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

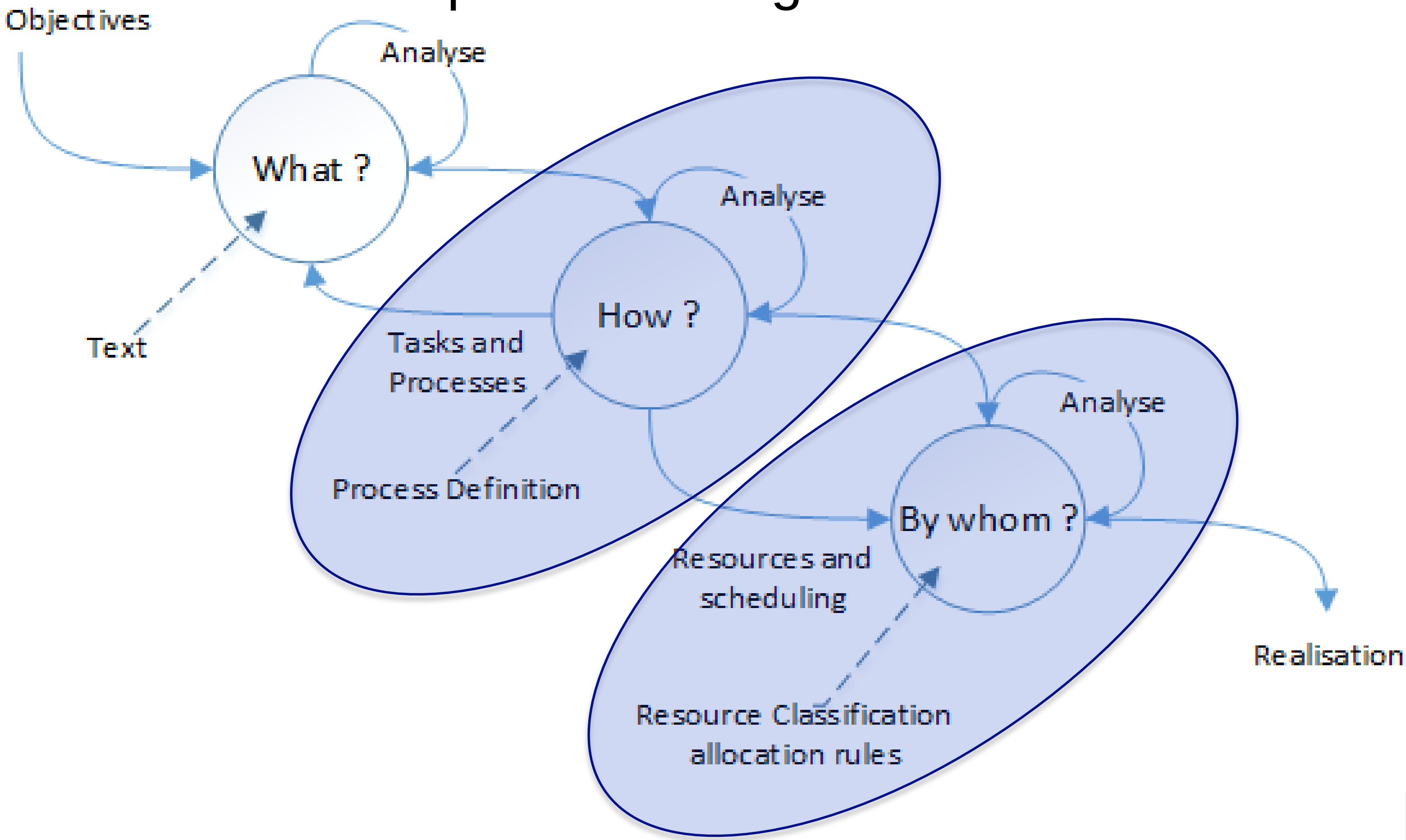
- **Research Context**
- **Towards Correct Business Processes Configuration**
  - Motivation
  - EVENT-B method
  - Formalizing Configurable Process Models
  - Formalizing Configuration steps
  - Configuration Guidelines
  - Verification and Validation
- **Towards Correct Cloud Resource Allocation in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

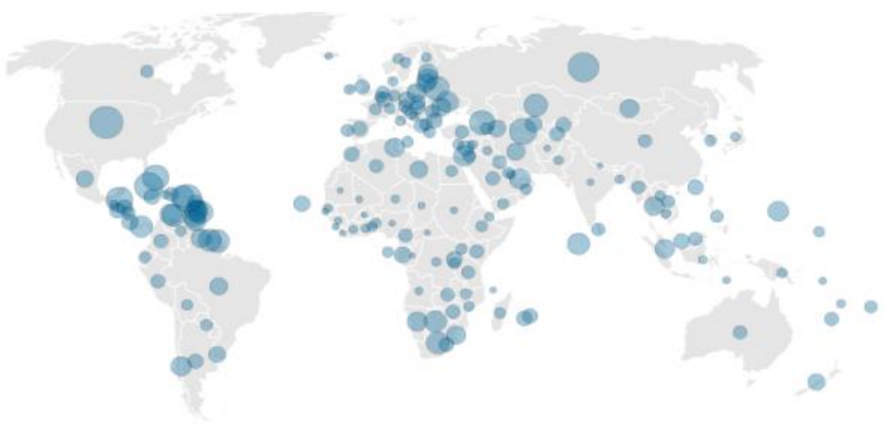


What are Configurable Process Models?

