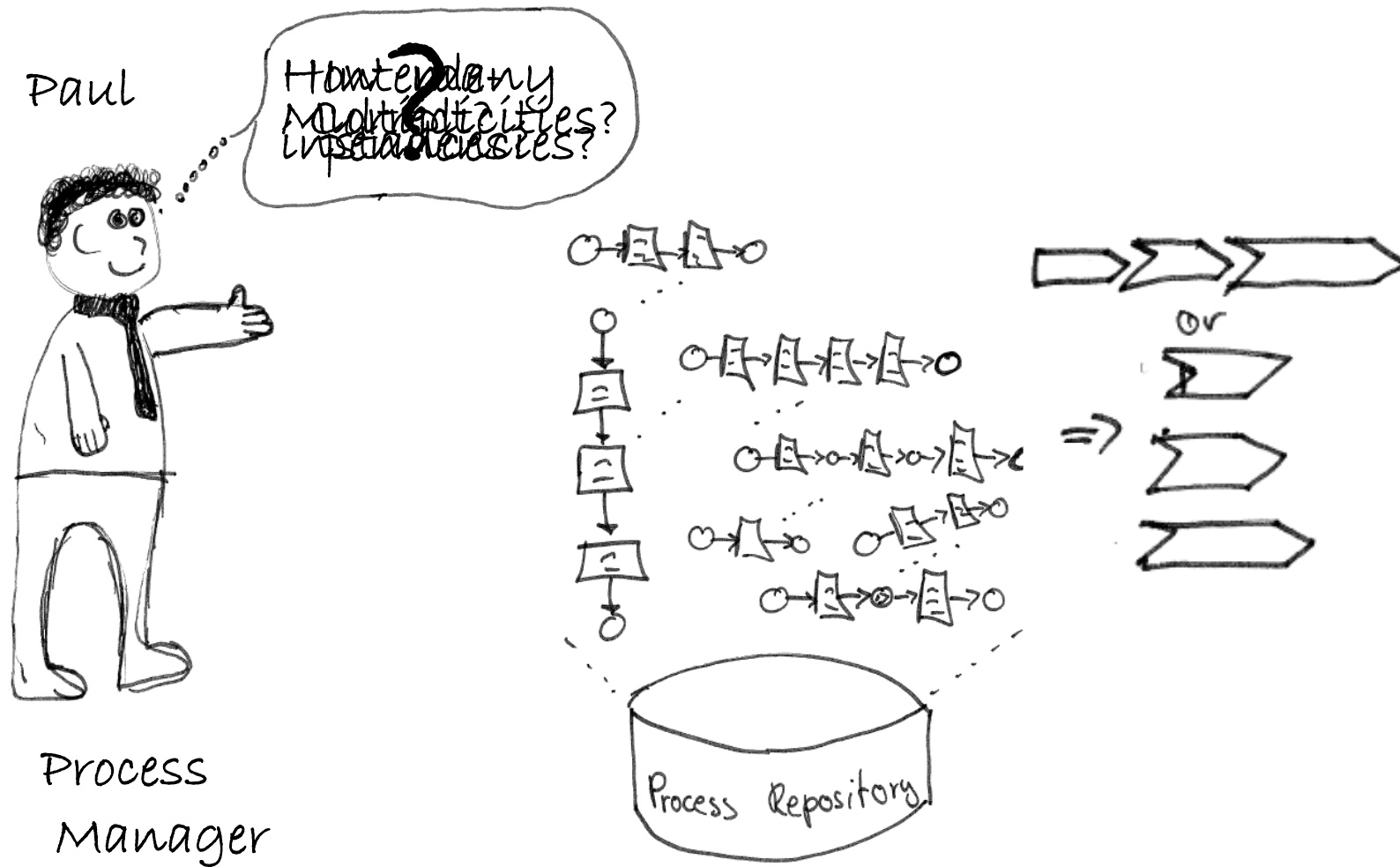


Business Process Architectures with Multiplicities: Transformation and Correctness

Rami-Habib Eid-Sabbagh, Marcin Hewelt,
Mathias Weske

Motivation



Definition Business Process Architecture

A business process architecture defines the relations between processes within a process collection, as well as the guidelines for organizing them.

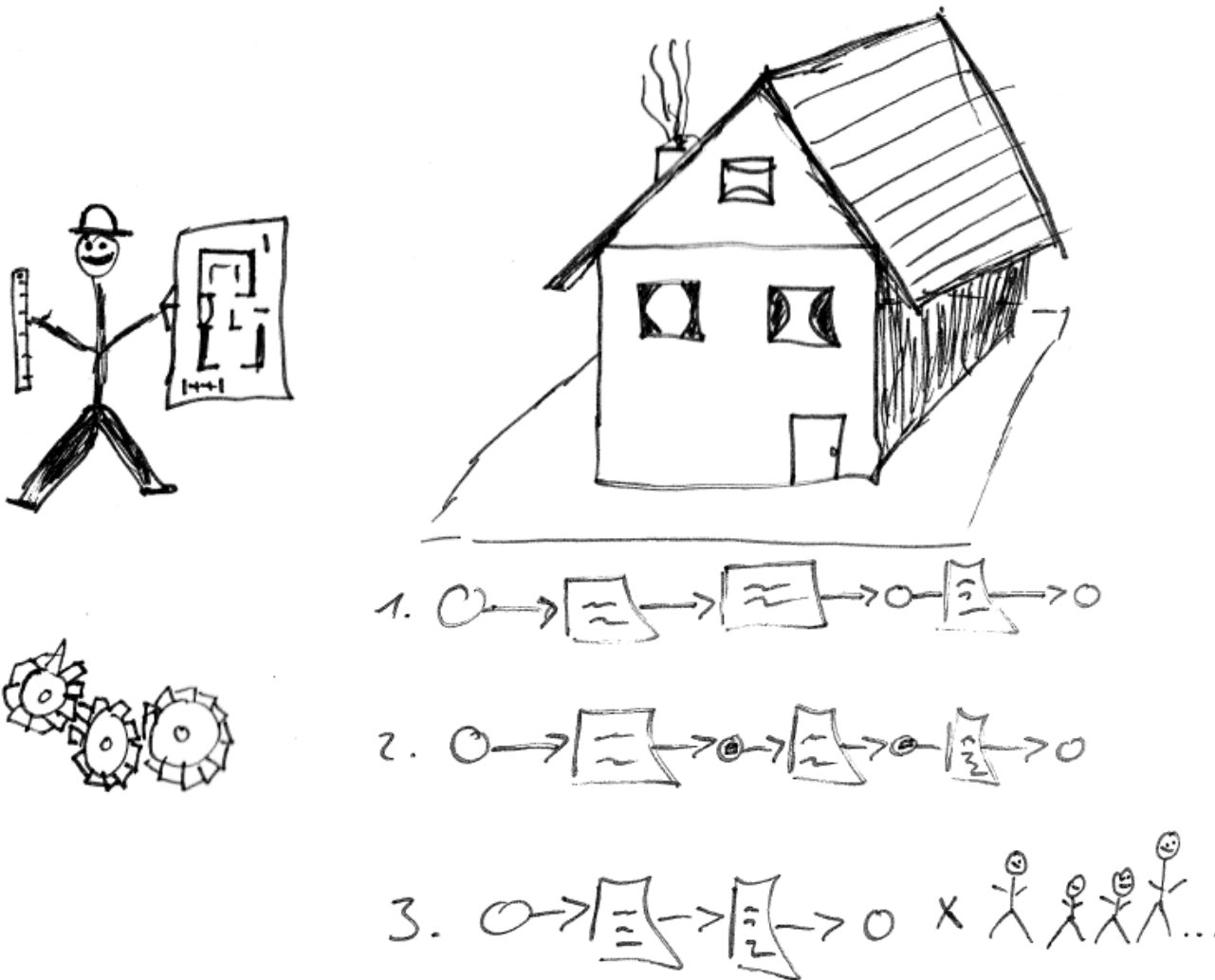
[Dijkman, R.M., Vanderfeesten, I., Reijers, H.A.: The Road to a Business Process Architecture: An Overview of Approaches and their Use. (2011)]

