Workflow-Processing and Verification for Safety-Critical Engineering: Conceptual Architecture

Deliverable D6.1

FFG – IKT der Zukunft
SHAPE Project
2014 – 845638

Table 1: Document Information

Project acronym:	SHAPE
Project full title:	Safety-critical Human- & dAta-centric Process management in Engineering projects
Work package:	6
Document number:	6.1
Document title:	Workflow-Processing and Verification for Safety-Critical Engineering: Conceptual Architecture
Version:	1
Delivery date:	01 October 2015 (M2)
Actual publication date:	
Dissemination level:	Public
Nature:	Report
Editors / lead beneficiary:	SIEMENS AG
Author:	Tudor Ionescu
Reviewers:	Cristina Cabanillas, Alois Haselboeck

Contents

1. Introduction

In the context of the SHAPE project, the current document represents the deliverable *D6.1 Workflow-processing and verification for safety-critical engineering – conceptual architecture*, which is one of the results of *T6.3 Realization of workflow-processing and verification architecture*. The overall goal of work package 6 is the integration of various tools, components, and algorithms into a novel architecture for flexible process execution and verification.

In the mobility industry, notably the railway domain, collaborative processes are often described in natural language and stored in conventional documents. Documents are complemented by emails and meeting minutes exchanged between project stakeholders. Execution logs of past processes also contribute to this unstructured repository of process information. The tacit knowledge of experienced employees finally represents the gluing factor that ensures the successful finalization of processes.

In the railway domain, non-functional requirements, such as safety, reliability, certifiability, and standard compliance of both the systems and the business processes used in creating them are key to the success of products and projects. Fulfilling these non-functional requirements using traditional process execution based on unstructured process information is extremely costly and time-consuming. From these reasons, the automation and optimization of business processes for developing railway systems are constantly sought after in large business organizations.

1.1. Challenges

The main problems with process automation and optimization based on unstructured information are:

- The difficulty of extracting semantic process models from process information in unstructured data stored in handbooks and logbooks manually compiled by engineers; and from execution logs of tools and past projects.
- Enacting the automation and optimization of business processes according to the semantic models obtained through process mining.
- Seamlessly and dynamically adapting running processes whenever (1) unexpected potentially harmful situations occur, (2) new insights are gained by means of process mining, and (3) new safety compliance requirements become available.

The list of architectural requirements and architectural design decisions aimed at capturing and providing solutions for these challenges is provided as a separate technical annex.

1.2. Purpose of this Document

The main purpose of this document is to provide a technical audience with information about an architecture for a workflow-processing and verification system for safety-critical engineering. This document represents the foundation for a detailed design document (Deliverable D6.2).

2. Conceptual Architecture

To address the aforementioned challenges, we have designed an architecture composed of 3 types of procedures designed as business processes which run in parallel and interact with each other. This 3-process model is shown in figure 1.