- Process Family: Different variants of the same process

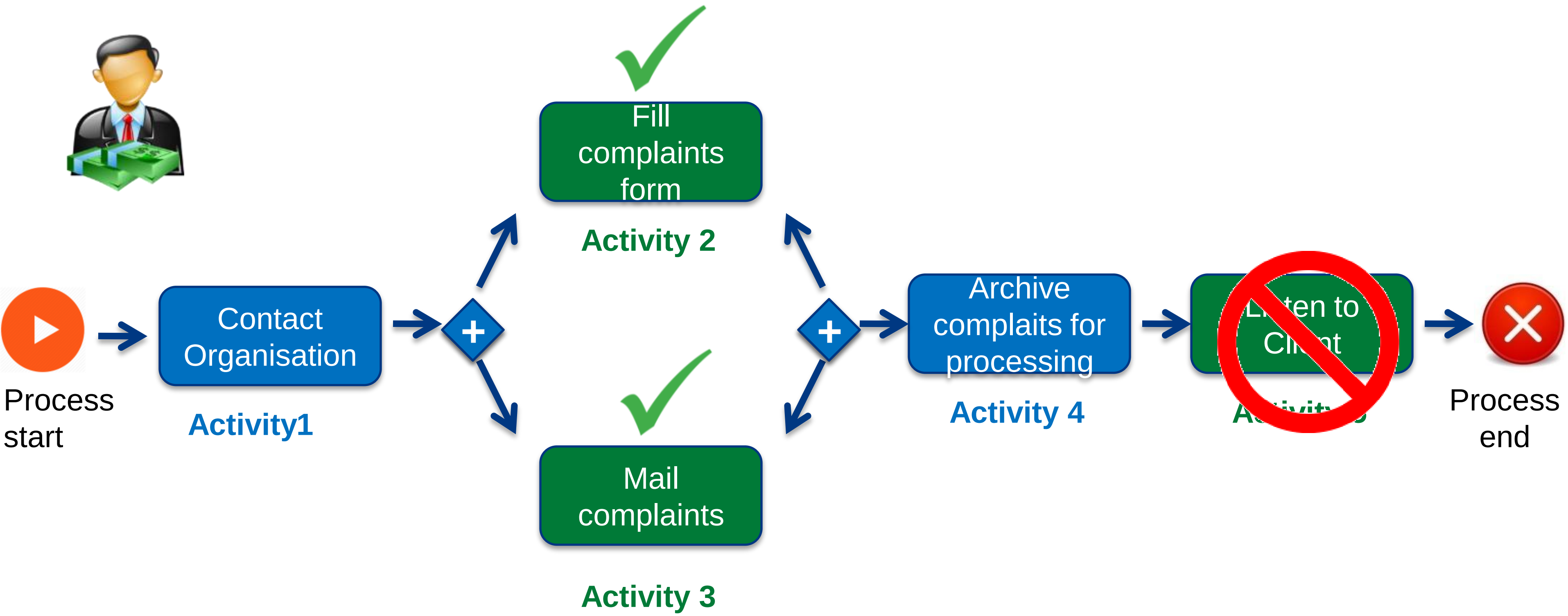
targeting customers' demographics

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**

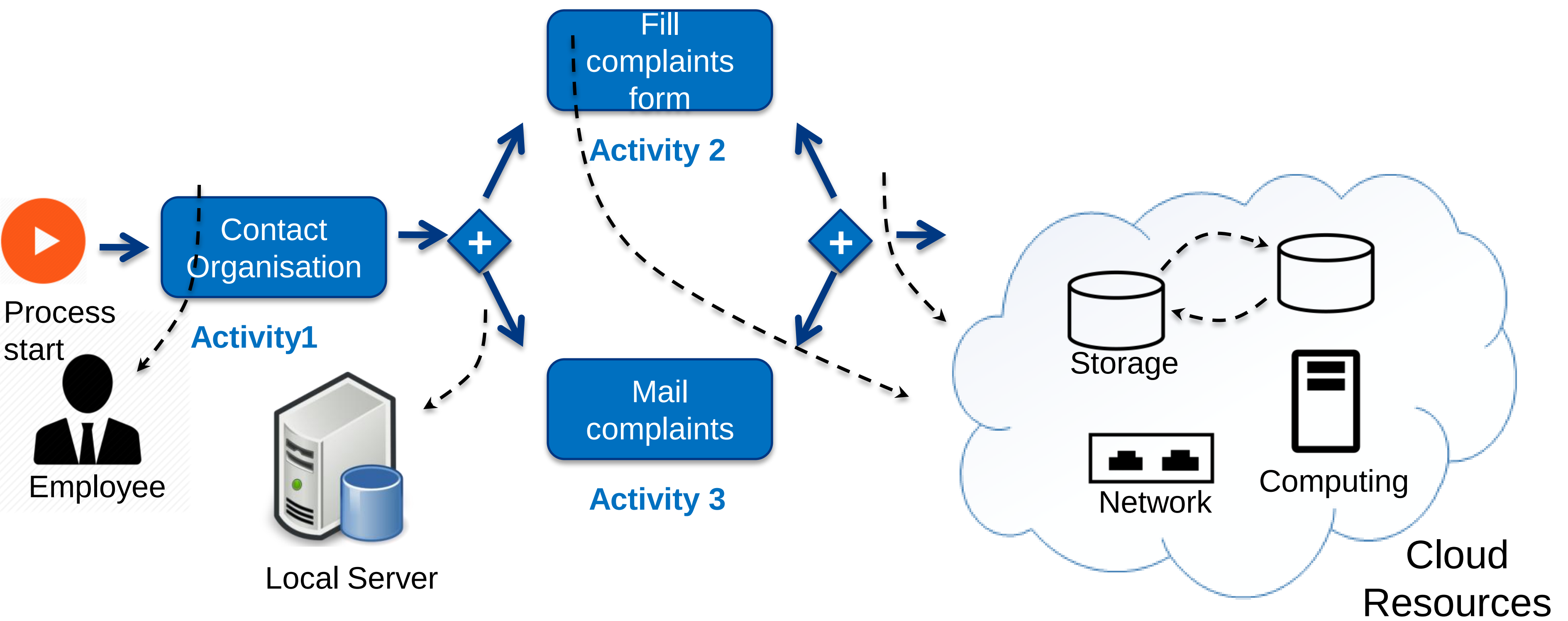
- **Observation: variant models often contain errors**
  - Why?
  - How to avoid them?

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



- ➢ ***Shareable***
- ➢ ***Elastic***

✓ ***lower costs, more flexibility*** and ***greater scalability***

Problems Statement (2 / 2)

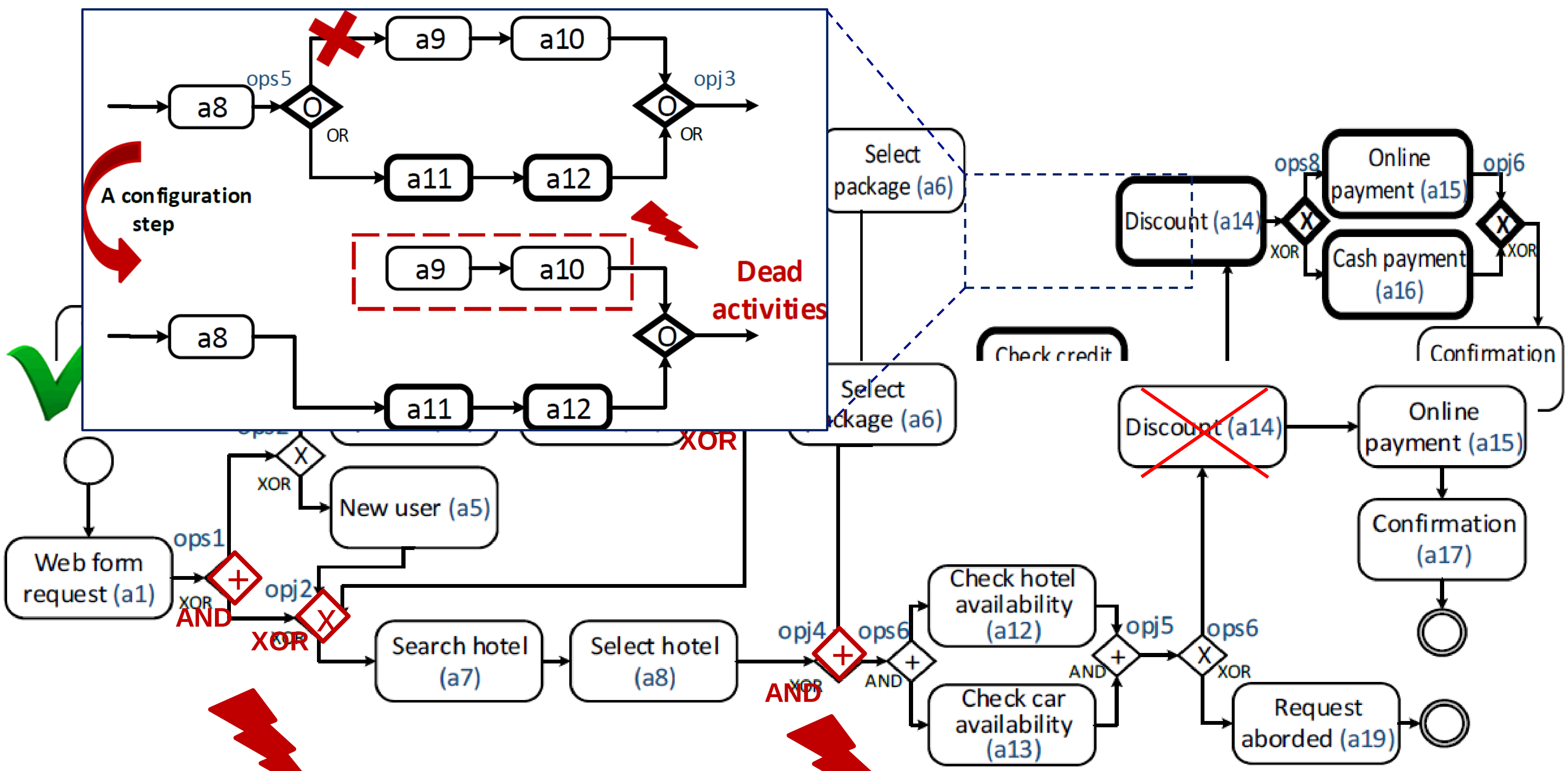
- **Assumption: analysts assigns resources to process activities.**
- **Observation: inconsistencies in the Cloud resource allocation behavior may occur.**
  - Why?
  - How to avoid them?

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



- Configuration guidelines allow to be in line with domain constraints
- Configuring a process model is a quite difficult task
- An example of such rules is: "if a9=OFF then a14=OFF"
- Analysts may easily be mistaken in selecting configuration choices

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.

