

## RESTification of JEngine

### Background

Today, business process management is a key approach to organize work, and many companies represent their operations in business process models. Recently, choreography diagrams have been introduced to represent interactions between business processes, run by different partners. While there is considerable work on using process models during process implementation, there is little work on using choreography models to implement interactions between business processes.

On the other hand, the REST architectural style has gained widespread acceptance as the foundation for building Web-based applications and so-called RESTful Web services or RESTful Web APIs. We have already introduced a novel approach to enrich BPMN choreography diagrams with annotations that facilitate capturing information required for an efficient implementation of REST-based information exchanges between two or more interacting business partners. We also have an existing engine (JEngine) that takes a process model as input and provides with a user interface to execute the process.

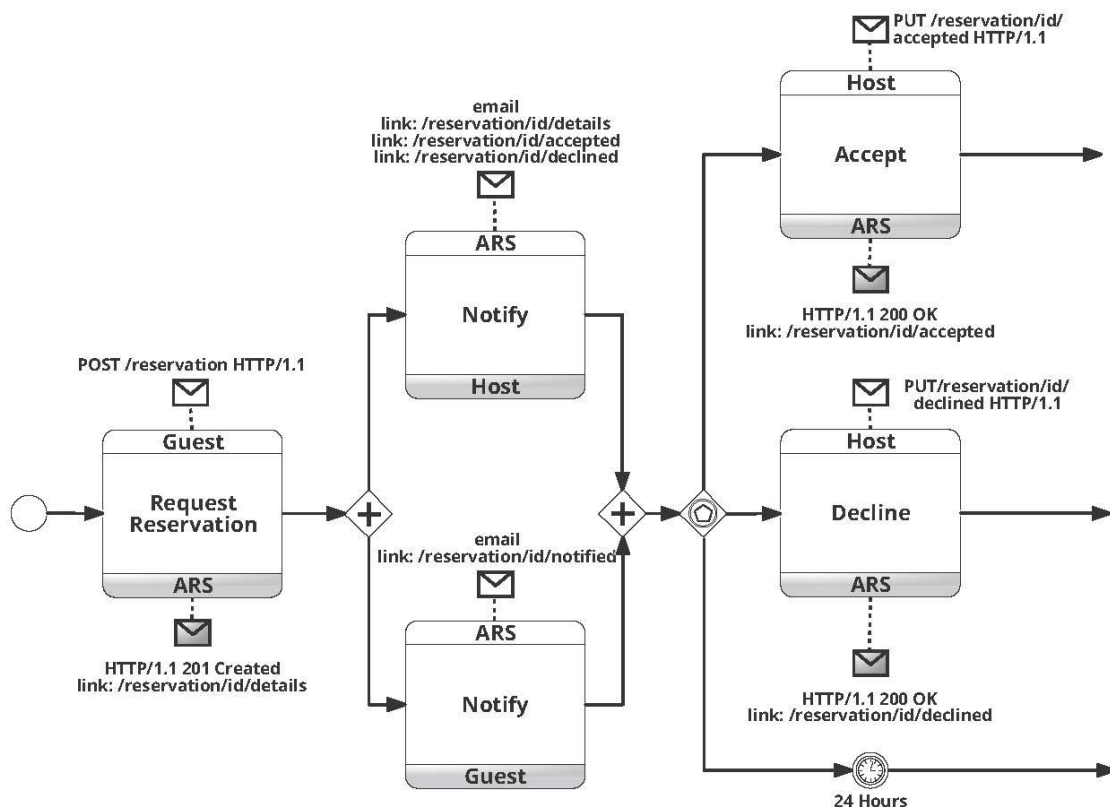


Figure 1: Part of a RESTful Choreography Diagram

Right now, we want to extend the JEngine with regard to the functionality of providing with a REST based interface automatically generated from annotated choreography models. This approach will facilitate the implementation even more.

## Project Goal

The goal of the project is to mitigate the gap between model level and implementation level. We use a model-driven approach where we start from the RESTful choreography diagrams and generate codes directly usable by JEngine. To be specific, we want to develop a plug-in for the engine that will take the choreography diagrams annotated with REST artifacts as input. We intend to come up with a series of message templates containing the code structure needed for the corresponding REST interaction designed in the choreography diagram, e.g., POST, PUT, GET, DELETE or Emails. JEngine then uses these templates to provide with the basic RESTful interface which can be tailored by the developers afterwards according to their specific needs.

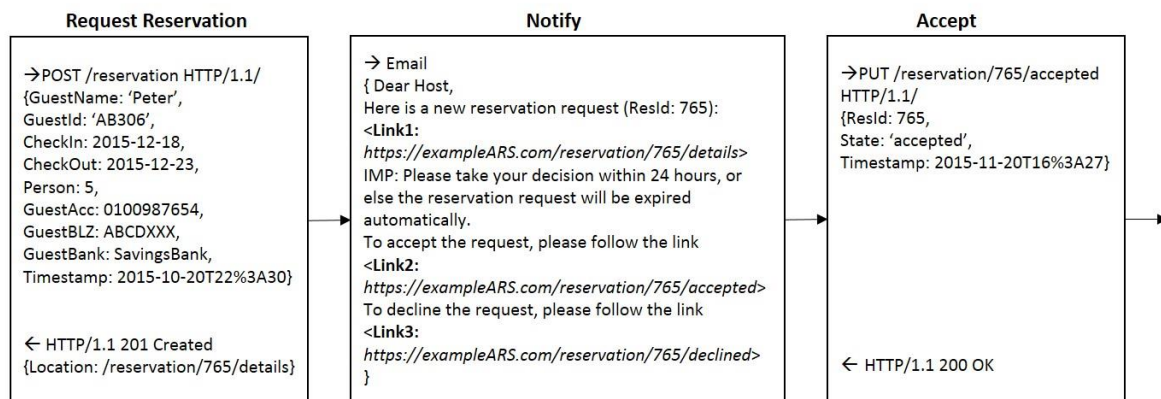


Figure 2: Message Templates Corresponding to Fig. 1 (Incomplete)

## Tasks

1. Design the extended architecture for JEngine
2. Create mapping between the as-is and to-be inputs for JEngine
3. Extend the engine based on the architecture and the mapping
4. Evaluate the usefulness and drawbacks of the approach

## Contact

- Professor Mathias Weske (mathias.weske@hpi.de)
- Sankalita Mandal (sankalita.mandal@ hpi.de)
- Adriatik Nikaj (adriatik.nikaj@ hpi.de)