Aim of our Approach

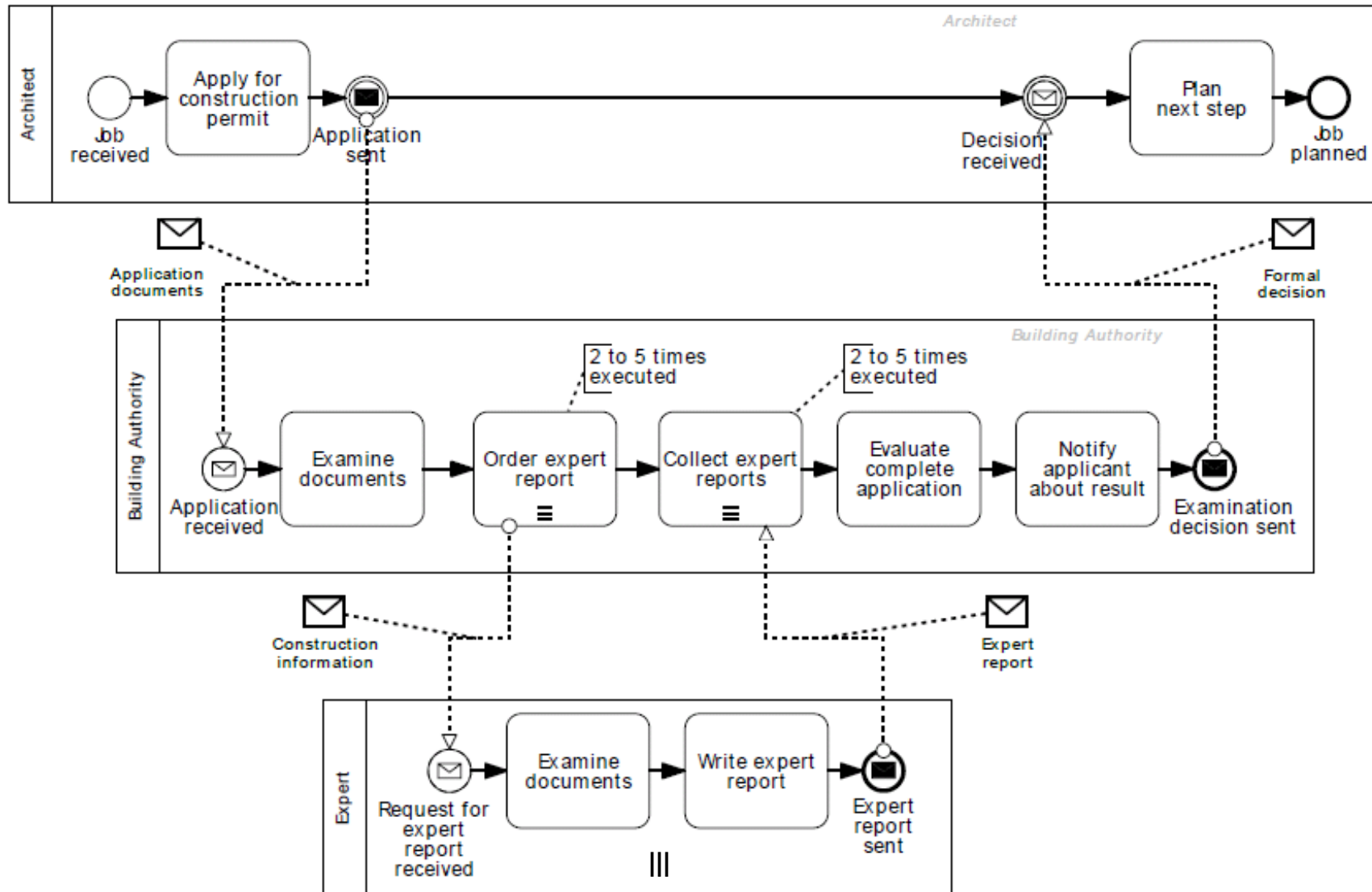
Business Process Architecture (BPA) methodology for process collections

- For depicting interdependencies
 - Trigger relations
 - Message flow relations
- For depicting **multi-instance** and **multi-communication**
- For **analysing** interdependencies with **multiplicities**

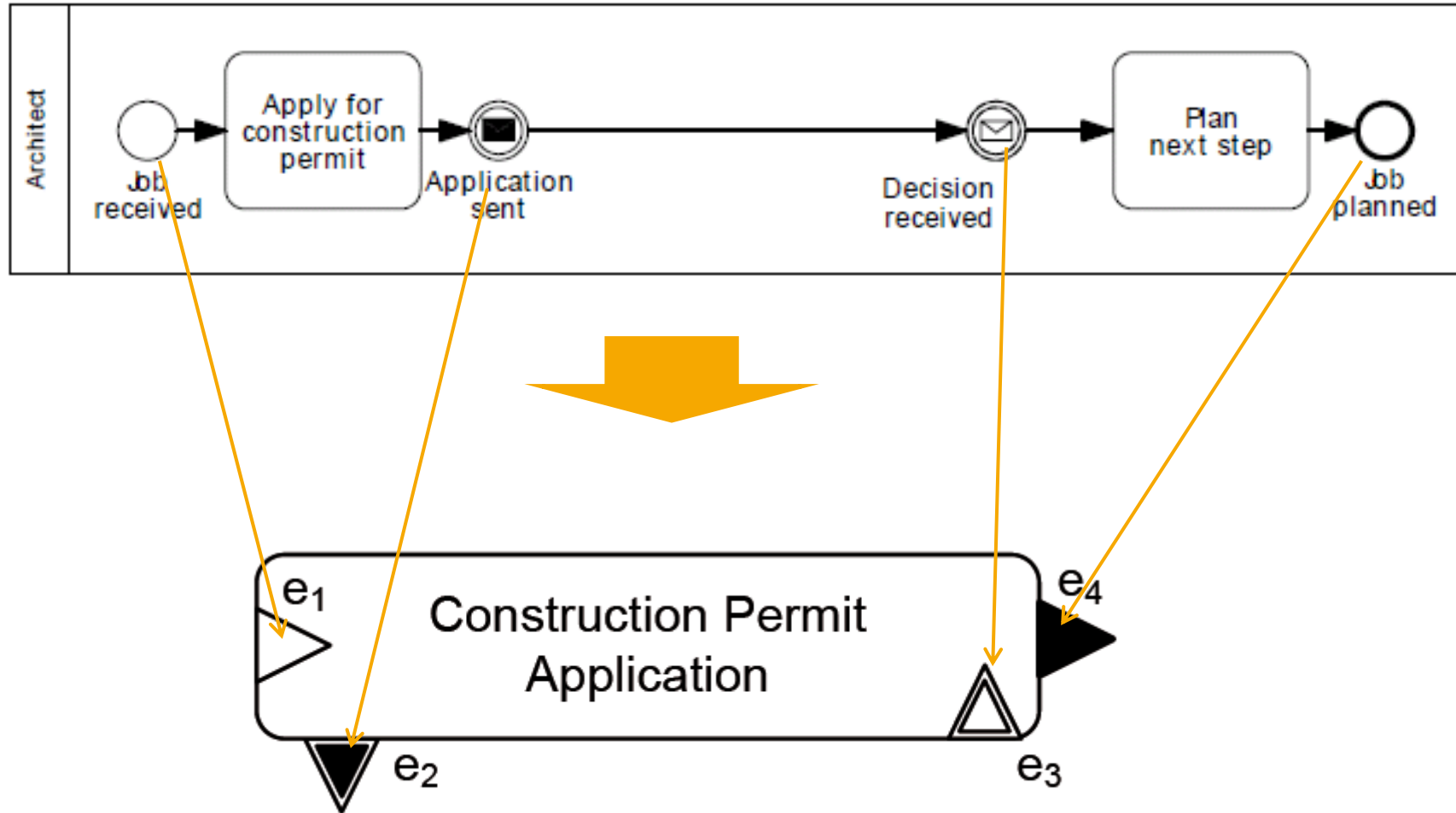
Scenario: Simplified Construction Permit Application