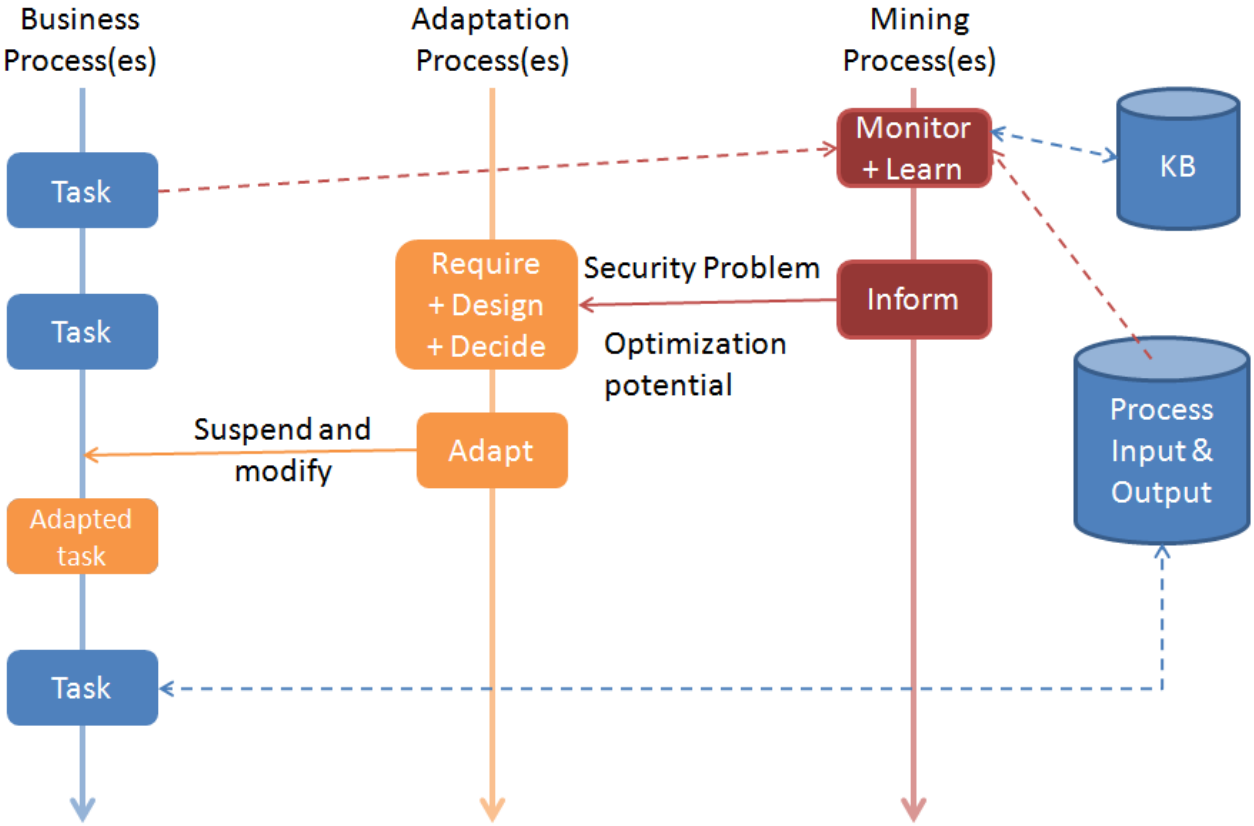

Figure 1: The three-procedure model for mining, optimizing, and adapting business processes.

A business process for configuring railway interlocking systems (from now on interchangeably referred to as *production procedure*) has been modeled in the BPMN (Business Process Model and Notation) language and implemented using the Camunda Suite. In this implementation, many of the tasks carried out manually before were automated and the process was deployed on an application server in the business unit. A key feature of BPMN suits like Camunda is represented by visual semantics: the workflows designed using visual model editors are seamlessly converted into executable code (e.g. Java byte code). In the proposed solution, semantic technologies are used to infer semantic process models, which refine existing models at runtime. The inferred models can be converted into executable code as well and injected into running process models without halting or restating them.

To achieve this, besides the production procedure, two additional procedures are implemented as business processes: *Process Mining* and *Optimization and Adaptation*. Unlike the mining process and the optimization & adaptation procedures (which are unique), the production

procedure (which covers all the potential business processes running in the organization) may change as a results of the process mining, optimization, and adaptation activities.

- Process mining

    - This procedure is also implemented using BPMN and Camunda and runs in parallel with the production business processes. Its main task consists of mining the unstructured process information repository and analyzing the logs of the production processes as well as other data from the repository, including emails and meeting minutes. The results of these analyses are used to inform the optimization and adaptation procedure below.

- Optimization and adaptation

    - This procedure uses the information from the mining procedure to optimize and adapt the main process. Adaptation is realized without interrupting the main process by scheduling new tasks or cancelling existing ones depending on newly discovered semantic process information. Adaptation preconditions are inferred by a semantic rule engine.

The mining process analyses data stemming from a variety of active and past production business processes using data mining techniques, such as text mining, clustering, classification, concept linkage, etc. These techniques are applied to the different kinds of data from the unstructured process information repository. Process mining allows the discovery and inference of new processes and the enhancement of existing ones.

The communication between the adaptation and the mining processes is realized using the “publish-subscribe” pattern: the mining process publishes new data mining results to the subscribing adaptation processes. While a single mining process is active at any moment, there can be several productive business processes coupled with one or several adaptation processes. In the latter case, the adaptation processes can each focus on a single concern, such as security or standard compliance. The adaptation processes use code injection to modify the behavior of the main productive processes at runtime. They may also require programming tasks and can therefore be stopped and restarted at any time, while the mining and the main processes are running. To determine the best adaptation strategy, programmers may also need to consult process experts for matching the mined semantic process models with real execution constraints. In this context, the key success factor for the project is the tight collaboration between scientists, software architects, programmers and the process experts from the business unit.

