., Delis, A., Huang, G. (eds.) Web Information Systems Engineering - WISE 2012, pp. 652–658. Lecture Notes in Computer Science, Springer Berlin Heidelberg (Jan 2012), [http://link.springer.com/chapter/10.1007/978-3-642-35063-4\\_49](http://link.springer.com/chapter/10.1007/978-3-642-35063-4_49)
6. Mangler, J., Rinderle-Ma, S.: Rule-based synchronization of process activities. In: 2011 IEEE 13th Conference on Commerce and Enterprise Computing (CEC). pp. 121–128. IEEE (Sep 2011)
7. Mangler, J., Stuermer, G., Schikuta, E.: Cloud process execution engine-evaluation of the core concepts. Arxiv preprint arXiv:1003.3330 (2010)
8. Mangler, J., Rinderle-Ma, S.: IUPC: identification and unification of process constraints. Tech. rep. (Apr 2011), <http://arxiv.org/abs/1104.3609>
9. Rinderle-Ma, S., Mangler, J.: Integration of process constraints from heterogeneous sources in process-aware information systems. In: Int'l Workshop Enterprise Modelling and Information Systems Architectures - EMISA 2011. LNI, GI (2011)
10. Sturmer, G., Mangler, J., Schikuta, E.: Building a modular service oriented workflow engine. In: Service-Oriented Computing and Applications (SOCA'09), IEEE International Conference on. p. 1–4 (2009)