Behavior-driven Business Process Development mit BPMN

W-JAX
München, 04.11.2014

Dr. Daniel Lübke – Senior Consultant
Tammo van Lessen – Senior Consultant

About us

ENTERPRISE BPM

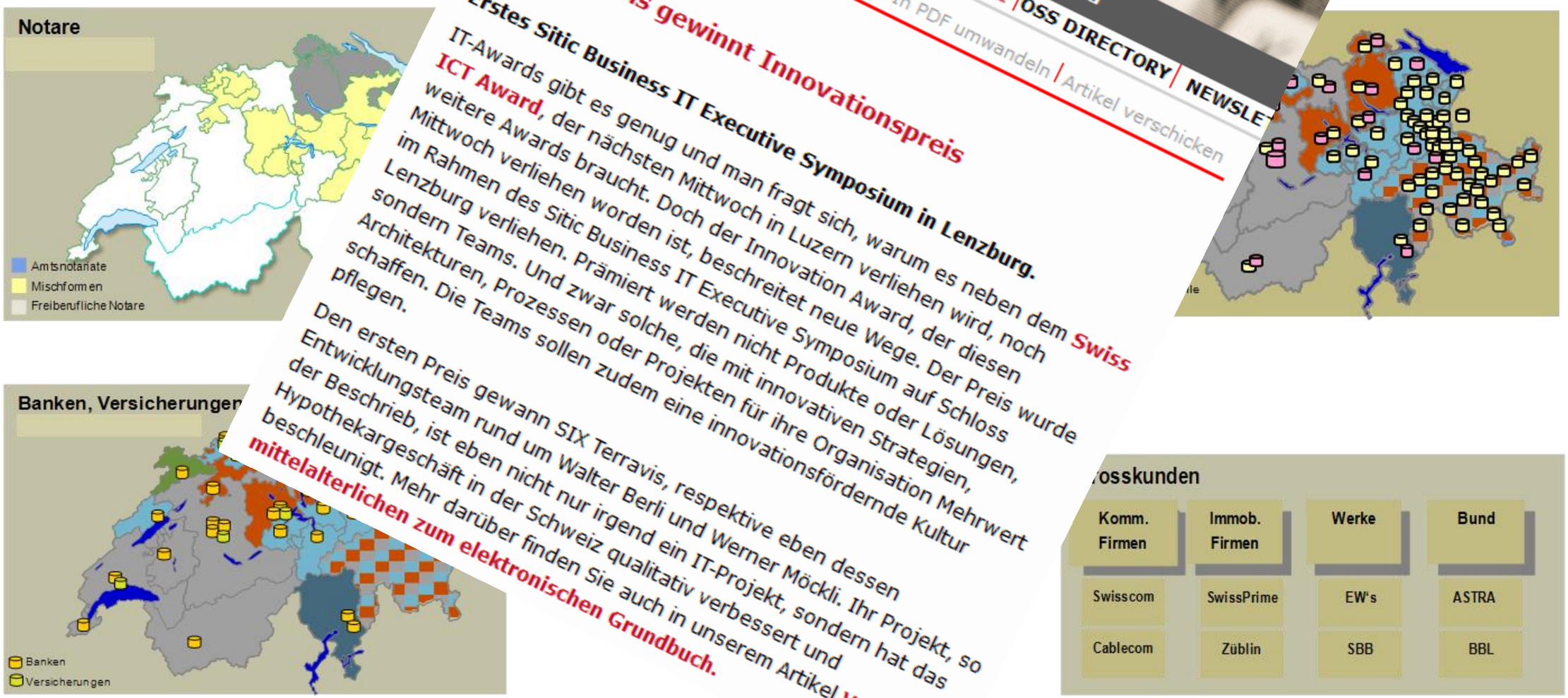
APACHE ODE

innoQ

BPMN

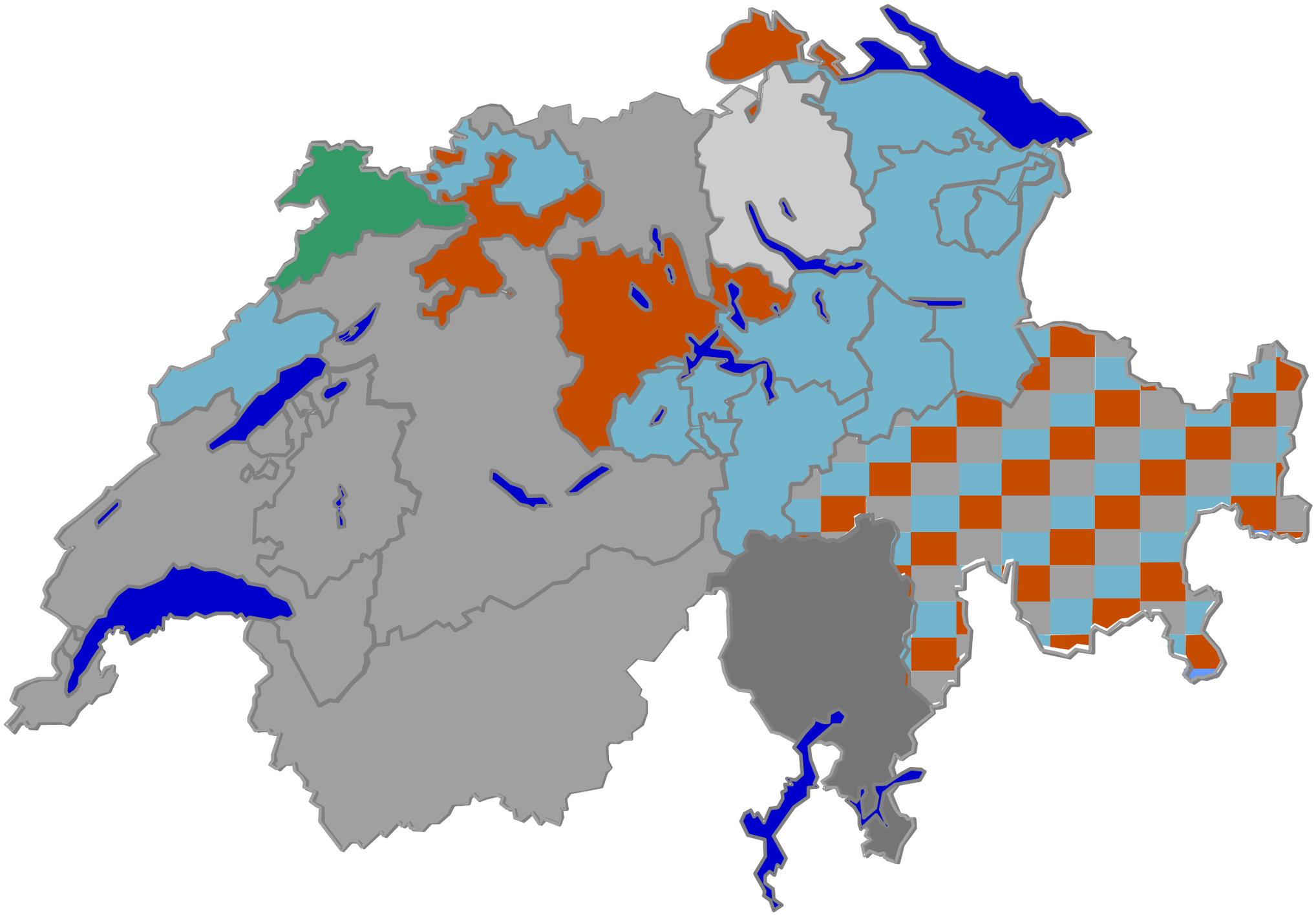
BPEL Unit

Context: Pr Hub Terravis



Challenges: Different notary systems

Challenges: Five different land registry systems



Starting point…

- 120+ pages textual description for three processes
  - People involved up to then were not used to define business processes
- Goal: Execute Processes on a BPMS

Given setting

- When developing executable business processes, process definitions become more complex than for highly abstract descriptive process descriptions
  - Computers cannot interpolate like humans
  - Steps need to be described in more details
  - Is true for both graphical models and textual descriptions
- Processes typically involve multiple parties
  - Different people, roles, departments, companies
  - In case of Terravis: Land Registries, Notaries, Banks, Register of Commerce, Pension Funds, Trustee, …

Resulting problems

- Large Models confuse (non-technical) people
- Large Models are generally harder to understand
- Modeling Notations might not be known to all stakeholders
- Text is often not precise
- Finding inconsistencies and contradictions in text is difficult
- For validating the whole process you need to bring all people together