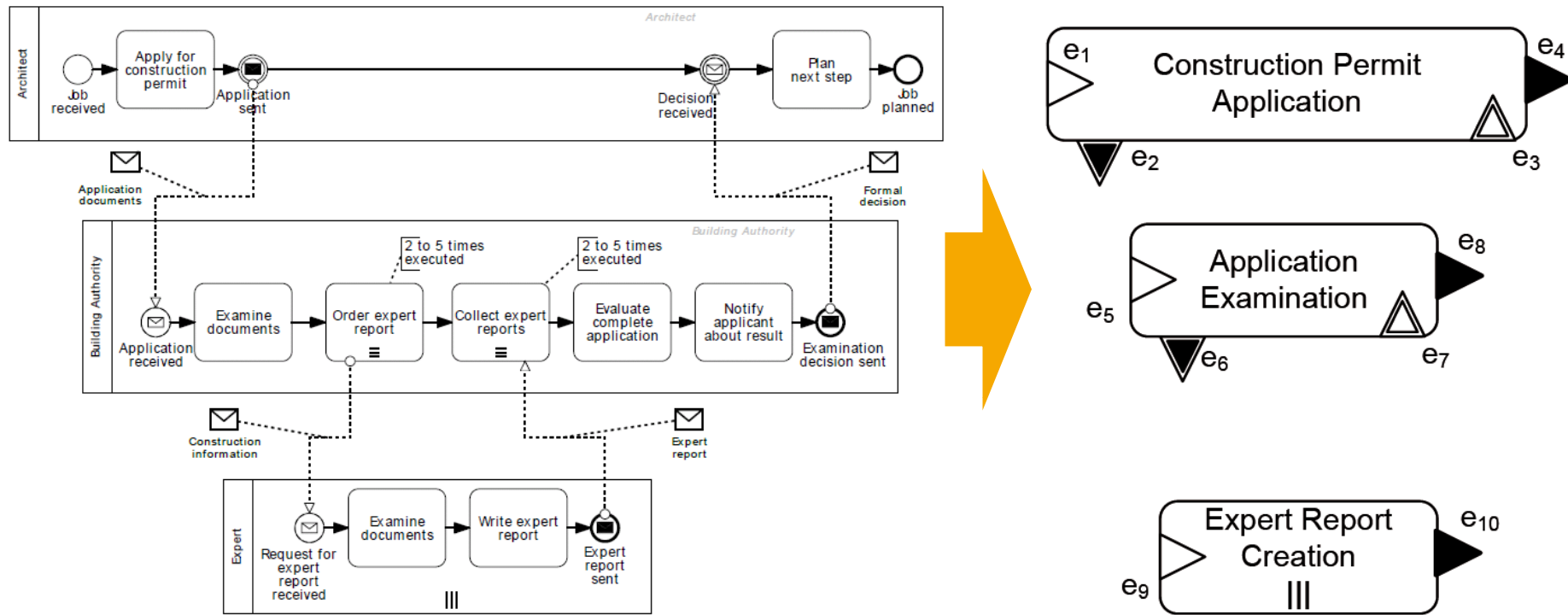
Construction Permit Application



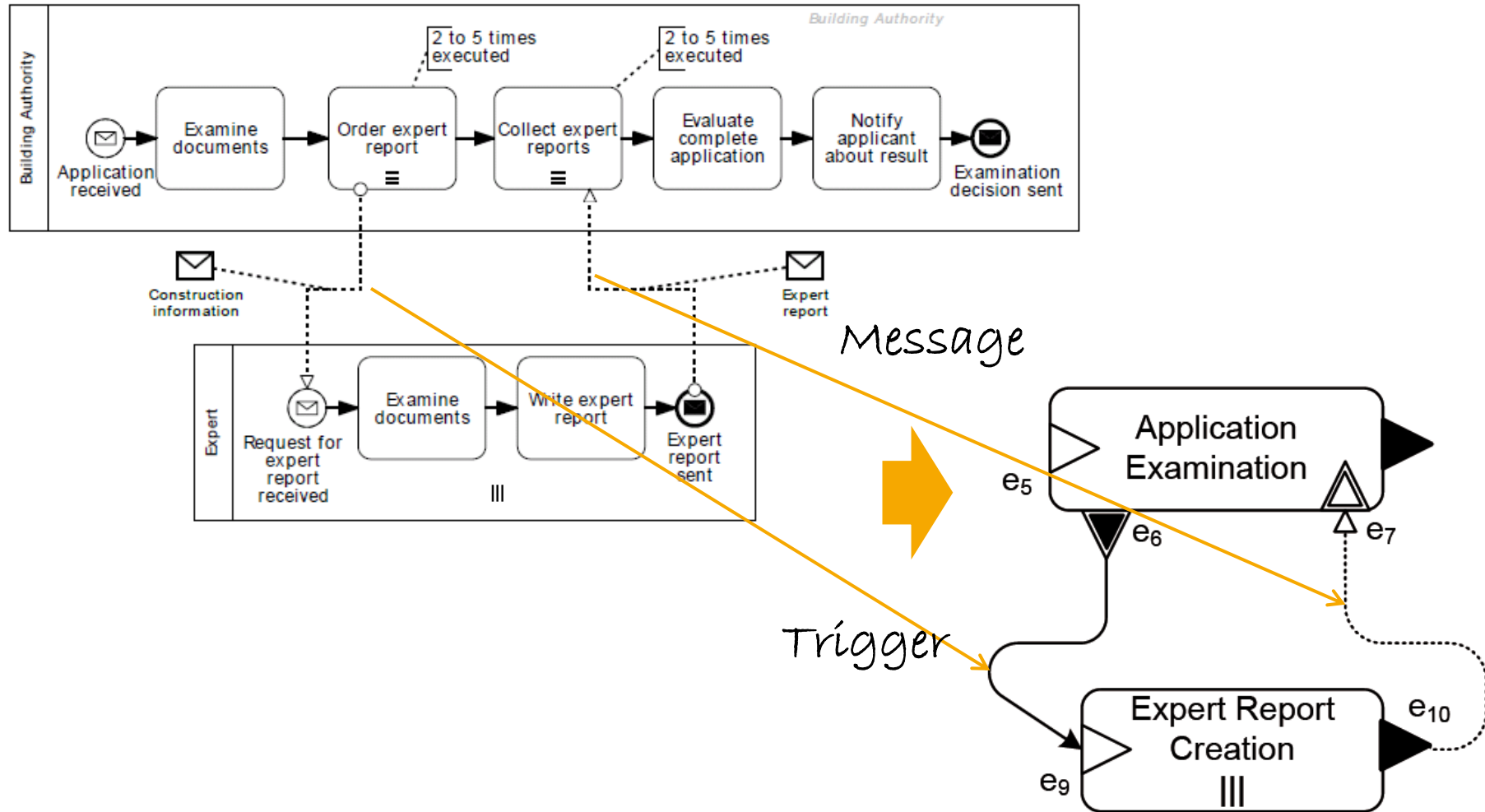
Deriving a BPA Process from a Process Model



