Formal Verification of Business Process Configuration within a Cloud Environment

Presented by
Souha Boubaker

Telecom SudParis, UMR 5157 Samovar, Paris-Saclay University, France
ENIT, UR-OASIS, University of Tunis El Manar, Tunisia
Outline

- Research Context
- Towards Correct Business Processes Configuration
  - Motivation
  - EVENT-B method
  - Formalizing Configurable Process Models
  - Formalizing Configuration steps
  - Configuration Guidelines
  - Verification and Validation
- Towards Correct Cloud Resource Allocation in Business Processes
  - Motivation
  - Event-B Model
  - Modeling Control Flow
  - Modeling Cloud Resource Allocation
  - Verification and Validation
- Conclusion
Outline

■ **Research Context**

■ **Towards Correct Business Processes Configuration**
  • Motivation
  • EVENT-B method
  • Formalizing Configurable Process Models
  • Formalizing Configuration steps
  • Configuration Guidelines
  • Verification and Validation

■ **Towards Correct Cloud Resource Allocation in Business Processes**
  • Motivation
  • Event-B Model
  • Modeling Control Flow
  • Modeling Cloud Resource Allocation
  • Verification and Validation

■ **Conclusion**
Research Context

- Business process design

Diagram:
- What?
- How?
- By whom?
- Process Definition
- Tasks and Processes
- Resources and scheduling
- Resource Classification allocation rules
- Analyse
- Realisation
- Objectives
- Analyse
- Text
What are Configurable Process Models?

- Process Family: Different variants of the same process

- targeting customers’ demographics
- executed by different branches
What are Configurable Process Models?

- Example: Complaints Process Model

Configurable elements with green color
Problems Statement (1 / 2)

- Assumption: analysts derive process variants from a configurable process

- Observation: variant models often contain errors
  - Why?
  - How to avoid them?

The correctness of the process configuration is of paramount importance in order to avoid execution errors
“Cloud adoption is growing at greater than 25% CAGR (compound annual growth rate)”

Jane Munn, IBM
Resource Allocation in Business Processes

- **Activity 1**: Contact Organisation
- **Activity 2**: Fill complaints form
- **Activity 3**: Mail complaints

Cloud Resources
- **Shareable**
- **Elastic**

- ✓ **lower costs**, **more flexibility** and **greater scalability**
Problems Statement (2 / 2)

- Assumption: analysts assigns resources to process activities.

- Observation: inconsistencies in the Cloud resource allocation behavior may occur.
  - Why?
  - How to avoid them?

The correctness and the efficiency of the Cloud resource allocation is still required by the tenant
Outline

- Research Context
- Towards Correct Business Processes Configuration
  - Motivation
  - EVENT-B method
  - Formalizing Configurable Process Models
  - Formalizing Configuration steps
  - Configuration Guidelines
  - Verification and Validation
- Towards Correct Cloud Resource Allocation in Business Processes
  - Motivation
  - Event-B Model
  - Modeling Control Flow
  - Modeling Cloud Resource Allocation
  - Verification and Validation
- Conclusion
Motivation

Configuring a process model may be quite difficult and

- Analysts may easily make mistakes in selecting configuration choices
- A configuration step
- Lack of synchronization
- Deadlock

Configuration guidelines allow to be inline with domain constraints

An example of such rules is: “if a9=OFF then a14=OFF”
Objectives

- Guide the process analyst to easily configure process models while preserving correctness.
  - Analyze and check the correctness of a configurable process
  - Assist analyst in order to derive correct variants

- Respect specific domain constraints: Configuration guidelines introduced by Rosemann, M. et al.

✓ Perform an incremental formal verification by checking correctness and domain constraints at each intermediate step of the configuration procedure.
The EVENT-B method

- **Two Key features:**
  - *Stepwise refinement model:* represent systems at different abstraction levels;
  - *Proof-based model:* the use of mathematical proofs to verify consistency between refinement levels.