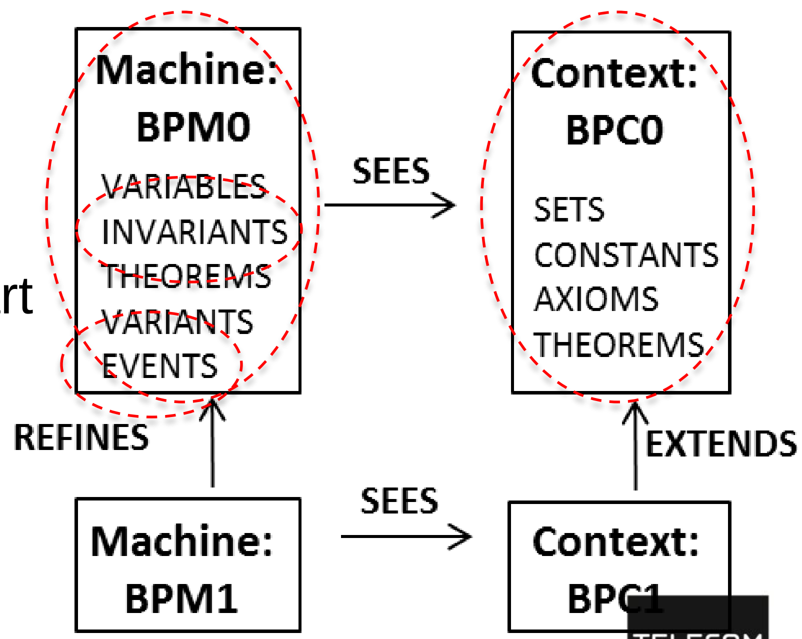
The EVENT-B method

Two Key features:

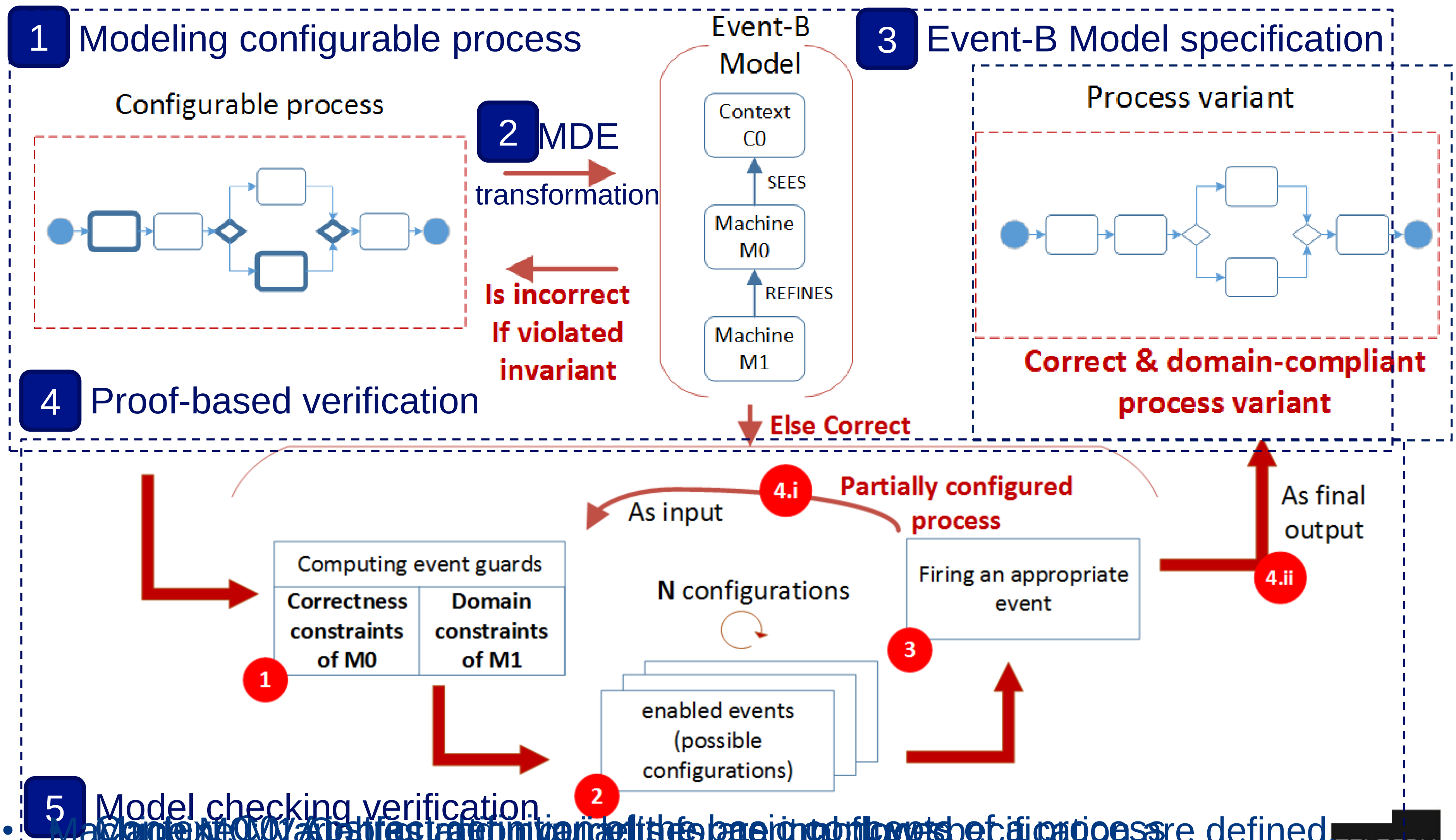
- *Stepwise refinement model:* represent systems at different abstraction levels;
- *Proof-based model:* the use of mathematical proofs to verify consistency between refinement levels.

Two types of entities :

- Contexts: the static part
- Machines: the dynamic part



Approach Overview



Formalizing Configurable Process Models

- **Machine M0 :**
  - *Invariants*:
    - Correctness constraints
      - Structural
      - Soundness
      - Behavioral
    - Configuration constraints
      - Activity
      - Connector
  - *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

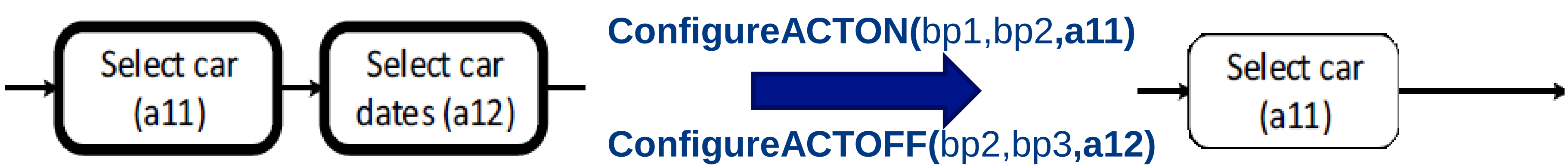
- **Connector Configuration invariants**
  - Invariants defining the configuration constraints for each type of connector are defined according to the table:

FROM-TO	OR	XOR	AND	seq
OR	√	√	√	√
XOR		√		√
AND			√	

Formalizing Configuration Steps

Activity Configuration Events

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

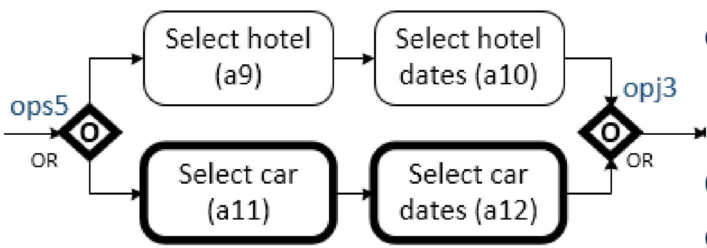


Configurable activities a11 and a12

Formalizing Configuration Steps

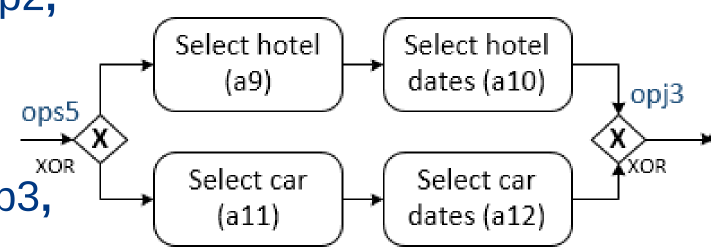
Connector Configuration Events

- Two events for each connector :
  - One event for the split configuration: *ConfigureORSplit*, *ConfigureXORSplit* and *ConfigureANDSplit*.
  - A second event for the join configuration: *ConfigureORJoin*, *ConfigureXORJoin* and *ConfigureANDJoin*.



