



Development methods for process-driven and composite mashup applications

Motivation

- Composite and service-oriented web applications (e. g. user interface mashups) apply the SOA paradigm at user interface layer
- *User Interfaces Services (UIS)* represent data, functions and user interface components that can be integrated dynamically and context-dependently
- Mashups provide a frontend for service-oriented architectures particularly in the context of business processes
- Problems need to be solved:
 - » Development of composite web applications is unstructured and detached from model-based requirements analysis
 - » Existing model-driven development methods do not address composite, process-oriented rich internet applications

Objectives

Model-driven development method

- Lightweight and iterative development process for process-oriented and composite mashup applications
- Semi-automatic transformations from a workflow to a composite web application bridging the gap between process model and user interface composition model
- Tool support and involvement of domain experts and end users
- Integration of adaptivity and quality aspects during the authoring process
- Supporting the authoring process by managing and reusing user interface components and other artefacts

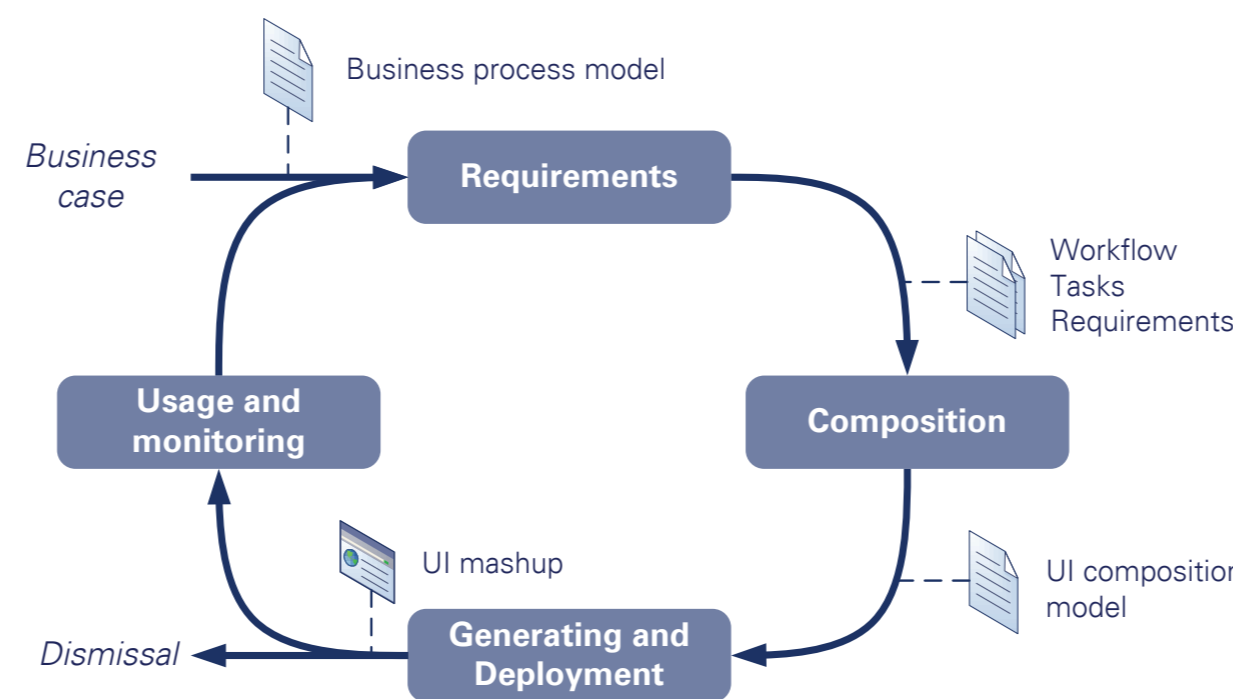
Architecture and runtime environment

- Development of an integration environment for workflow engines and mashup runtimes in order to synchronize and execute process-oriented mashup applications
- Approaches for monitoring and assuring quality aspects and for optimizing the user interface