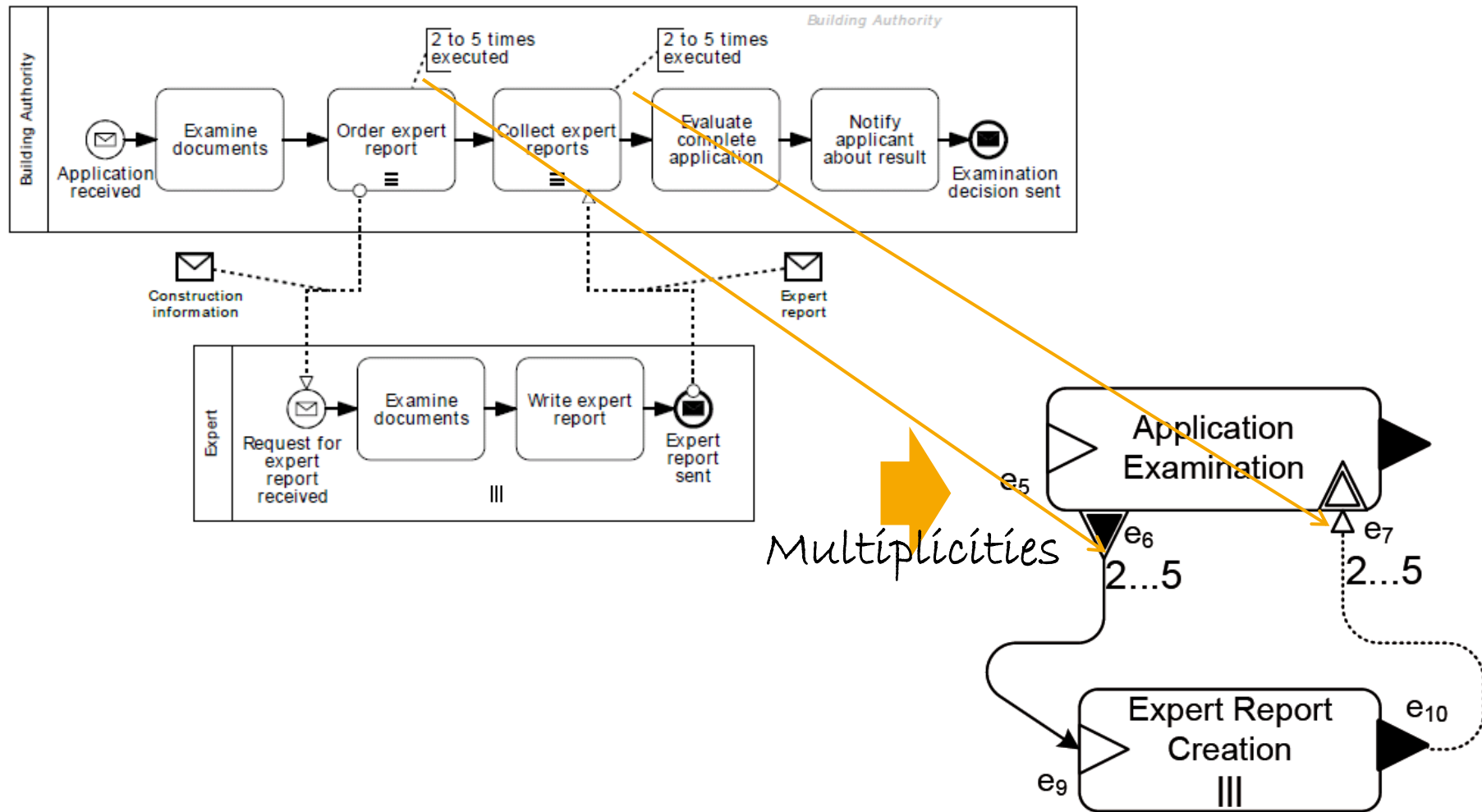
Deriving BPA Processes



Deriving BPA Trigger and Message Flow Relations



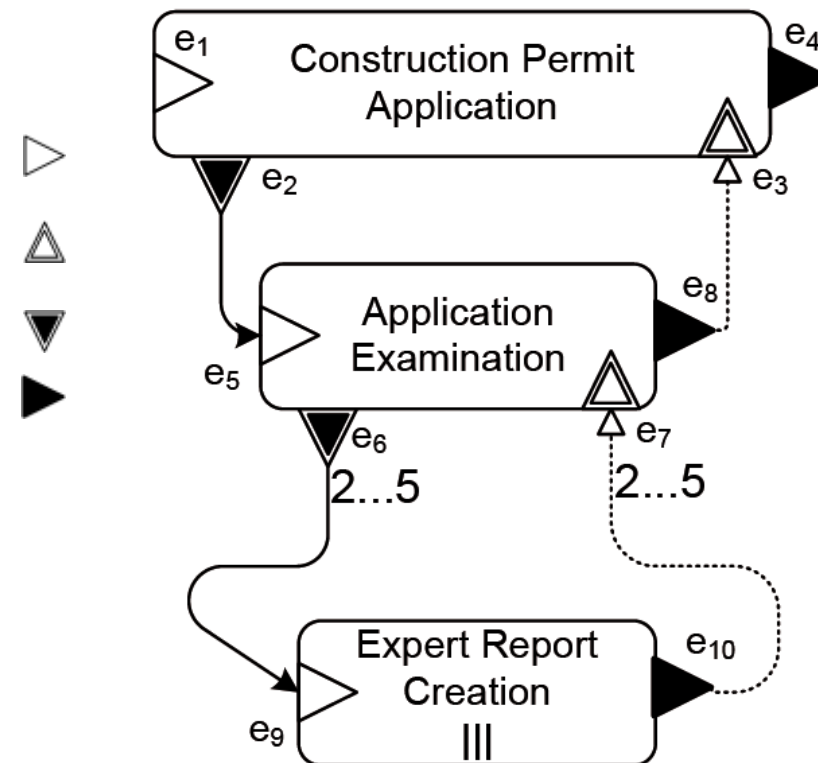
Adding Multiplicities to BPA Events



Business Process Architecture (BPA)

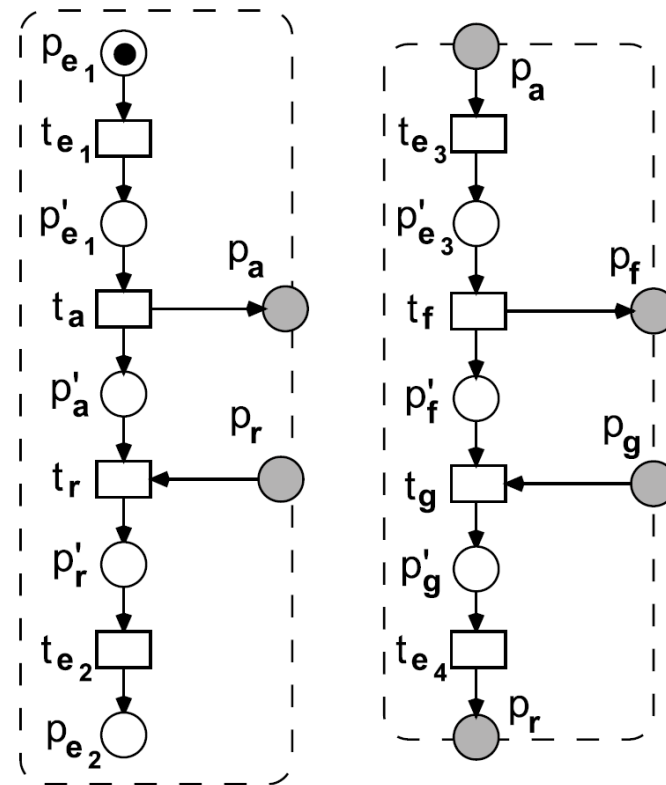
A BPA consists of

- Processes
 - Limited to sequence of events
 - Start event
 - Intermediate catching events
 - Intermediate throwing events
 - End event
- Events have multiplicities
- Relations
 - Trigger flow →
 - Message flow - - - ->