ConfigureORSplit(bp1,bp2**, ops5,{a9,a11},XOR,..)**

ConfigureORJoin(bp2,bp3**, opj3,{a10,a12},XOR,..)**



Formalizing Configuration Steps

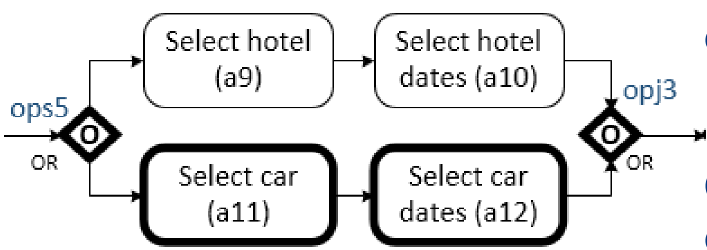
- **Connector Configuration Events**
  - Each connector configuration event has to consider the following requirements:
    - The configuration constraints for each type of connector
    - Only configurable nodes can be removed to avoid unreachable ones;
    - The connectors types matching checking in 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

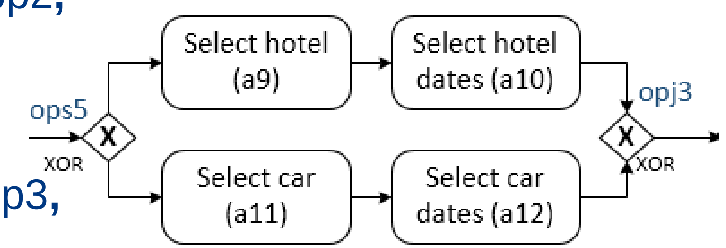
Connector Configuration Events

- Example: the configuration of ***opj3*** to AND could never be applied if ***ops5*** has been already configured to XOR



ConfigureORSplit(bp1,bp2**, ops5,{a9,a11},XOR,..)**

ConfigureORJoin(bp2,bp3**, opj3,{a10,a12}**, ~~**AND**~~ **.)**



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*.