Deriving process models

1. Try to understand text
2. Resolve conflicts, inconsistencies
3. Resolve missing information
4. Model the Process
5. Design necessary Services
6. Model the Executable Process

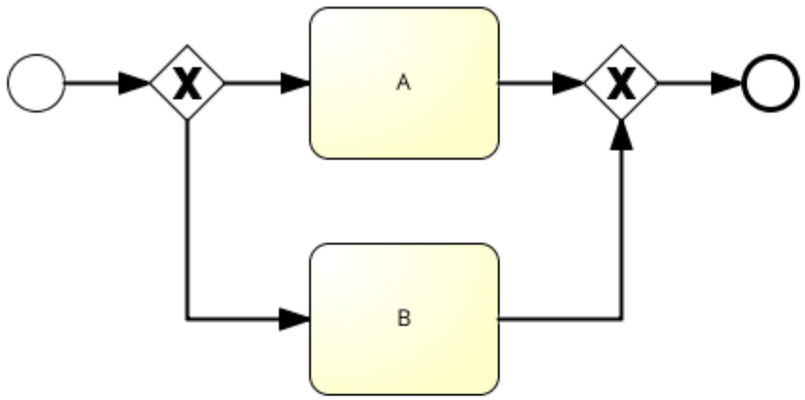
Modelling Scenarios

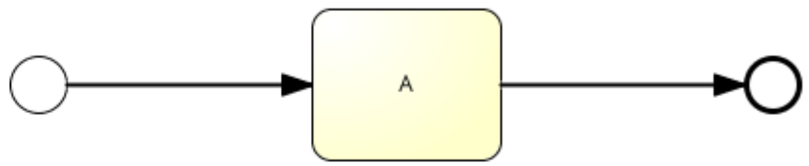
- A scenario is one deterministic path through a process
  - No choices to be made
  - People can think well in scenarios

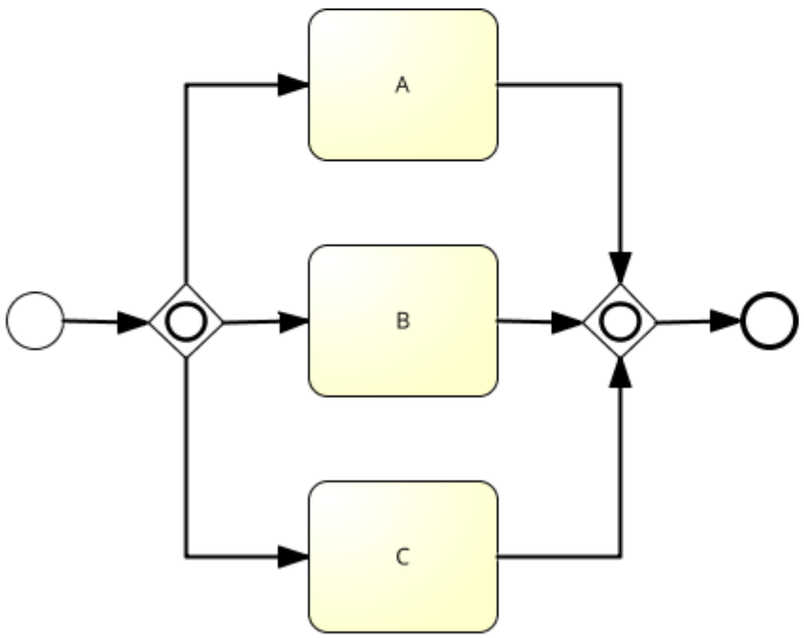
- Scenarios can be easily derived from the large process model

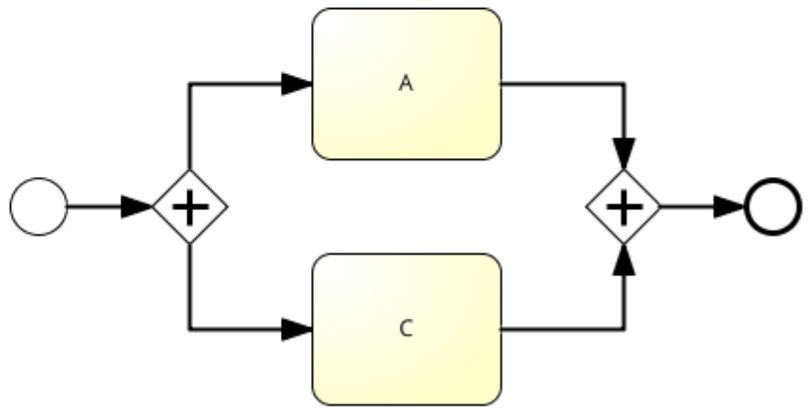
Rule 1: No choices!