- **Two types of entities:**
  - Contexts: the static part
  - Machines: the dynamic part
Approach Overview

1. Modeling configurable process
   - Configurable process

2. MDE transformation
   - Is incorrect if violated invariant

3. Event-B Model specification
   - Correct & domain-compliant process variant

4. Proof-based verification
   - Machine M0: Variables and invariants for control flow specification are defined.
   - Machine M1: Configuration guidelines are introduced

5. Model checking verification
   - N configurations
     - Firing an appropriate event
     - Partially configured process
     - As input
     - As final output
     - Computing event guards
       - Correctness constraints of M0
       - Domain constraints of M1

- Machine M0 validation invariance: the basic control specifications are defined.
Formalizing Configurable Process Models

Machine M0:
- **Invariants:**
  - Correctness constraints
  - Configuration constraints
- **Events:** configuration steps
  - Activity configuration
  - Connector configuration: either a split or a join connector

Machine M1:
- **Invariants and events guards defining Configuration Guidelines**
Correctness Constraints

- **Structural Invariants**
  - Except the initial and the final nodes, each activity have exactly one incoming and one outgoing arc;
  - A split connector has:
    - exactly one incoming and
    - at least two outgoings arcs;
  - A join connector has:
    - at least two incomings arcs and
    - exactly one outgoing;
Correctness Constraints

- **Soundness Invariants**
  - All nodes of the process can be activated (i.e. every node can be reached by the initial activity);
  - For each activity in the process, there is at least one possible sequence leading from this activity to a final activity, i.e. the termination is always possible.
Correctness Constraints

- Behavioral Invariants
  - The configuration of a business process model may affect the soundness by two types of potential errors:
    - lack of synchronization : 3 invariants
    - Deadlocks : 3 invariants
  - These situations result from a mismatch between splits and joins.
Configuration Constraints

- Activity Configuration invariants
  - An invariant defining the model once an activity is removed: OFF activity configuration
  - An invariant defining the model after keeping an activity: ON activity configuration

- Connector Configuration invariants
  - Invariants defining the configuration constraints for each type of connector are defined according to the table:

<table>
<thead>
<tr>
<th>FROM-TO</th>
<th>OR</th>
<th>XOR</th>
<th>AND</th>
<th>seq</th>
</tr>
</thead>
<tbody>
<tr>
<td>OR</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>XOR</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AND</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
Formalizing Configuration Steps

- **Activity Configuration Events**
  - Two events:
    - *ConfigureACTON* event keeps an activity;
    - *ConfigureACTOFF* event excludes an activity.

```plaintext
ConfigureACTON(bp1, bp2, a11)
ConfigureACTOFF(bp2, bp3, a12)
```

Configurable activities a11 and a12
Formalizing Configuration Steps

- **Connector Configuration Events**
  - Two events for each connector:
    - One event for the split configuration: $\text{ConfigureORSplit}$, $\text{ConfigureXORSplit}$ and $\text{ConfigureANDSplit}$.
    - A second event for the join configuration: $\text{ConfigureORJoin}$, $\text{ConfigureXORJoin}$ and $\text{ConfigureANDJoin}$.
Formalizing Configuration Steps

Connector Configuration Events

- Each connector configuration event has to consider the following requirements:
  - The configuration constraints for each type of connector
  - Only configurable nodes can be removed to avoid unreachable ones;
  - The connectors types matching checking in order to prevent erroneous situations.

✓ we added for each event corresponding guards that should hold in order to apply a configuration step.
Formalizing Configuration Steps

- **Connector Configuration Events**
  - Example: the configuration of \textit{opj3} to AND could never be applied if \textit{ops5} has been already configured to XOR

\[
\text{ConfigureORSplit}(bp_1, bp_2, ops_5, \{a_9, a_{11}\}, \text{XOR}, \ldots)
\]

\[
\text{ConfigureORJoin}(bp_2, bp_3, opj_3, \{a_{10}, a_{12}\}, \text{AND}, \ldots)
\]
Injecting Configuration Guidelines in the Model

**Machine M1:**

- *Configuration guidelines* are introduced to depict relevant inter-dependencies between the configuration decisions in order to be inline with *domain constraints*.

