ABCD Meeting

Introduction to Savara

Testable Architecture
January 2014

Scribble Open Source Project

Home: http://www.scribble.org

Work in progress, but includes links to:

- Language specification
- Discussion forum (google group)
  - Subscribe for release notifications
- Implementations
  - Java
  - Python

Background

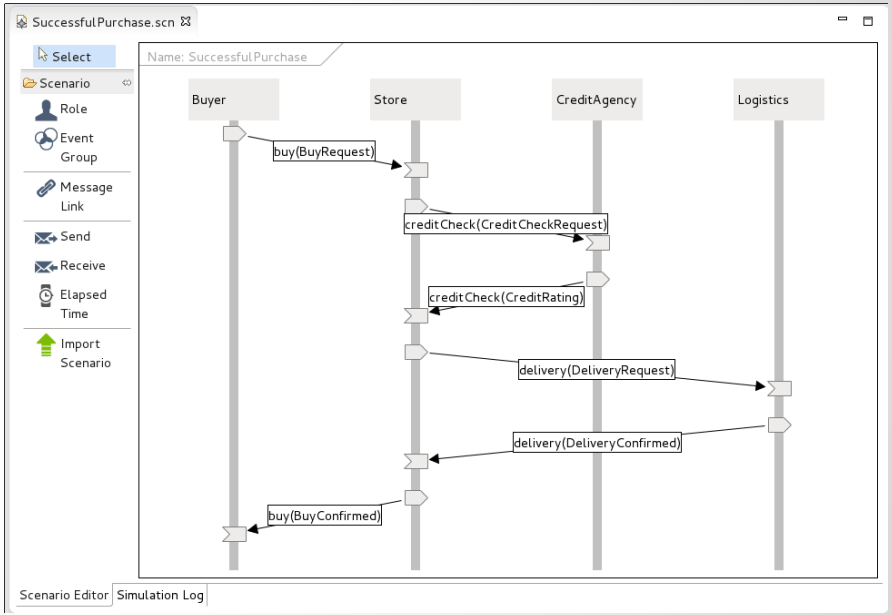
- Project established in 2009 as collaboration between Red Hat and Cognizant
- Aim to provide tooling support for Testable Architecture
  - “ensure artifacts at all stages of the development lifecycle can be verified against each other to ensure deployed system meets original requirements”
- Based on second iteration of Scribble

Scribble Version Differences

- Theoretical basis was not completed
  - making it harder to derive ‘valid’ artifacts from the descriptions
- Concepts currently missing from latest version:
  - introduces - means of connecting to a new role in the middle of a protocol
  - repeat - syntactic sugar, but needs to be considered potentially for easy of use
  - global escape - a “try/catch” construct to break out of some behaviour based on the occurrence of some situation (timeout, cancel pattern)
  - fork/join - was not part of previous version, but implemented as an extension in Savara
  - conditional - was not part of previous version, but again may make it easier to describe some protocols

Capturing requirements

- Scenarios created to describe the different paths through a business process
- Links define message type and have an associated example message

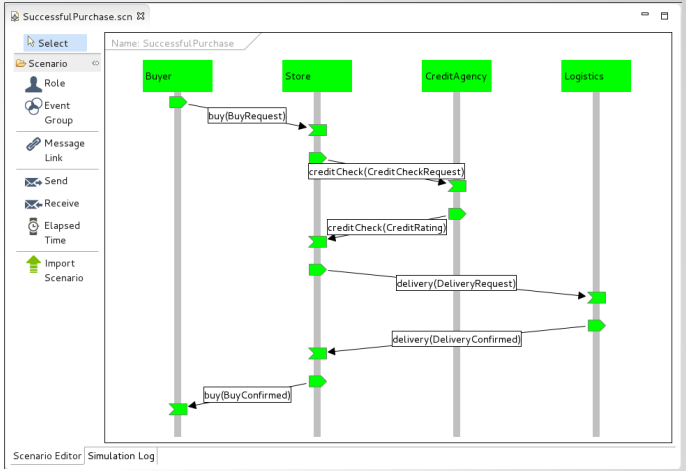


Deriving BPMN2 Models

- We use a set of scenarios to derive the behaviour of multiple roles
- Used to create a set of BPMN2 Process Models, one per role
- Set of BPMN2 Process Models then aggregated to create a BPMN2 Choreography
- Benefit of this approach is that we can use the same mechanism to create a Choreography from other endpoint descriptions (e.g. BPEL process definitions)

Verify Choreography

- Choreography can be verified against the initial requirements (scenarios) by simulation
  - Endpoint simulator (state machine) created for each role involved in a scenario, driven by projected scribble local protocol definitions
  - Example messages associated with links in the scenario are evaluated against the appropriate role simulators
  - Valid simulation turns node green, invalid turns the node red



Generate and Verify Endpoints

- Projected local descriptions can be used to generate:
  - Service design artifacts (e.g. BPMN2 Process Model)
  - Service implementation skeletons:
    - BPEL process definitions
    - Java code with behaviour

- Simulate scenarios against service design and artifacts
  - Appropriate endpoint simulator instantiated
  - If endpoint is a service implementation, then sent messages intercepted and compared against expected message defined in scenario

Future Governance Integration (1)

- Design time governance
  - SRAMP based SOA Repository
  - Manages artifacts, their associations and additional metadata
  - Customisable governance (BPMN2) workflows used to manage artifacts through lifecycle, from development, testing, production to retired.
- Future areas for investigation
  - Static impact analysis
    - Interfaces - determine whether service interfaces have changed in an incompatible manner
    - Service behaviour incompatibility - great in theory, but potentially not useful in practise

Future Governance Integration (2)

- Runtime governance
  - Capture execution events to understand how a business transaction has behaved
  - Extract correlation information to understand which events belong to what conversation
- Future areas for investigation
  - Dynamic impact analysis
    - Reconstruct business transaction using captured events and correlation information
    - Compare behaviour of business transaction as a subset of the global behaviour defined by a Choreography
    - Constraint validation, enforcement and reaction

Further Information

Website: http://www.jboss.org/savara

Demo: http://vimeo.com/53007939

(Requirements to Deployed Services in 15 minutes)