- Only parallel gateways are allowed



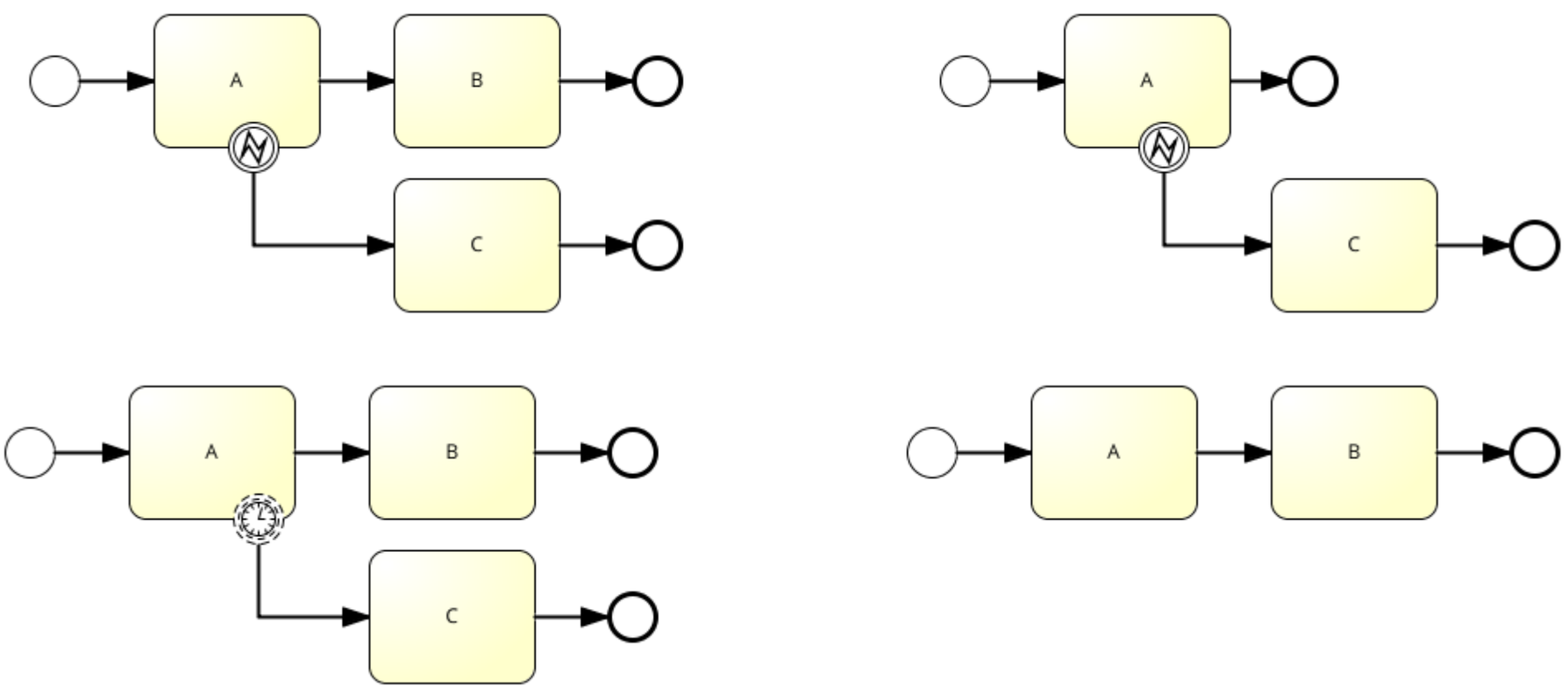






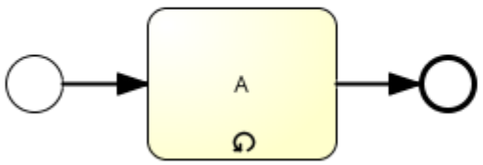
Rule 2: No unfired Boundary Events!