Concept

Development process

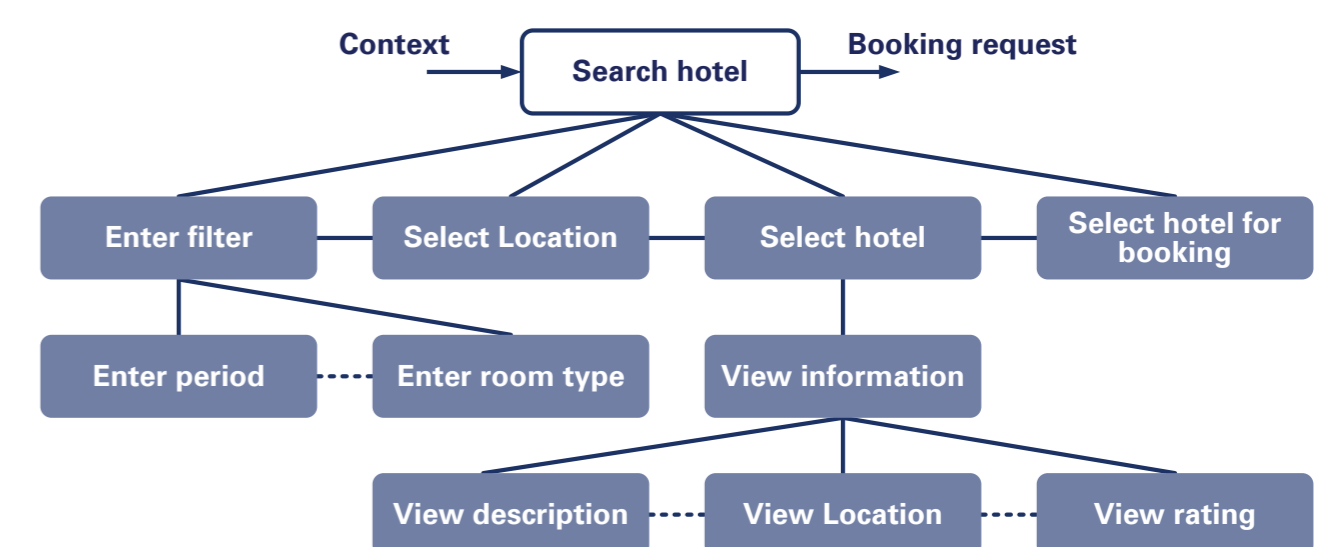
- *Requirements analysis* by decomposing human activities of business processes into UI-oriented task models
- *Composition* by finding adequate UIS on the basis of the task specification
- *Monitoring and assurance* of quality aspects defined in the task specification



Development process for process-oriented and composite mashup applications

Task Model

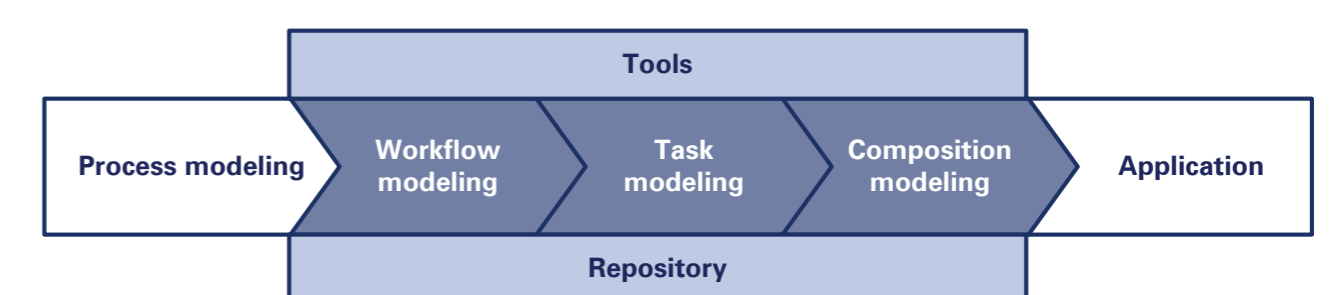
- Refinement of business process model by requirements for user interface components
 - » Hierarchical decomposition
 - » Temporal constraints
 - » Data in- and outcome
 - » Pre- and post-conditions
 - » Involved roles, users and services
 - » Goals and Actions
 - » Quality requirements
- Semantic description of a UI composition for authoring, management and reuse of existing solutions