- Such guidelines are expressed via logical expressions of the form *If-Then-rules*. 
Verification & Validation

- Verification using formal Proofs

  - Using the *Rodin tool*, our model generated 358 proof obligations (POs);

  - In order to demonstrate the model correctness, all generated proofs should be proved and discharged

  - Every defined invariant must be preserved and proved using these proofs

  - (272 POs ≈ 76%) were automatically discharged; and more complex ones (86 POs ≈ 24%) were interactively discharged
Verification & Validation

Interactive Proving Interface in Rodin
Verification & Validation

- Validation by animation using ProB

  - It allows the modification of the state of the model by *triggering the enabled events* that modify variables using constants.

  - It allows to play different scenarios and check the behavior of the Event-B model
The lack of synchronization situation is not possible.
Case Study

- How can our approach assist process analyst in applying correct configuration steps?

- Results:
  - our approach allows to:
    - save time and facilitate the identification of the configuration steps;
    - guarantee a correct process model at each configuration step;
    - derive domain-compliant process variants based on the configuration guidelines.
Outline

- **Research Context**
- **Towards Correct Business Processes Configuration**
  - Motivation
  - EVENT-B method
  - Formalizing Configurable Process Models
  - Formalizing Configuration steps
  - Configuration Guidelines
  - Verification and Validation
- **Towards Correct Cloud Resource Allocation in Business Processes**
  - Motivation
  - Event-B Model
  - Modeling Control Flow
  - Modeling Cloud Resource Allocation
  - Verification and Validation
- **Conclusion**
Motivation

Stores values/delay the same 
Properties and constraints
available as a capacity
covering activity need

Network1 was allocated by two 
activity instances of a19

Web form request

Find user profile (a2) → Recommend hotels (a3) → Select package (a6)

Common Shareable Elastic

Search hotel (a7) → Select hotel (a8)

New user (a5)

Common Shareable Elastic

Discount (a14)

Online payment (a15)

Confirmation (a17)

Request aborted (a19)

Compute3 should have the same 
Properties and constraints 
as well as a capacity covering activity need!
Resources Properties

A Cloud resource can be:

- **Elastic OR Non-elastic.**
  - A resource is *elastic* if we can change its capacity at runtime.
  - A resource is non-elastic if its capacity is fixed and cannot be modified at runtime.

- **Shareable OR Non-Shareable**
  - A resource is Shareable if it can be allocated by many activities' instances.
  - A resource is non-Shareable if it can be used by only one activity instance.

- **Shareable**
  - **Exclusive Shareable**
    - If its resource instances can be allocated by activities' instances but **not consumed at the same time**
  - **Common Shareable**
    - If its resource instances can be allocated and used by several activities' instances **at the same time**
Event-B Model

- Machine BPM0, the control flow perspective is modeled.
- Machine BPM1, the process execution instances are introduced.
- Machine BPM2, the allocated resources by a process activity are added and the shareability property of a cloud resource is pointed out.
- Machine BPM3, the resource perspective is refined by adding running resource instances.
- Machine BPM4: the elasticity property of a cloud resource is added.
First Level of Refinement: Introduces Execution Instances
- The sequencing between Business process execution events

An activity instance life cycle

- CreateBP(bp,acts,and_activDep, or_activDep)
- AddBpInst(bp,bpi)
  - bp_instances_type:=bp
  - AddACTInst(bp,ac,ai)
    - ACTInstances_type(ai):=a
    - ACTInstances_bpInstances(ai):=bpi
    - ACTInstances_State(ai):=initiated
- CancelACTInst(ai)
- RunACTInst(ai)
  - ACTInstances_State(ai):=running
- FailACTInst(ai)
- CompleteACTInst(ai)
Second Level of Refinement: Introduces Resource Perspective

- The allocation dependency: denotes for each process the relation of a possible allocation between a resource and an activity. (pattern Direct Allocation (WRP-01) defined by N. Russel et al.).
  - A relation AllocationDep.