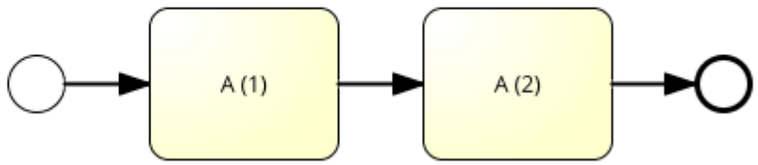
- If a boundary event is not fired, it is deleted
- If an interrupting boundary event is fired, all activities after the activity are deleted



Rule 3: Unrolled loops!

- Looped activities are duplicated as often as the scenario requires

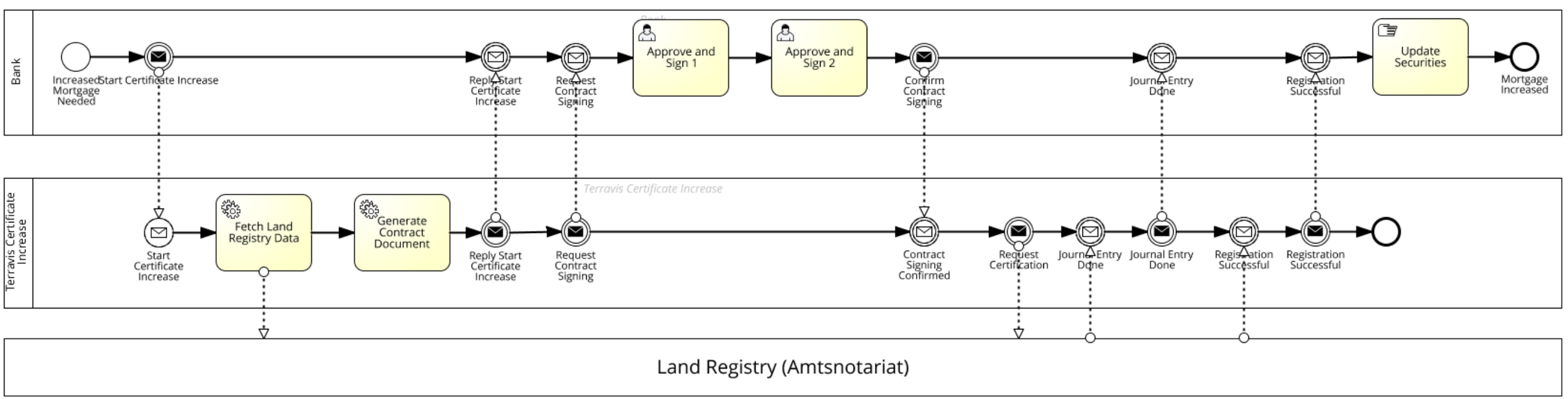




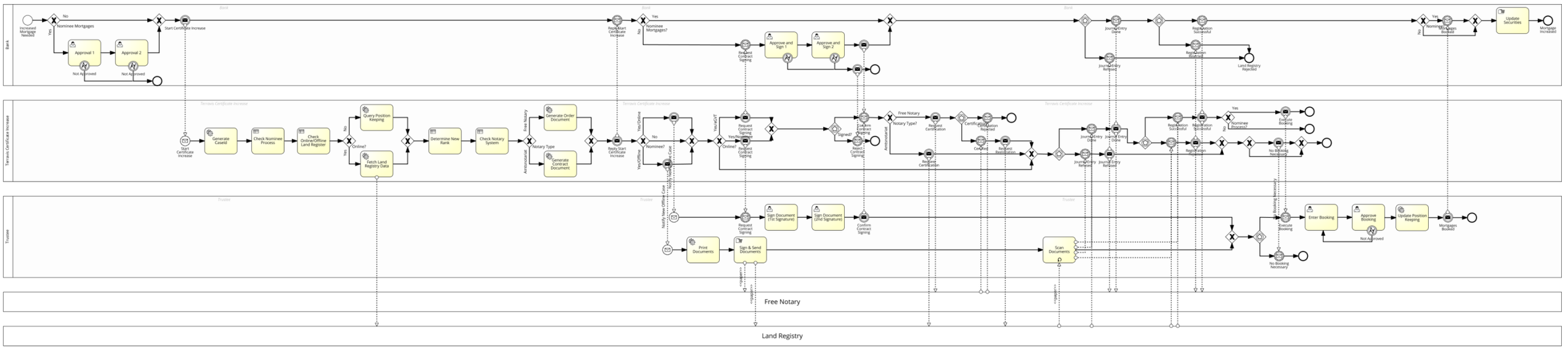
Rule 4: No unimportant sequential activities