Open nets

- Petri nets with interface places and composition
 - Well established formalism
- Challenges
 - Multiple instances
 - Multi-communication
 - Multicast / Multireceive
 - Splitter / Collector



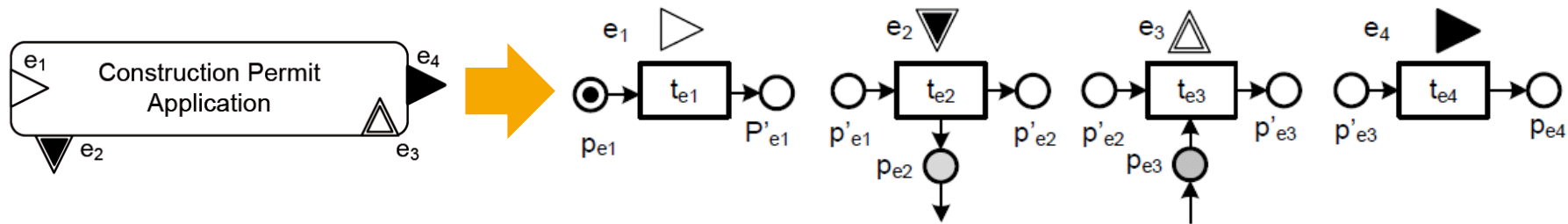
BPA Analysis Algorithm

1. Transform BPA processes into Open nets
2. Create intermediary Open nets
3. Compose resulting nets
4. Create correctness-formulae
5. Analyse composed Open net

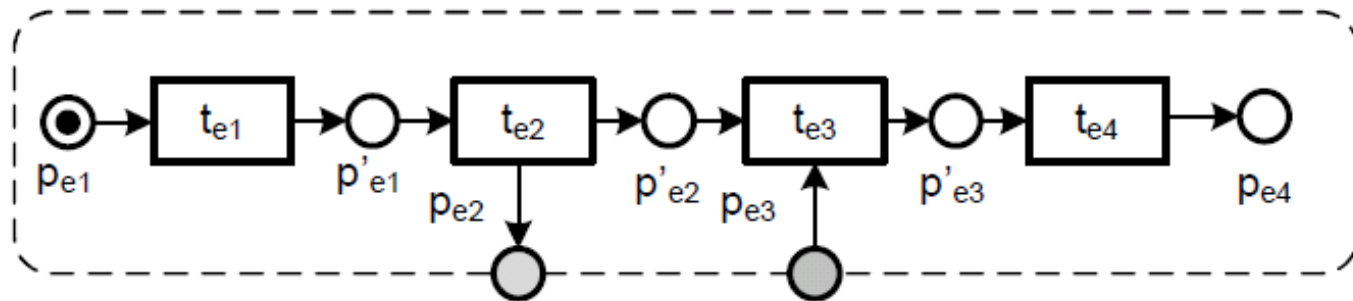
Transformation: BPA Process to Open Nets