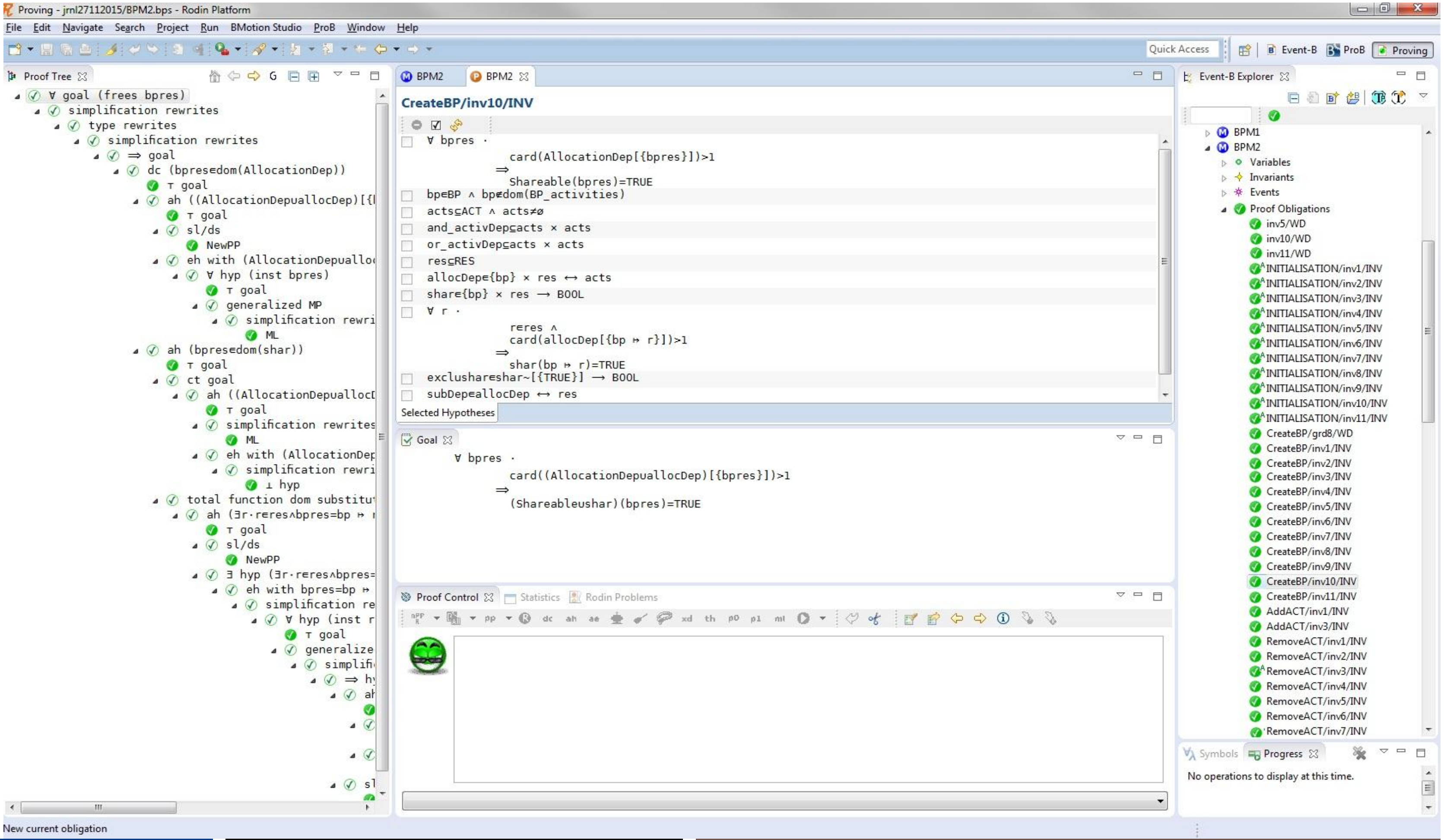
Verification & Validation

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**



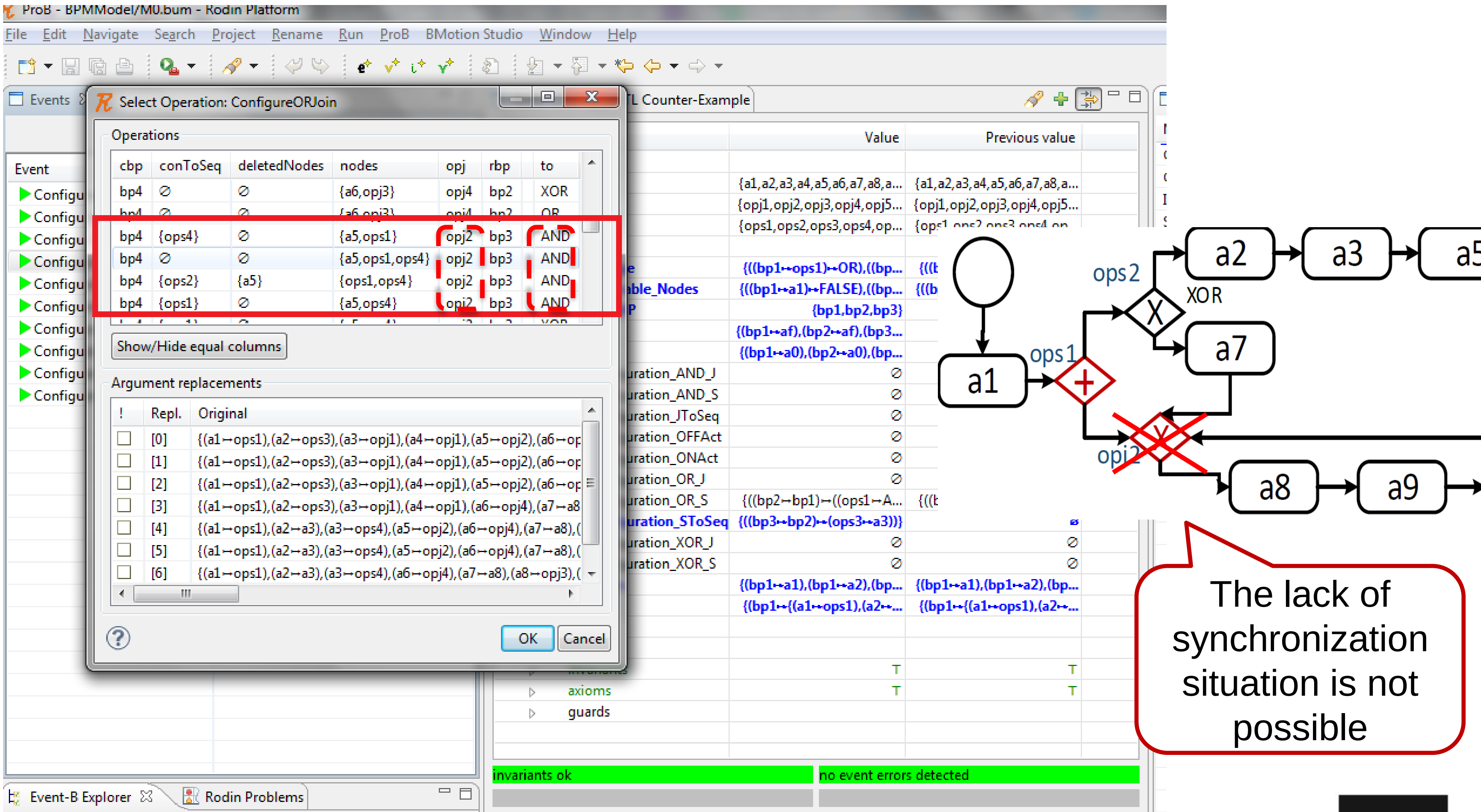
Verification & Validation

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

Verification & Validation



Case Study

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

- **Results:**

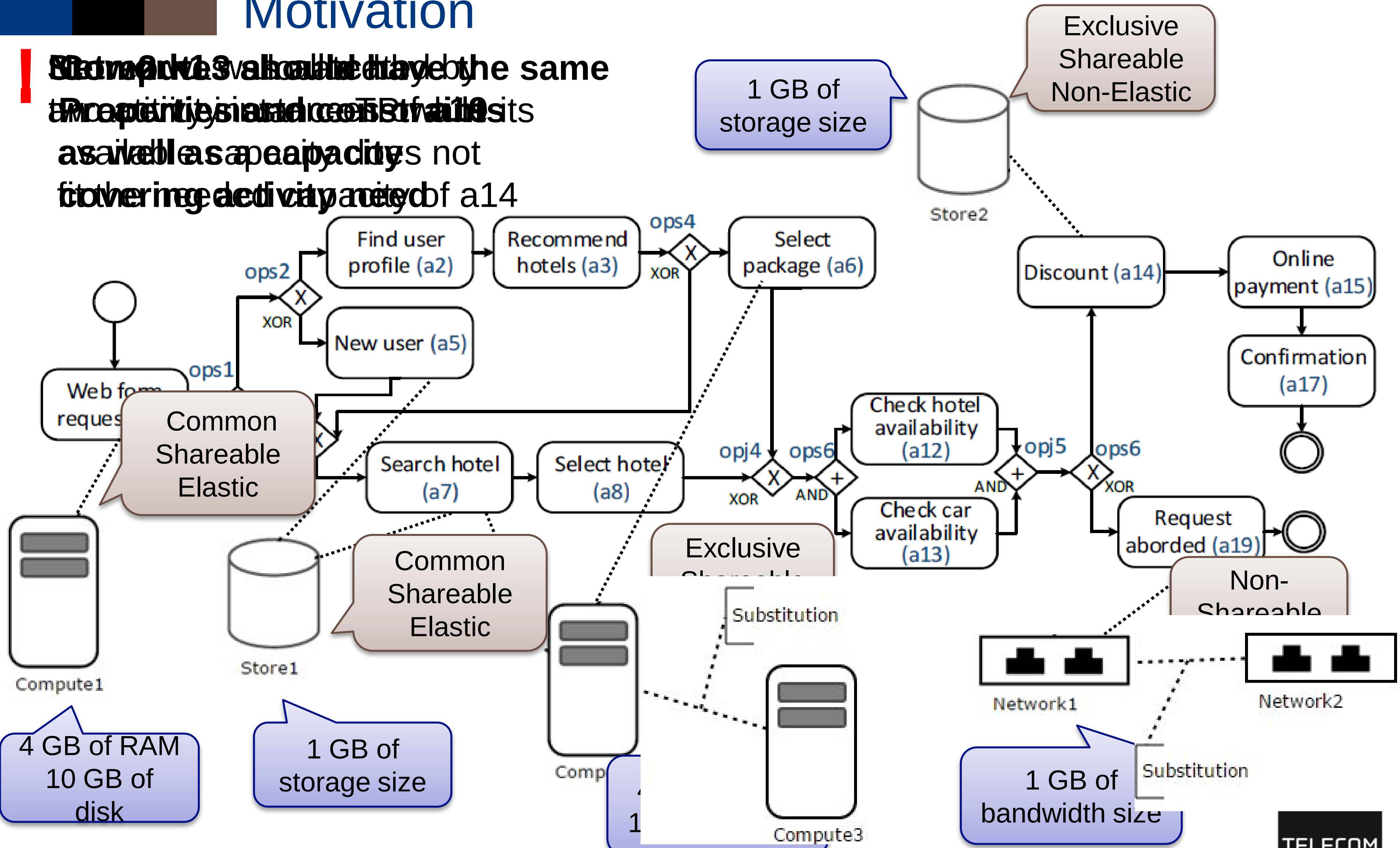
  our approach allows to:

  - 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


Exclusive Shareable Non-Elastic
1 GB of storage size
Store2
ops4
Find user profile (a2)
Recommend hotels (a3)
XOR
Select package (a6)
Discount (a14)
Online payment (a15)
ops2
XOR
New user (a5)
Confirmation (a17)
ops1
Web form request
Common Shareable Elastic
Search hotel (a7)
Select hotel (a8)
opj4
ops6
XOR
AND
Check hotel availability (a12)
Check car availability (a13)
opj5
AND
ops6
XOR
Request aborded (a19)
Exclusive
Non-
Common Shareable Elastic
Compute1
Store1
Substitution
Compute3
Network1
Network2
Substitution
4 GB of RAM 10 GB of disk
1 GB of storage size
1 GB of bandwidth size