The proposed solution helps to reduce the process execution time and costs through process automation and optimization. For safety-critical processes, the adaptation techniques employed

help to seamlessly solve unexpected safety-related issues. This is facilitated by semantic technologies and a strict separation of concerns using the proposed 3-procedure method.

2.1. Document Structure

The following architecture description is based on the 4+1 Views Model [1]. Architectural views are used to describe software architectures from the vantage point of different stakeholders in the development lifecycle, whereby each view may address different cross-cutting design concerns.

The following architectural views will be used to describe the architecture of the SHAPE system:

- **Logical view:** The logical view is concerned with the functionality that the system provides to end-users. UML Diagrams used to represent the logical view include Class diagram, Communication diagram, Sequence diagram [1].
- **Development view:** The development view illustrates a system from a programmer's perspective and is concerned with software management. This view is also known as the implementation view. It uses the UML Component diagram to describe system components. UML Diagrams used to represent the development view include the Package diagram [1].
- **Process view:** The process view deals with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the runtime behavior of the system. The process view addresses concurrency, distribution, integrators, performance, and scalability, etc. UML Diagrams to represent process view include the Activity diagram [1].
- **Physical view:** The physical view depicts the system from a system engineer's point of view. It is concerned with the topology of software components on the physical layer, as well as the physical connections between these components. This view is also known as the deployment view. UML Diagrams used to represent physical view include the Deployment diagram [1].

Starting from selected use cases, architecturally relevant requirements are derived (cf. Annex A: Architectural Requirements), which serve as drivers for architectural design decisions or key design decisions (cf. Annex B: Architectural Design Decisions). These decisions are aimed at fulfilling the architectural requirements and are classified in the document by referring to one of the aforementioned architectural views. This approach corresponds to the decision view in software architecture [2].

Annexes C provides a collection of architectural styles, design patterns, and methods considered useful for implemented the architectural design decisions from Annex B. Finally, annex D. provides an qualitative analysis of the architectural qualities of the system with respect to the design decisions taken.

3. Relevant Use Cases

In this version of the specification, we focus on an adaptation use case, whereby the adaptation is triggered by a finding obtained through process mining. Such a finding could reflect one of the following situations:

- A manual task modeled as a user task in BPMN can be automated and implemented as a service task.
- An existing service task does not entirely fulfill the requirements of the productive process and needs to be adapted accordingly.
- A potential safety hazard is identified in the production procedure and the productive process needs to be adapted accordingly.
- New regulations for the railway industry go into effect and the production process has to be adapted accordingly.

In all these cases, the adaptation must be possible without disrupting the main business process.

4. Summary and Future Work

In the mobility industry collaborative processes are often described in natural language and stored in Word and PDF handbooks and logbooks. This unstructured information is complemented by emails and meeting minutes resulting from the communication between project stakeholders (customers, managers, engineers). Execution logs of past processes also contribute to this unstructured repository of process information. In the railway domain, non-functional requirements, such as safety, reliability, certifiability, and standard compliance of both the systems and the business processes used in creating them are key to the success of products and projects. As the fulfillment of these non-functional requirements is extremely costly and time-consuming, the automation and optimization of business processes for developing railway systems are constantly sought after in large business organizations.

To enable automation and optimization of a business process for configuring railway interlocking systems, a BPMN (Business Process Model and Notation) workflow was implemented using the Camunda Suite, which supports visual semantics and executable code generation from BPMN models. In the proposed solution, semantic technologies are used to infer semantic process models, which refine existing models at runtime. The proposed solution helps reduce the process execution time and costs through process automation and optimization. This is facilitated by semantic technologies and a strict separation of concerns using a 3-process approach: (1) a productive process monitored by (2) a mining process and dynamically refined by (3) an adaptation process.

In D6.2 we will elaborate on the methods and mechanisms for implementing the three-procedure model described in this deliverable. These elaborations will mostly concern the logical and the developmental views.

Bibliography

[1] P. B. Kruchten, "The 4+ 1 view model of architecture.," *IEEE Software,* vol. 12, no. 6, pp. 42-50, 1995.

[2] P. Kruchten, R. Capilla and J. C. Dueas, "The decision view's role in software architecture practice," *IEEE Software,* vol. 26, no. 2, pp. 36-42, 2009.

Technical Annex

A. Architectural Requirements
B. Architectural Design Decisions
C. Architectural Styles and Design Patterns
D. Architectural Quality Attributes