Example of a task model for in the context of a business process activity

Authoring process

- Development of an authoring tool for modeling tasks
- Evaluate and extend existing tools for workflow and UI modeling
- Semi-automatic transformations for creating skeletons of UI compositions



Authoring process

Runtime environment

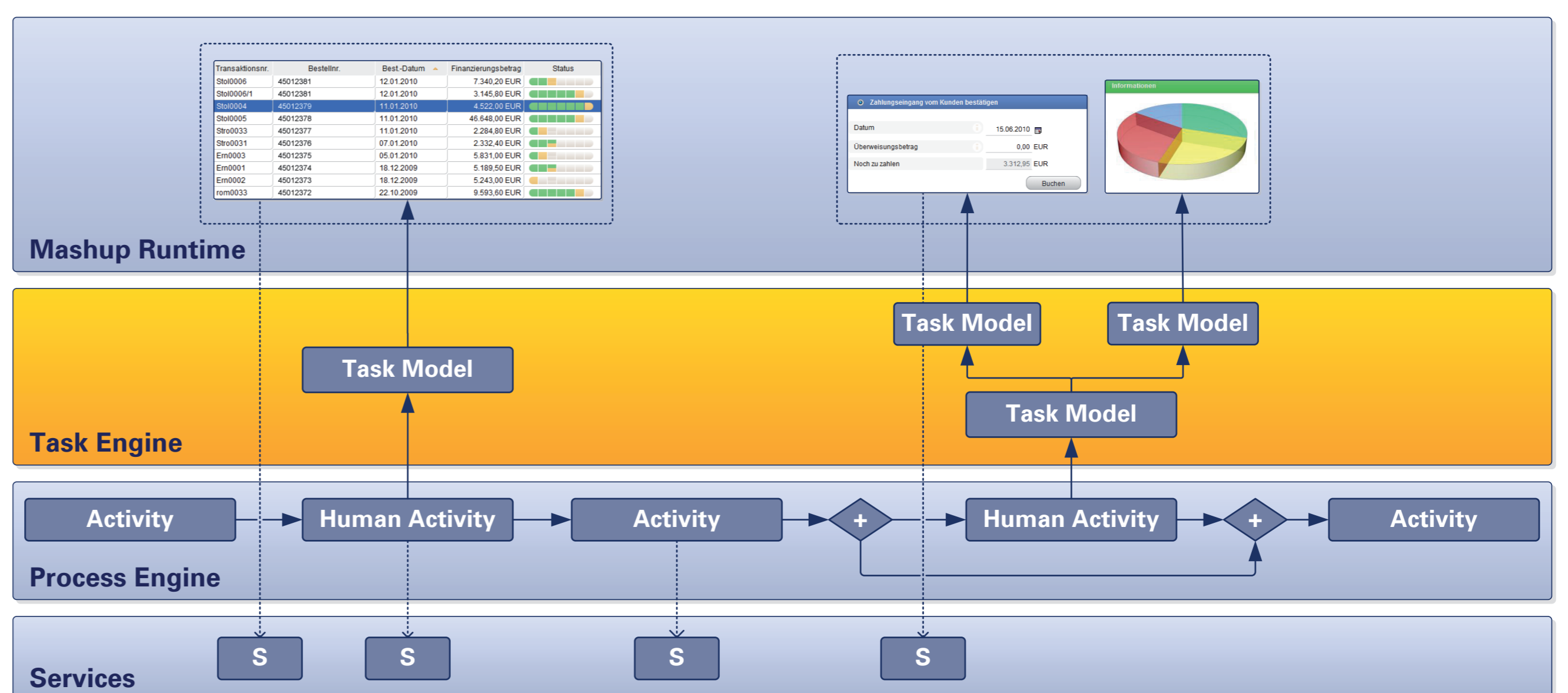
- Using existing solutions for workflows (e.g. BPEL engines) and UI mashups (e.g. CRUISe-Runtime)
- Development of an integrated runtime for separation and synchronisation of UI mashups and workflows
- Extending the user interface description of UIS and the *UIS Registry* for finding adequate and task-based UIS

Project is funded by:

Saxonia Systems
Wir lieben IT.

Europa fördert Sachsen.
ESF
Europäischer Sozialfonds

(ESF-080939514)



Bridging the gap between UI mashups and workflows with the help of task models