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

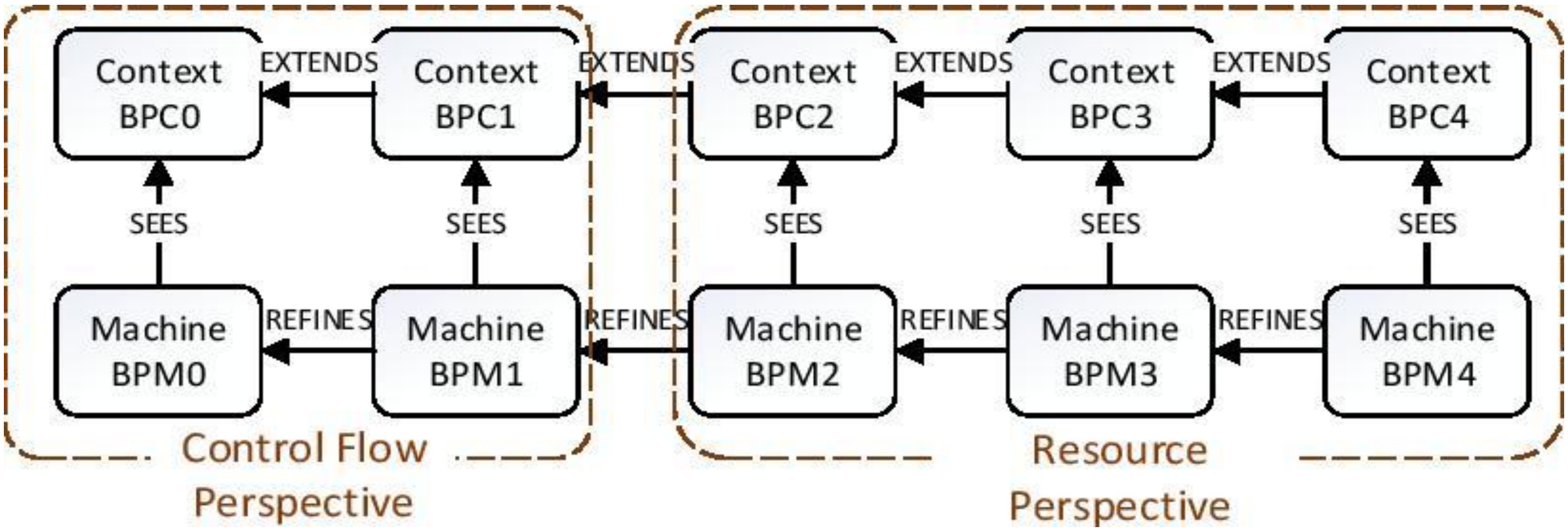
Exclusive Shareable

- 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

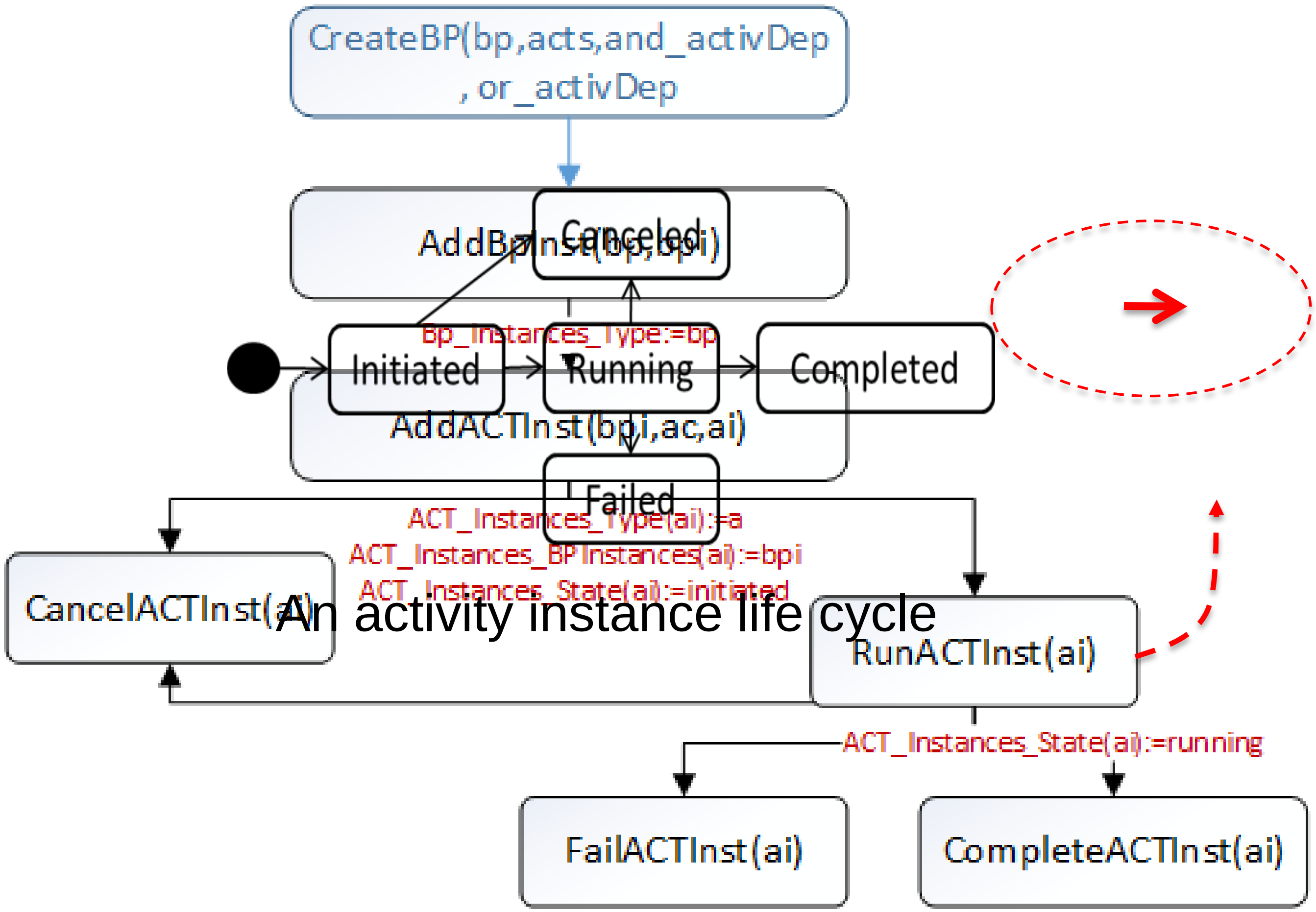


- Machine BPM0, the control flow perspective is modeled.
- Machine BPM1, the process execution instances are introduced.
- Machine BPM2, the allocated resources by a process activity are added and the shareability property of a cloud resource is pointed out.
- Machine BPM3, the resource perspective is refined by adding running resource instances.
- Machine BPM4: the elasticity property of a cloud resource is added.

TELECOM SudParis

Modeling Control Flow

- **First Level of Refinement: Introduces Execution Instances**
  - The sequencing between Business process execution events



An activity instance life cycle

Modeling Cloud Resource Allocation

- **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.

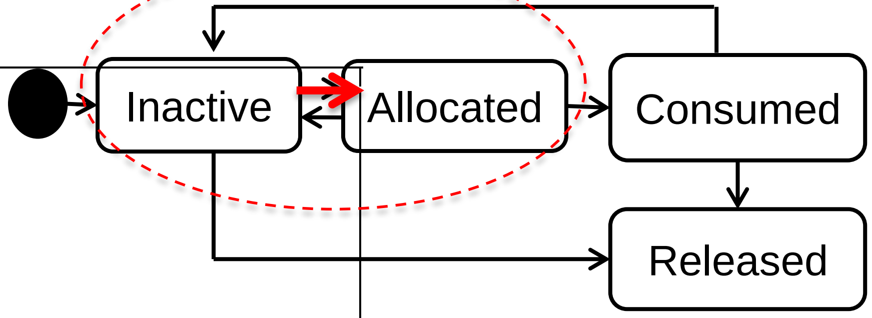
    - A relation *SubstitutionDep*

Modeling Cloud Resource Allocation

- **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**



AllocateRESInst $\triangleq$
ANY ai ri
WHERE
grd1: $ai \in ACT\_Instances$
grd2: $ACT\_Instances\_State(ai) = initiated$
grd3: $ri \in RES\_Instances \wedge ri \mapsto ai \notin Allocated$
grd4: $BP\_Instances\_Type(ACT\_Instances\_BP\_Instances(ai)) \mapsto RES\_Instances\_Type(ri) \in AllocationDep - 1[\{ACT\_Instances\_Type(ai)\}]$
grd5: $ri \in Inactive \vee BP\_Instances\_Type(ACT\_Instances\_BP\_Instances(ai)) \mapsto RES\_Instances\_Type(ri) \in ExclusiveShareable - 1[\{FALSE\}]$
grd6 $RES\_Instances\_BP\_Instances(ri) = ACT\_Instances\_BP\_Instances(ai)$
THEN
act1: $Inactive := Inactive \setminus \{ri\}$
act2: $Allocated := Allocated \cup \{ri \mapsto ai\}$

***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.

$$\texttt{Inv6:}\ \forall ai, ri.(ai \in ACT\_Instances \wedge ri \in RES\_Instances \wedge ACT\_Instances\_State(ai) = completed \wedge ri \mapsto ai \in Consumed \wedge ri \mapsto ai \in dom(ACTInstance\_RES\_Needs) \Rightarrow ACTInstance\_RES\_Needs(ri \mapsto ai) \leqslant RESInstance\_Capacity(ri))$$

$$\texttt{Inv5:}\ \forall bp, r.(r \in RES \wedge bp \in BP \wedge bp \mapsto r \in dom(AllocationDep) \wedge Elastic(bp \mapsto r) = FALSE \Rightarrow SUM((\{bp \mapsto r\} \times BP\_activities[\{bp\}]) \lhd ACT\_RES\_Needs) \leqslant RES\_Capacity(r))$$

Modeling Cloud Resource Allocation

- **Fourth Level of Refinement: Models Cloud Resource Elasticity**
  - *Elasticity Events:*
    - *ResizeUpRESInst* increases the capacity of a resource instance according to the activities instances needs.
    - *ResizeDownRESInst* decreases the capacity of a resource instance in case it is unnecessary to the activity instance.