- Unimportant sequential activities can be removed or replaced by one activity
  - To focus the discussion on important elements
  - To save space
  - To reduce number of elements

Scenario 1: Bank-owned mortgage – success



Derived Process Model



- We cannot show the real process here
- But that's pretty close
- “ Average Sized“ Process

Further validation

- Scenarios show Process Control Flow
  - What is done when by whom
- Often Data is important as well
  - Especially in collaborative scenarios
- Every Role is played by one Representative
  - “Simulation on Paper“
  - Write “Data“ on Paper and pass it around according to the Scenario Model
  - If documents are to be generated, you can also use drafts here
- Can every person complete his/her task?

- Later you can annotate the Example Data to Scenario Model Elements

What about QA?

Testing

- Saved this Project!
- Especially Unit Testing with BPELUnit
  - WS-* Integration Stack
- However, Test Cases are
  - time-consuming during creation
  - hard to maintain

Scenario-based Testing

- Test first! Test behavior, not code
- Scenarios with the elicitated example data can be used as test cases
  - Perhaps data need to be sanitized
  - e.g. Dummy Data replaced by real data (IDs, Names, ...)
- Scenarios discussed in Workshops are ideal as Acceptance Tests

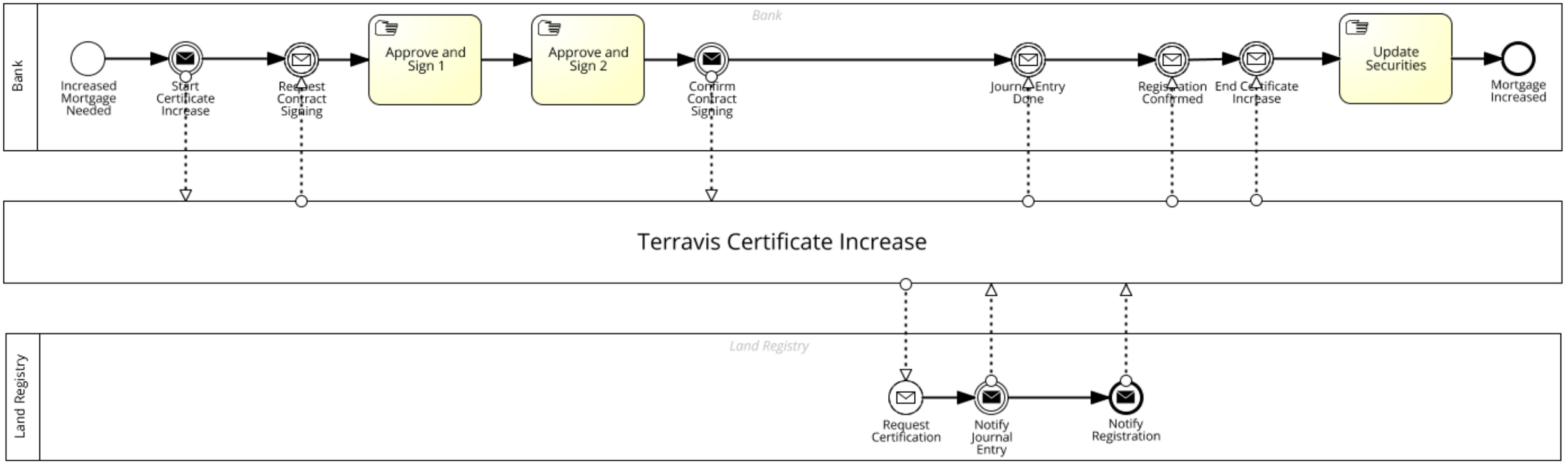
Sounds familiar?