BPA process

Transformation of process events

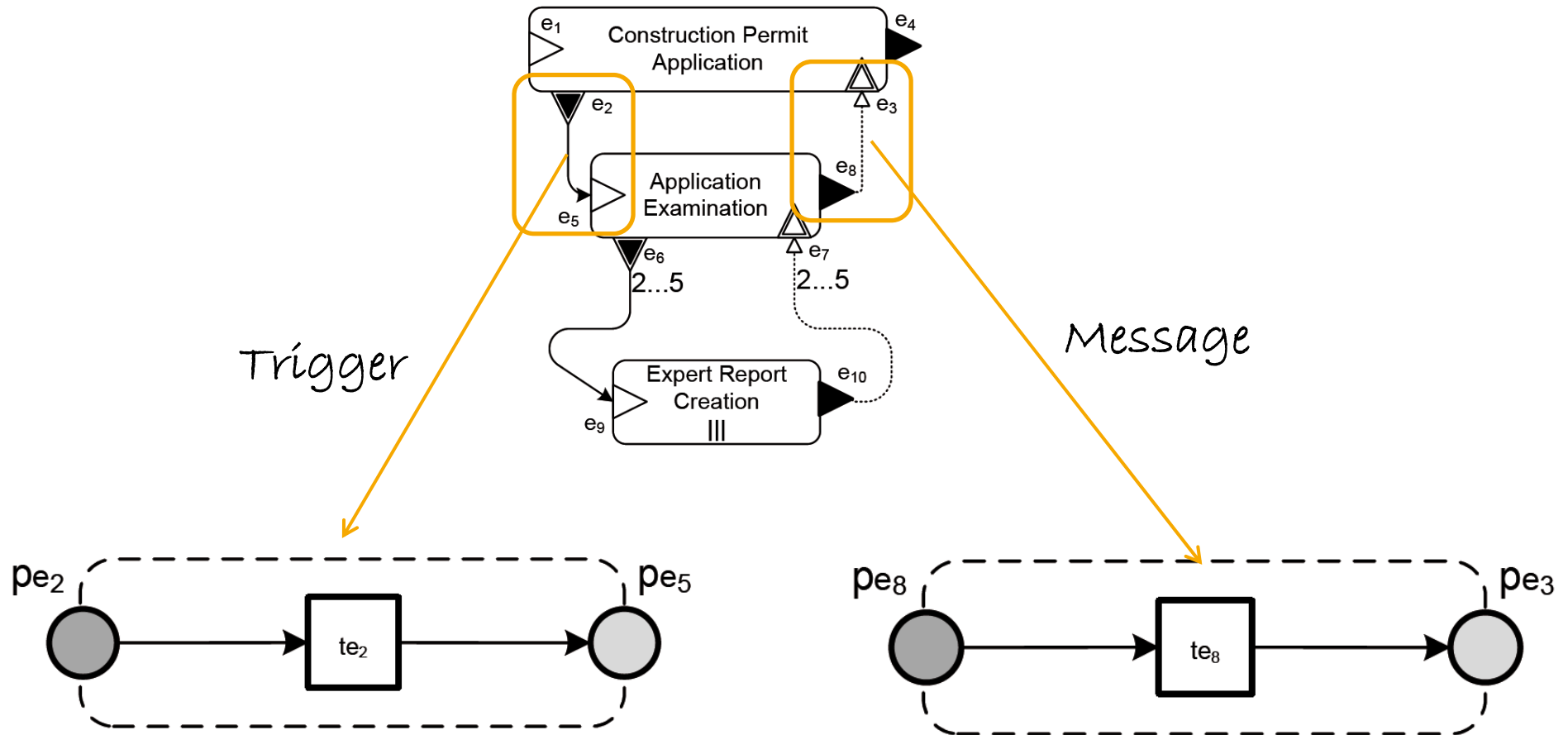


Behavior of BPA process



Simple Intermediary Open Nets

Trigger Flow / Message Flow Relations

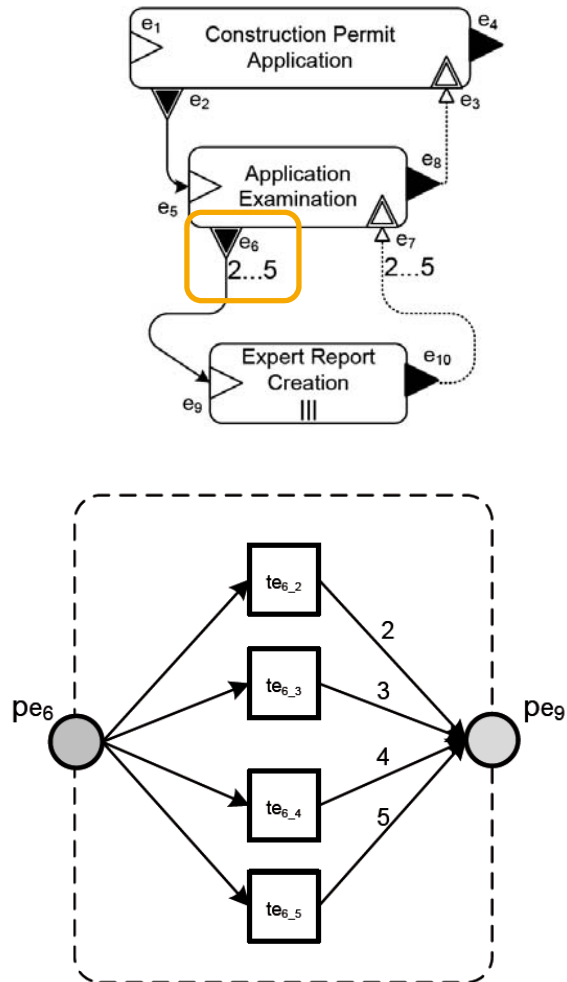


Intermediary Nets

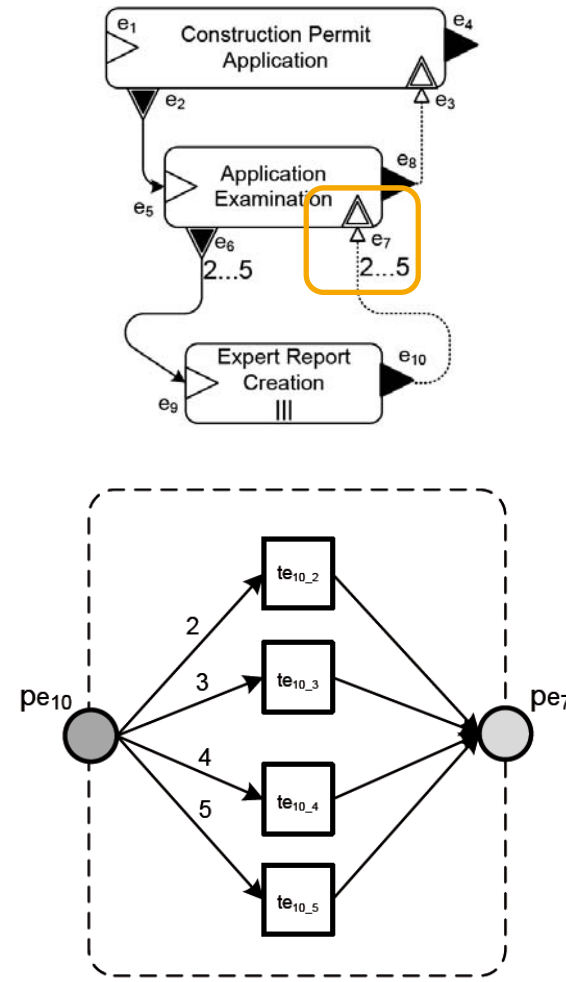
Net Schema for Multicast/ Multireceive



Multicast



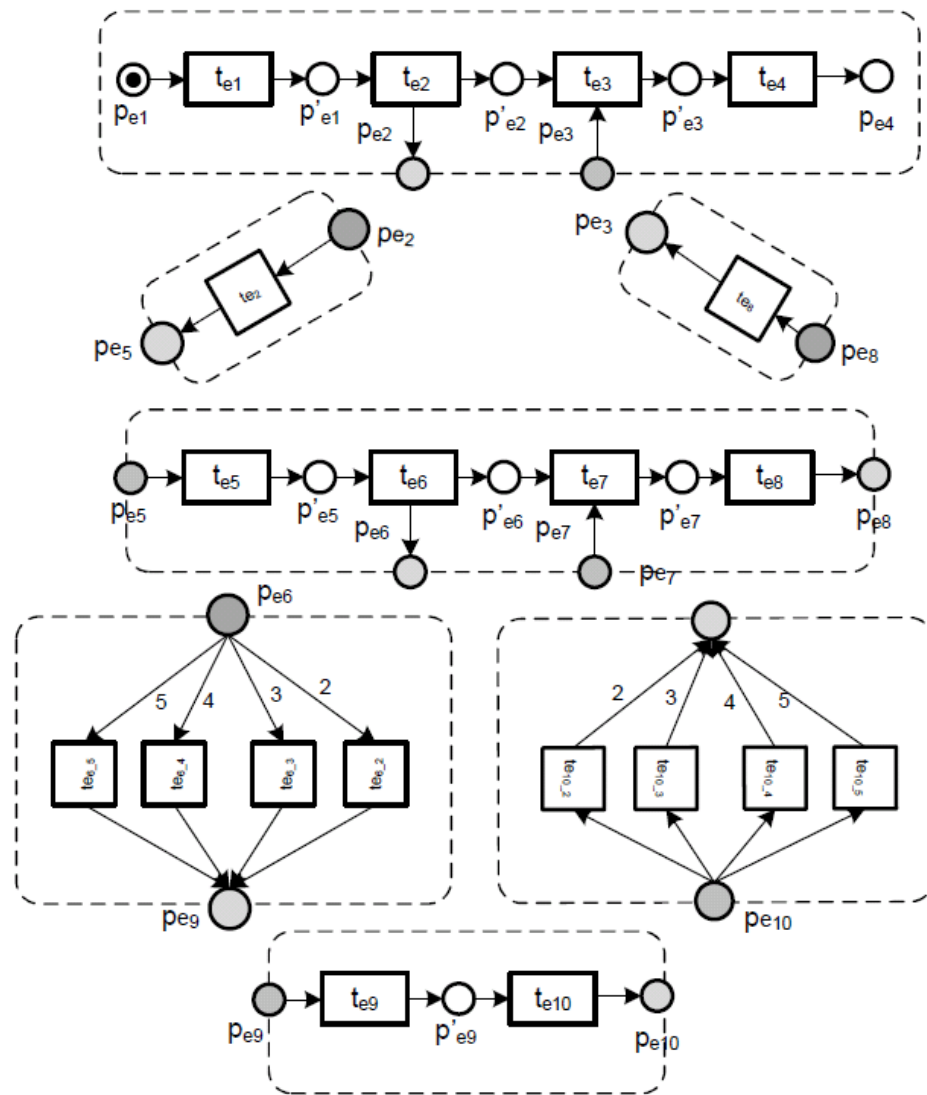
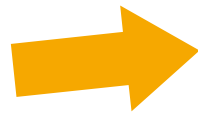
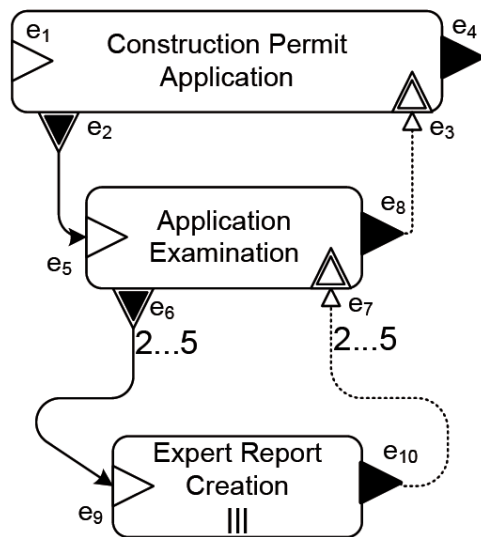
Multireceive