Verification & Validation

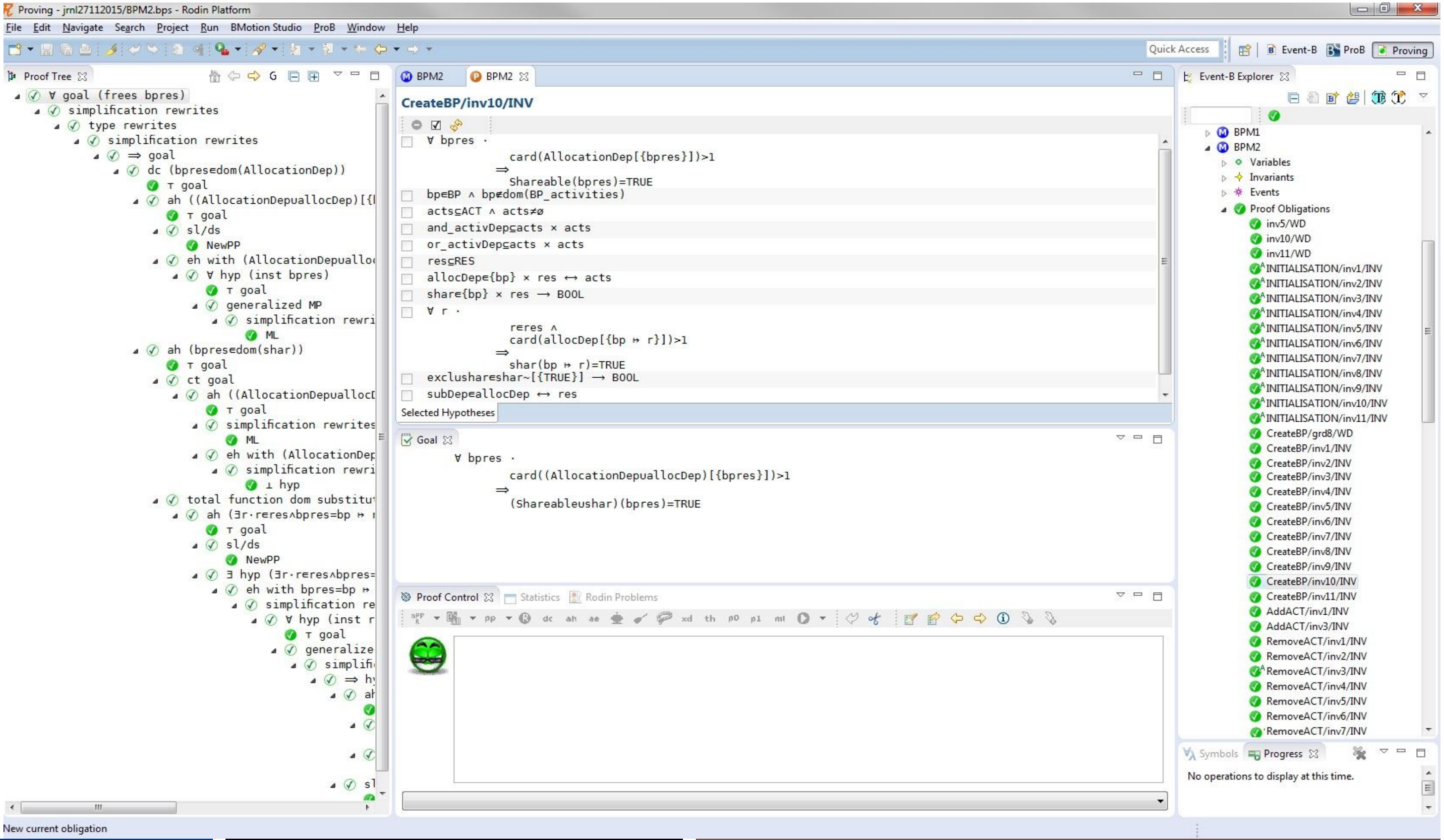
Verification using formal Proofs

- 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

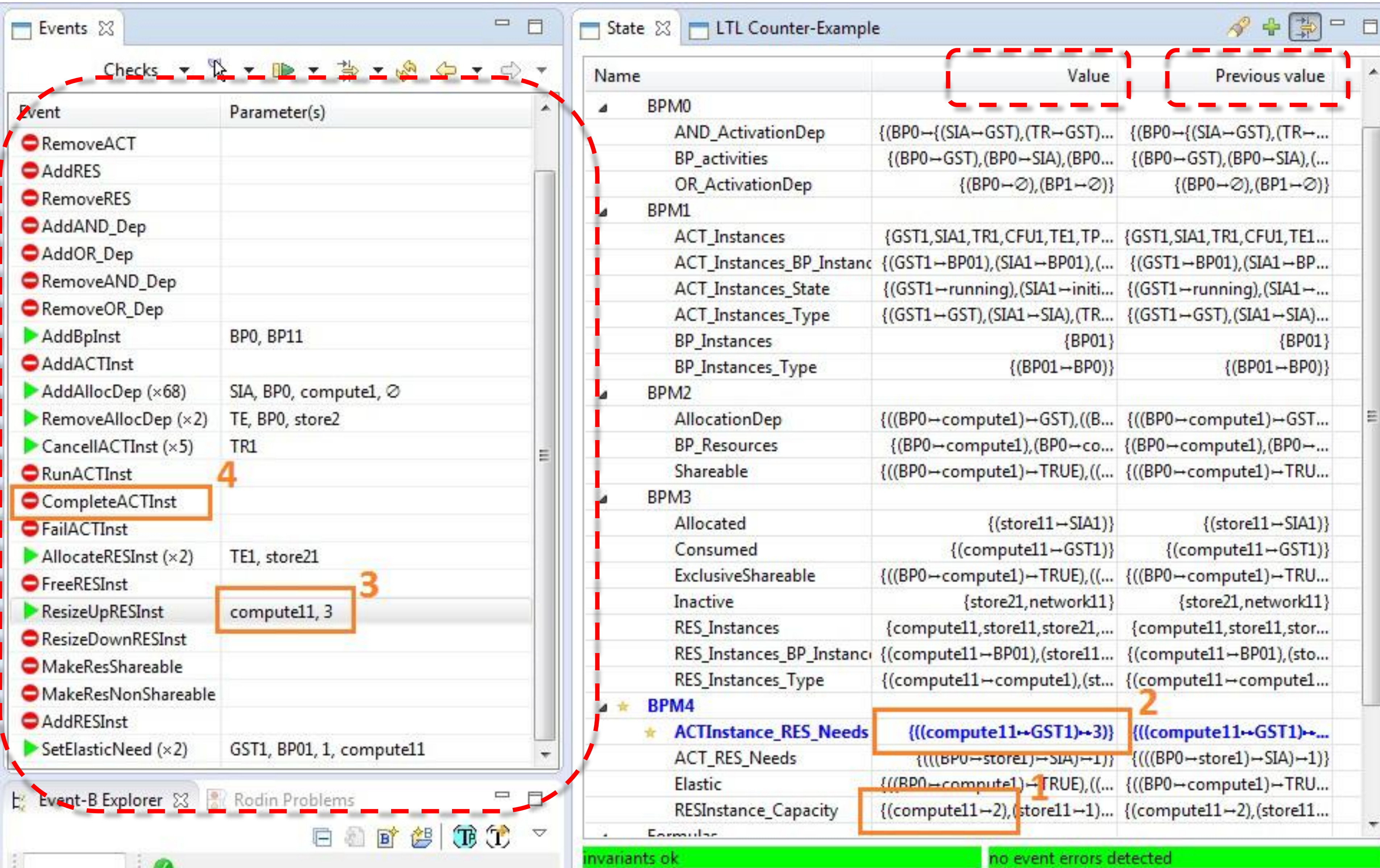
- **Interactive Proving Interface in Rodin**