Behavior-driven Design

- Uncle Bob: „Specification, not verification“
- BDD uses textual scenarios to describe acceptance criteria (Given/When/Then)
- Ubiquitous Language (see DDD)
- Tooling support
  - Reads and understands formal parts of the specification
  - Spec clauses are transformed into test parameters.
- Upfront Collaboration between different stakeholders

Test Case Modeling

- From Business Point of View



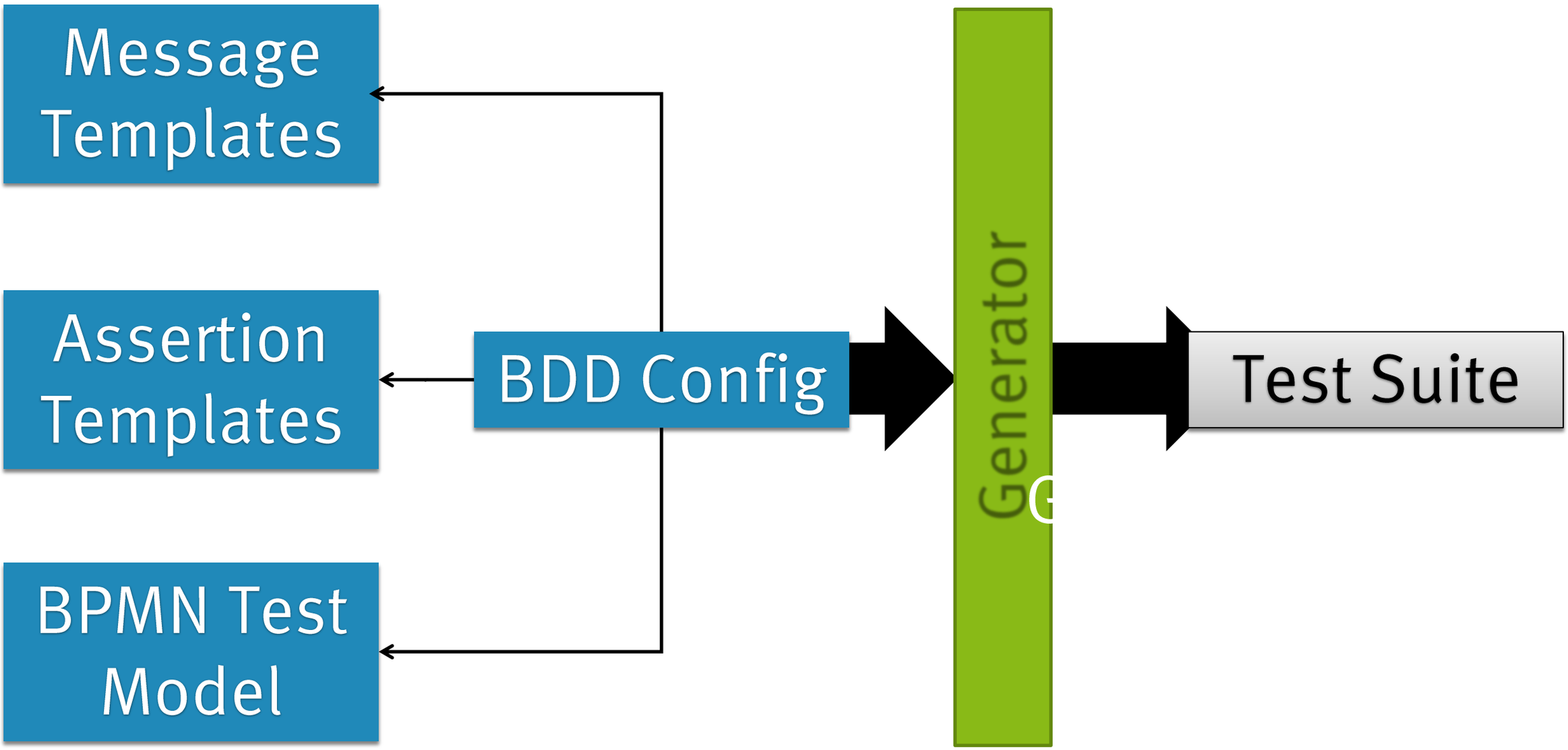
Test Data in Business Language

- Messages, e.g.
  - Order for an Increased Mortgage to the amount of CHF 1000000
- Assertions, e.g.
  - Order was successful
  - New Amount is CHF 1000000
- Attached directly to activites and messages