Composition of Transformed Processes and Intermediary Nets

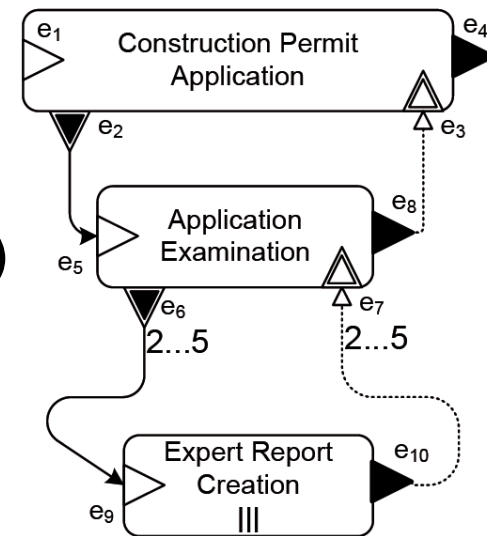


- Fuse matching places



BPA Correctness Criteria

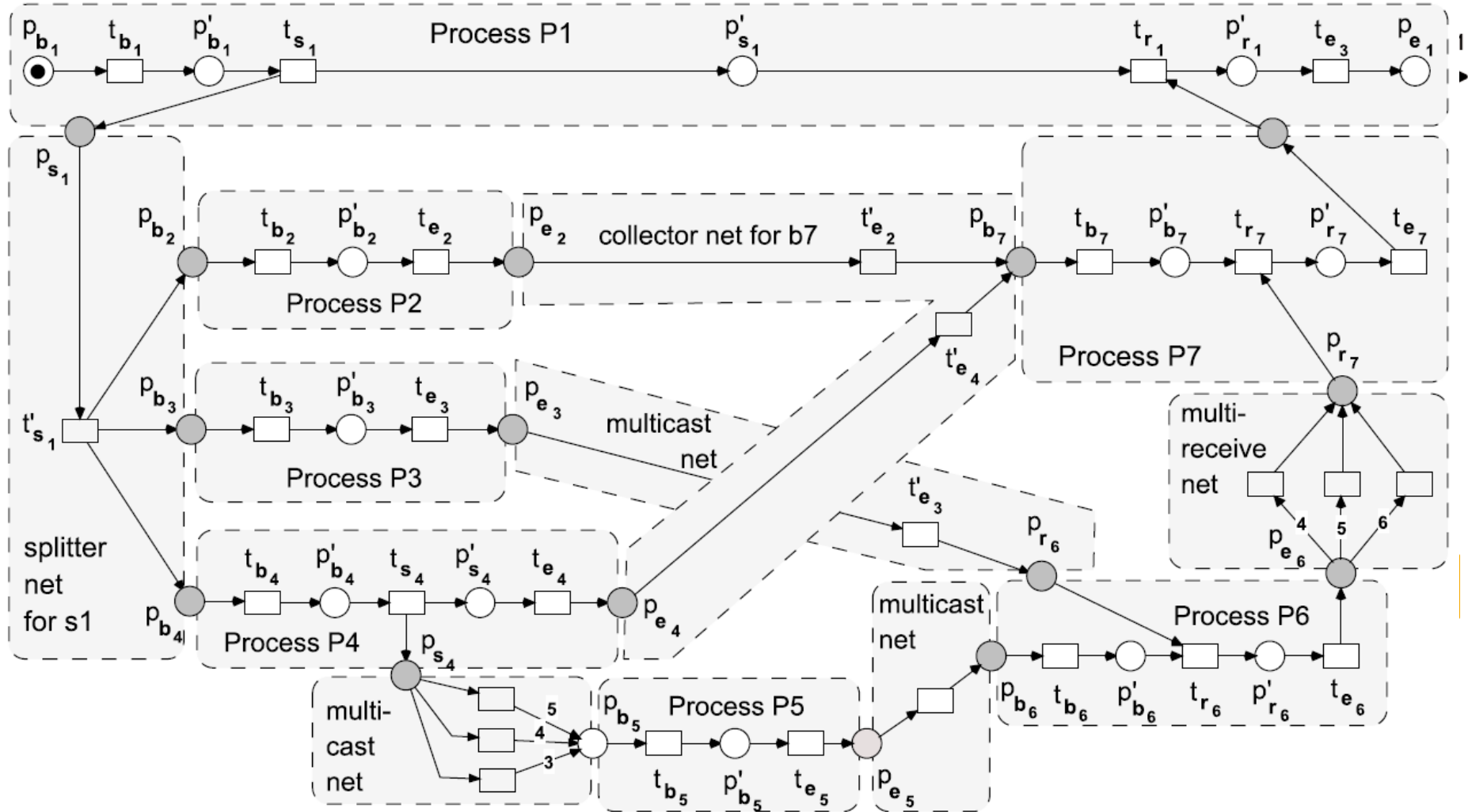
- (Lazy-) termination
 - Each process instance terminates
 - (at least one process instance terminates)
- No livelocks
- No dead processes
 - Each instantiated at least once in terminating runs



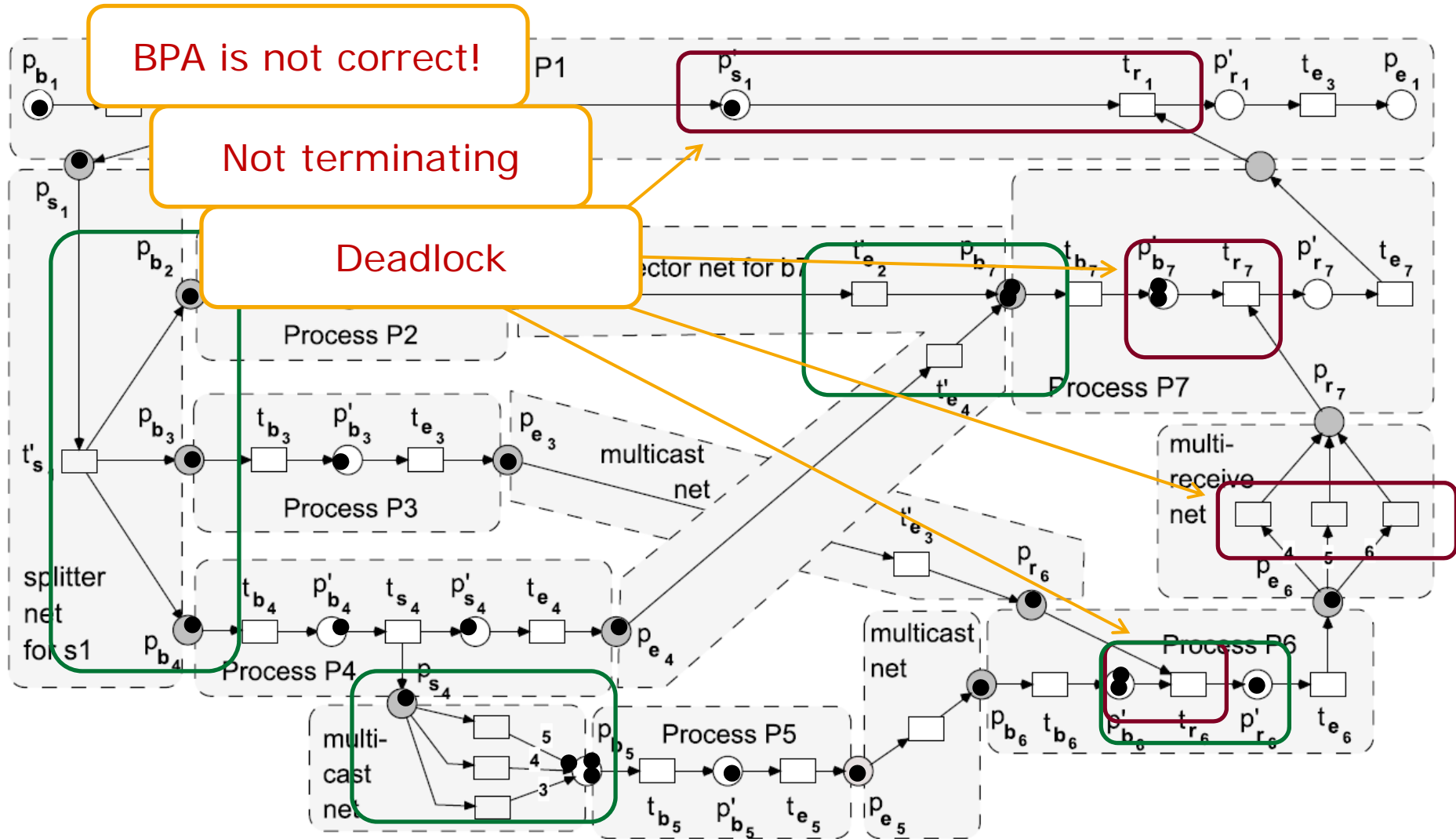
Translate into ctl-formulae and
model check