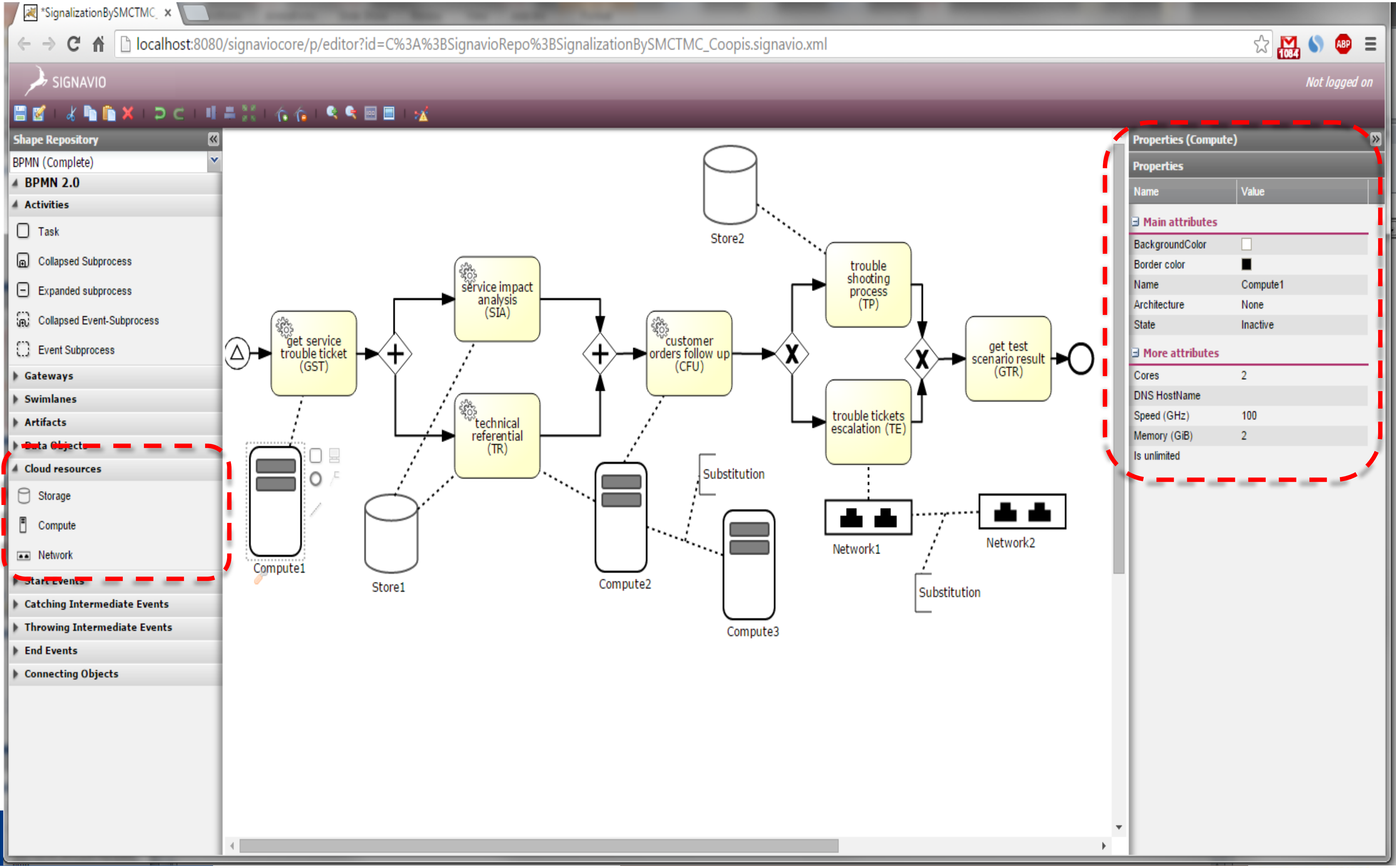
Verification & Validation

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


Events
Checks
Event
Parameter(s)
RemoveACT
AddRES
RemoveRES
AddAND_Dep
AddOR_Dep
RemoveAND_Dep
RemoveOR_Dep
AddBpInst
BP0, BP11
AddACTInst
AddAllocDep (×68)
SIA, BP0, compute1, ∅
RemoveAllocDep (×2)
TE, BP0, store2
CancelACTInst (×5)
TR1
RunACTInst
4
CompleteACTInst
FailACTInst
AllocateRESInst (×2)
TE1, store21
FreeRESInst
3
ResizeUpRESInst
compute11, 3
ResizeDownRESInst
MakeResShareable
MakeResNonShareable
AddRESInst
SetElasticNeed (×2)
GST1, BP01, 1, compute11
Event-B Explorer
Rodin Problems
State
LTL Counter-Example
Name
Value
Previous value
BPM0
AND_ActivationDep
BP_activities
OR_ActivationDep
{(BP0↦∅),(BP1↦∅)}
BPM1
ACT_Instances
ACT_Instances_BP_Instanc
ACT_Instances_State
ACT_Instances_Type
BP_Instances
{BP01}
BP_Instances_Type
{(BP01↦BP0)}
BPM2
AllocationDep
BP_Resources
Shareable
BPM3
Allocated
{(store11↦SIA1)}
Consumed
{(compute11↦GST1)}
ExclusiveShareable
Inactive
{store21,network11}
RES_Instances
RES_Instances_BP_Instanc
RES_Instances_Type
2
BPM4
ACTInstance_RES_Needs
{((compute11↦GST1)↦3)}
ACT_RES_Needs
Elastic
1
RESInstance_Capacity
invariants ok
no event errors detected

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

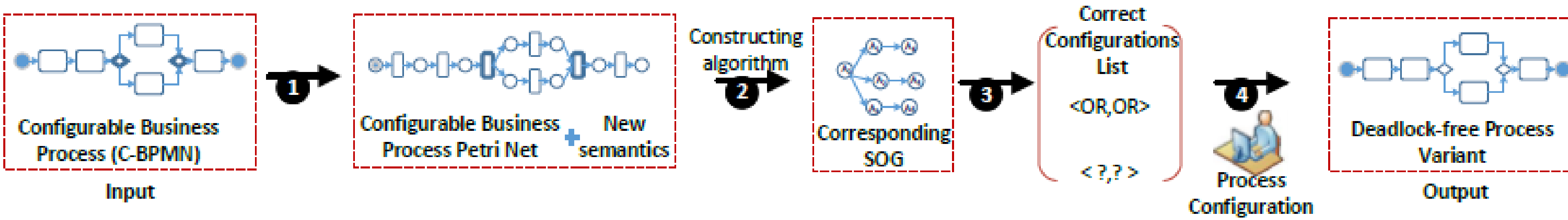
- **A formal verification model** for resource allocation in business process while considering:
  - Cloud resources properties
  - Different relationships between activities and resources.

- 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:

  [4] Souha Boubaker, Amel Mammar, Mohamed Graiet and Walid Gaaloul, « A Formal Guidance Approach for Correct Process Configuration», 14th International Conference On Service Oriented Computing, ICSOC 2016.

  [5] Souha Boubaker, Amel Mammar, Mohamed Graiet and Walid Gaaloul, An Event-B Based Approach for Ensuring Correct Configurable Business Processes, 23nd IEEE International Conference on Web Services, ICWS 2016, 460-467.

- 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:

  [1] Mohamed Graiet, Amel Mammar, Souha Boubaker and Walid Gaaloul, «Towards Correct Cloud Resource Allocation in Business Processes», IEEE Transactions on Services Computing 2016, *To appear*.

  [2] Souha Boubaker, Amel Mammar, Mohamed Graiet and Walid Gaaloul, «Formal Verification of Cloud Resource Allocation in Business Processes Using Event-B». 30th IEEE International Conference on Advanced Information Networking and Applications, AINA 2016, 746-753.

  [3] Souha Boubaker, Walid Gaaloul, Mohamed Graiet et Nejib Ben Hadj-Alouane, «Event-B based approach for verifying Cloud resource allocation in business process », 12th IEEE International Conference on Services Computing, SCC 2015, 538-545.

Thank you for your attention

Souha Boubaker

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