Visualizing Use Case Sets as BPMN Processes

**Daniel Lübbe**, Kurt Schneider, Matthias Weidlich
Leibniz Universität Hannover, FG SE
Hasso Plattner Institute, Potsdam
Your next 10 Minutes…

1. The problem we are working on
2. How and why BPMN visualization helps
3. Stakeholders & Users
4. Example
5. Generation Algorithm
6. How it works
7. Visualization Example
8. Case Study
9. Pros and Cons
10. Next steps & Open questions
1. The problem we are working on

- Ordering, managing, and a huge number of Use Cases can be problematic
  - No explicit control-flow
  - No explicit ordering
  - No explicit dependencies

This has led to the introduction of extra models
- Like Use Case Charts

However, no automatic generation
- Requires manual, costly updates
- Models not suitable for other tasks

Use Case #9: Betreuer plant Endvortrag

Use Case #10: Betreuer plant Anfangsvortrag

Use Case #8: Betreuer plant Zwischenvortrag

Use Case #7: Betreuer ändert Thema

Use Case #11: Betreuer meldet Arbeit an

Use Case #12: Betreuer sucht Zweigtutachter

Use Case #13: Betreuer gibt Studenten Thema für Abschlußarbeit aus

Use Case #14: Betreuer trägt Zweigtutachter vor

Visualizing Use Case Sets as BPMN Processes
2. How and why BPMN visualization helps

- Use Cases can be ordered in a business process
  - Especially in service-oriented applications Use Cases have to be ordered along business processes
  - Therefore use a business process language to visualize the dependencies
  - Business Processes visualize global control-flow across Use Cases

- Which Use Cases form business processes
  Which do not?
  - Generation of business processes can solve this automatically

- BPMN is supposed to be the new standard
  - Hopefully understood by a large number of business analysts
3. Stakeholders & Users

- **Requirements Engineers**
  - Can spot “curious” spots
  - Can switch perspectives if interview partner is bored

- **Users & Other Stakeholders**
  - Can check whether order is right/valid or not

- **Business Analysts / Business Process Designers**
  - Can see whether Use Case model supports business process
  - Can use business process as a starting point for their modelling
4. Example

• Large project already seen:
  – University Application: Thesis Management
  – Spans functionality for starting, managing, supervising, and rating theses
  – 21 Use Cases
  – 6 Roles
  – Constitutes to 1 large business process and some supporting Use Cases

• Whole process unknown beforehand
  – No documented process
  – Tacit organizational knowledge
    • People know what to do
  – Overall Process had to be documented
5. Generation Algorithm

- Convert Use Case Control-Flow to small BPMN Processes

- Use Cases define their preconditions, postconditions and triggers
  - Matching those results in a global control-flow
6. How it works

Algorithm 1 Creation of a BPMN process for a single Use Case
1: P := new BPMNProcess();
2: StartEvent := P.add(new StartEvent(UC.PreConditions));
3: if UC.Triggers.Count > 1 then
4: ParallelGateway := P.add(new ParallelGateway());
5: LastElement := P.add(new ParallelGateway());
6: StartEvent.connectTo(ParallelGateway());
7: for all Trigger in UC.Triggers do
8: Event := P.add(new IntermediateEvent(Trigger));
9: ParallelGateway.connectTo(Event);
10: Event.connectTo(LastElement);
11: end for
12: else
13: LastElement := P.add(new IntermediateEvent (UC.Triggers[0]));
14: StartEvent.connectTo(LastElement);
15: end if
16: ConvertScenario(UC.MainScenario, LastElement);
17: for all Step in UC.Steps do
18: do something
19: end for
20: EndEvent := P.add(new EndEvent(UC.PostConditions));
21: LastElement.connectTo(EndEvent);

Algorithm 2 Conversion of Scenarios to BPMN
1: Function ConvertScenario(Scenario, LastElement):
2: for all Step in Scenario.Steps do
3: if Step.IsJumpTarget then
4: XORGateWay := P.add(new XORGateWay());
5: LastElement.connectTo(XORGateWay);
6: LastElement := XORGateWay;
7: end if
8: P.add(new Activity(Step));
9: if Step.isExtended then
10: XORGateWay := P.add(new XORGateWay());
11: LastElement.connectTo(XORGateWay);
12: LastElement := XORGateWay;
13: for all Extension in Step.Extensions do
14: ConvertScenario(Extension, LastElement);
15: end for
16: end if
17: if Scenario.JumpsBack then
18: LastElement.connectTo(GetXORGateWayFor(Scenario.JumpTarget));
19: end if
7. Visualization Example

**Visualization Example**

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Action</th>
<th>Primary Actor</th>
<th>Stakeholders</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>Student applies for Thesis</td>
<td>Student</td>
<td>Secretary (Academic Examination Office)</td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. **Stakeholders**: Secretary (Academic Examination Office) selects "Apply for Thesis".
2. **Student**: fills out application form and submits it.
3. **System**: shows confirmation.

**Extensions**: none

---

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Action</th>
<th>Primary Actor</th>
<th>Stakeholders</th>
</tr>
</thead>
<tbody>
<tr>
<td>#2</td>
<td>Academic Examination Office approves Thesis</td>
<td>Student</td>
<td>Secretary (Academic Examination Office)</td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. **Stakeholders**: Manager (Academic Examination Office) approves Thesis.
2. **Student**: receives approved Thesis.
3. **System**: shows confirmation.

**Extensions**: none

---

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Action</th>
<th>Primary Actor</th>
<th>Stakeholders</th>
</tr>
</thead>
<tbody>
<tr>
<td>#3</td>
<td>Student selects Topic</td>
<td>Student</td>
<td>Supervisor (Academic Examination Office)</td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. **Student**: chooses most interesting topic.
2. **Supervisor**: approves topic.
3. **System**: shows confirmation.

**Extensions**: none

---

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Action</th>
<th>Primary Actor</th>
<th>Stakeholders</th>
</tr>
</thead>
<tbody>
<tr>
<td>#4</td>
<td>Supervisor approves Topic</td>
<td>Supervisor</td>
<td>Student</td>
</tr>
</tbody>
</table>

**Main Success Scenario**

1. **Supervisor**: hands out Topic.
2. **System**: shows confirmation.

**Extensions**: (left out)
8. Case Study

• Business Process Visualization
  – Can spot Missing preconditions, postconditions and triggers
  – Can identify unknown parallelism
  – Can automatically partition Use Cases according to business processes
  – Can spot missing Use Cases
9. Pros and Cons

- Easy way to reclaim overview of large Use Case sets
- Easy to spot mistakes in conditions
- Easy to identify possible parallelism
- Easy to spot missing Use Cases in business process
- Business Processes can be (pre-)generated in SOA projects

- BPMN cannot directly express pre- and postconditions
- Use Cases cannot model complex control-flow structures
- Not possible (yet) to generate business objects
- Hard to automatically layout business process
10. Next steps & Open questions

- Non-literal matching of conditions & triggers
  - Case-based Reasoning
  - Non-literal search

- Modelling of business objects
  - Extraction from Use Cases

- Integration into Oryx Business Process Designer
  - Making it useful for Requirements Engineers

- Evaluation
Conclusions

• Visualization can support detection of
  – Wrong conditions
  – Missing Use Cases
  – Unknown parallelism

• In the example, it was helpful

• Further evaluation is needed

Thank you for your attention!
Any Questions?
Daniel Lübke <daniel.luebke@inf.uni-hannover.de>