- The substitution dependency: captures the possibility to replace a resource by another to perform some work in case of its unavailability or absence.
  - A relation SubstitutionDep
Second Level of Refinement: Adds the shareability property

- *Shareability Constraints*
  - a resource may be shareable in a given process and non-shareable in another.
  - only shareable resources may have several allocation dependencies;
  - Two shareability properties: *Exclusive shareable* and *common shareable resource*. 
Modeling Cloud Resource Allocation

- **Third Level of Refinement: Adds the resource instances**

Exclusive **shareable** resource instances can be allocated and used by different activities' instances but not at the same time,
Modeling Cloud Resource Allocation

Fourth Level of Refinement: Models Cloud Resource Elasticity

- Support the pattern **Capability-based Allocation (WRP-08)** defined by N. Russel et al.

- The allocation is based on the matching of specific activities requirements with the capabilities of resources.

```
Inv6: \forall ai, ri.(ai \in ACT_Instances \land ri \in RES_Instances \land ACT_Instances_State(ai) = completed \land ri \mapsto ai \in Consumed \land ri \mapsto ai \in \text{dom}(ACTInstance\_RES_\text{Needs}) \Rightarrow ACTInstance\_RES_\text{Needs}(ri \mapsto ai) \leq RESInstance\_Capacity(ri))
```

```
Inv5: \forall bp, r.(r \in RES \land bp \in BP \land bp \mapsto r \in \text{dom}(AllocationDep) \land Elastic(bp \mapsto r) = FALSE \Rightarrow \text{SUM}(\{(bp \mapsto r) \times BP\_activities[{bp}]\} < ACT\_RES_\text{Needs}) \leq RES\_Capacity(r))
```
Fourth Level of Refinement: Models Cloud Resource Elasticity

- **Elasticity Events:**
  - \textit{ResizeUpRESInst} increases the capacity of a resource instance according to the activities instances needs.
  - \textit{ResizeDownRESInst} decreases the capacity of a resource instance in case it is unnecessary to the activity instance.
Verification & Validation

Verification using formal Proofs

- Each invariant should be established by the initialisation and preserved by each event.

- Using the *Rodin tool*, our model generated 338 proof obligations (POs);
  - (257 POs ≈ 76%) were automatically discharged; and more complex ones (81 POs ≈ 24%) were interactively discharged.
Verification & Validation

Interactive Proving Interface in Rodin
Verification & Validation

- Validation by animation using ProB
  
  - It allows to check the correctness/validity of the model by playing different scenarios;
  
  - At each moment, it is possible to know which event are enabled or not
Verification of the development

- Verification using ProB
Signavio Extension (proof of concept)
Outline

■ Research Context
■ Towards Correct Business Processes Configuration
  • Motivation
  • EVENT-B method
  • Formalizing Configurable Process Models
  • Formalizing Configuration steps
  • Configuration Guidelines
  • Verification and Validation
■ Towards Correct Cloud Resource Allocation in Business Processes
  • Motivation
  • Event-B Model
  • Modeling Control Flow
  • Modeling Cloud Resource Allocation
  • Verification and Validation
■ Conclusion
Conclusion

- **A formal verification model** to
  - Analyze and check the correctness of a configurable process model
  - Ensure correct derived variants with respect to configuration guidelines

- **A formal verification model** for resource allocation in business process while considering:
  - Cloud resources properties
  - Different relationships between activities and resources.

- Integration of Cloud Resource description in Signavio Editor
Current work

- An approach for process configuration based on a reduced SOG (Symbolic observation graph) that groups the behavior of all correct configurations

- The set of correct configurations combinations is extracted and supplied to the analyst at design time
Future work

- An approach for process fragments consolidation and merging while considering correctness constraints

- An approach for process resources QoS management and verification
References

- Formal approach for verifying the **correctness and domain compliance** of a configurable process model and its derived variants.
  
  The work was published in two conferences proceedings:
  
  

- Formal approach for ensuring a **correct and consistent** Cloud resource allocation in business process modeling.
  
  The work was published in two conferences proceedings and a peer reviewed journal:
  
  
  
Thank you for your attention

Souha Boubaker
E-mail: souha.boubaker@telecom-sudparis.eu