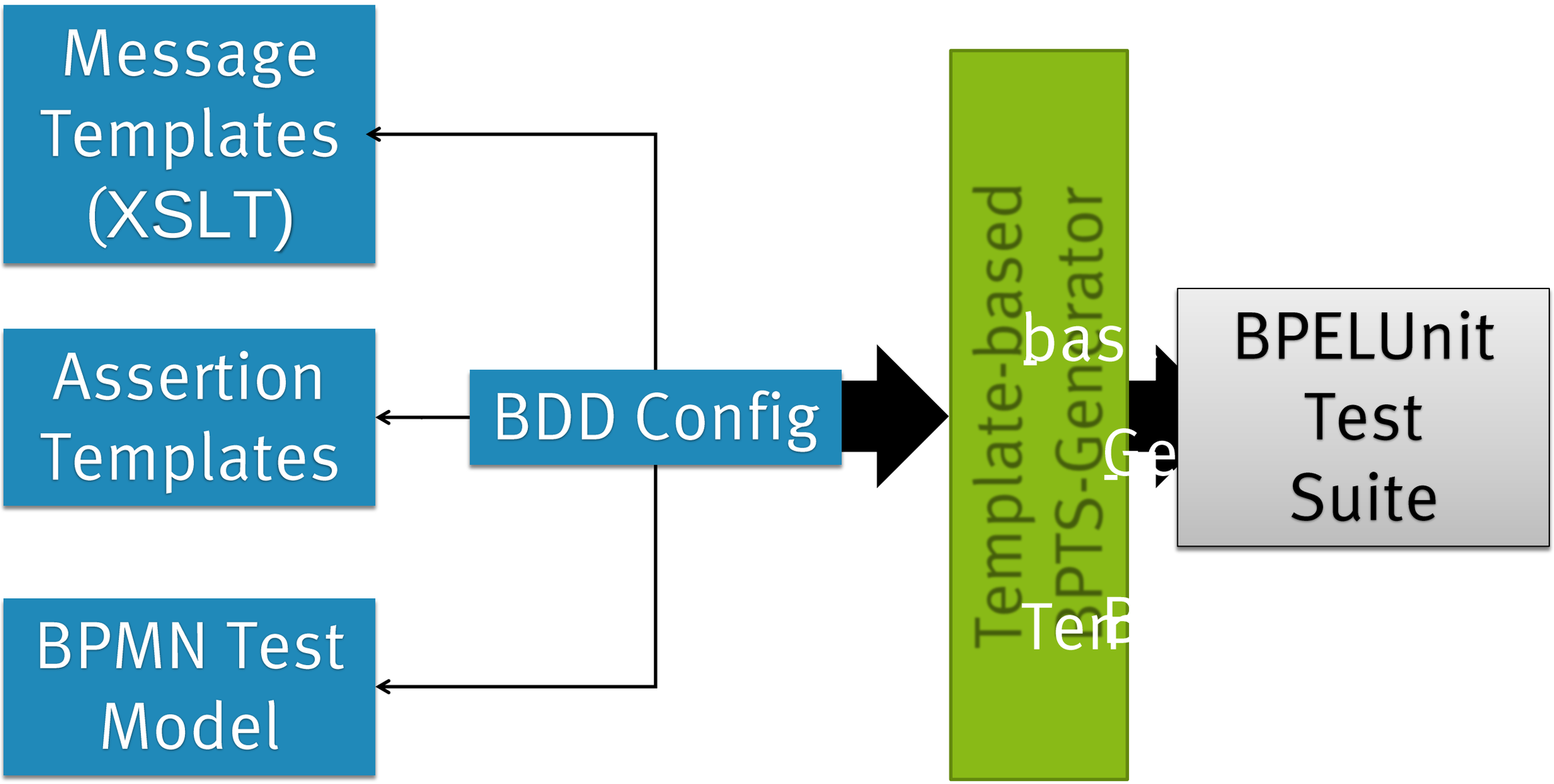
Test Data Extraction via Regexp

- Messages, e.g.
  - Order for an Increased Mortgage to the amount of CHF (.*)
- Assertions, e.g.
  - Order was (.*)
  - New Amount is CHF (.*)

BDD Generator Architecture



Our Architecture



Procedure

1. Model a Test Scenario
2. Annotate Activites and Messages with Data and Assertions
3. Define Mappings for Data, Assertions and Services
4. Generate Test Suite
5. Execute Test Suite against process implementation

Demo Time

Conclusions & Outlook

- Validation of non-trivial Process Models can be improved by using Scenarios
- Scenarios are Starting Point for BDD-style testing
- Easy Creation of Test Cases
- Ubiquitous Language and Templates make topic accessible to broader audience
- Better Maintainability

Thank you!

Dr. Daniel Lübke
daniel.luebke@innoq.com
@dluebke

Tammo van Lessen
tammo.van-lessen@innoq.com
@taval

http://www.innoq.com