BPA – Public Administration

Is this BPA correct?

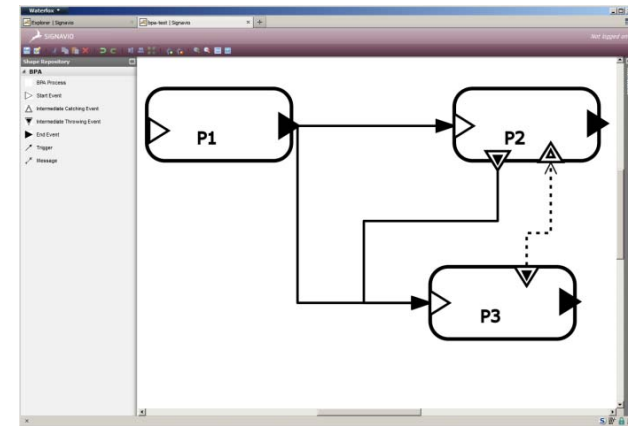
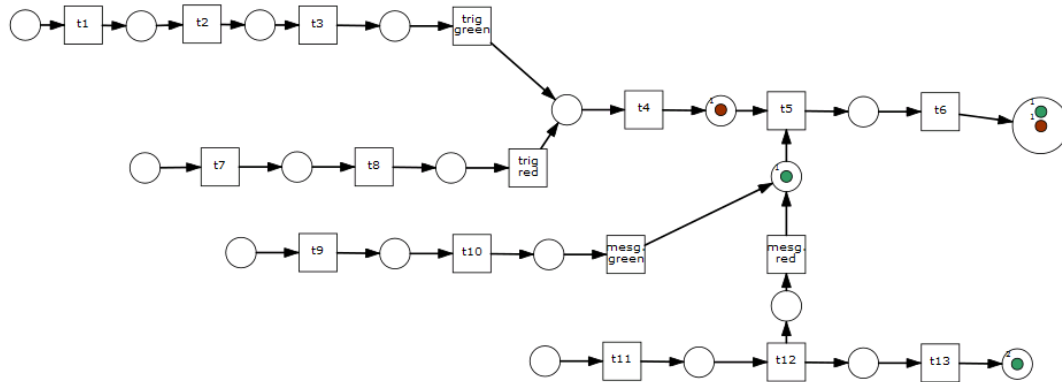
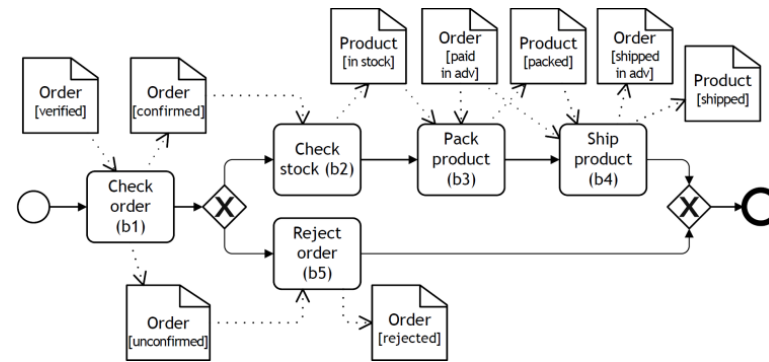


Transformation of Use Case



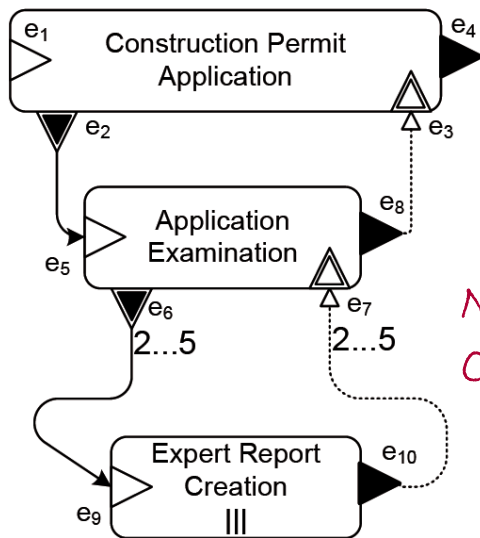
Future Work

- BPA and data
- BPA analysis tool
- Correlation in BPA



Summary

BPA concept

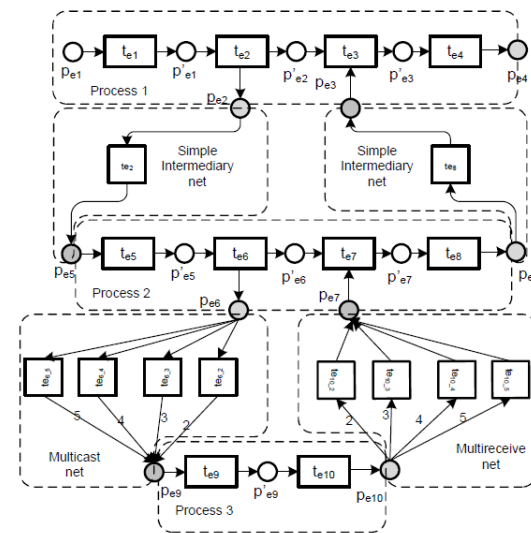


Trigger/
Message
flows

Multi-
Communication

Correctness
criteria

Transformation
into Open nets



Analysis according to
correctness criteria

Summary

- BPA concept
 - Trigger/ flows
 - Multi-Communication
 - Correctness criteria
- Modular transformation into Open nets
- Analysis according to correctness criteria

Limitations of previous approaches

- **Pattern Approach**
 - Only 1:1 Interaction
 - Difficult to check multiple relations
 - Only direct interdependencies considered

- **TriggerFlow-Net Approach (CAiSE 2013)**
 - no analysis for multiplicity aspects

Limitations

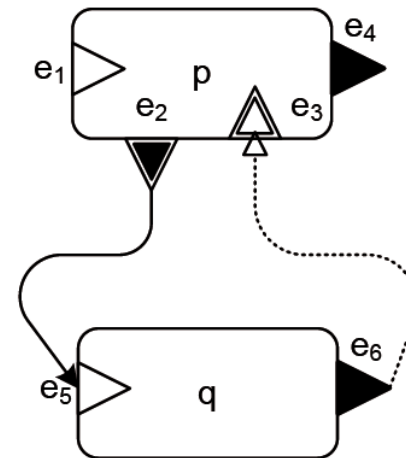
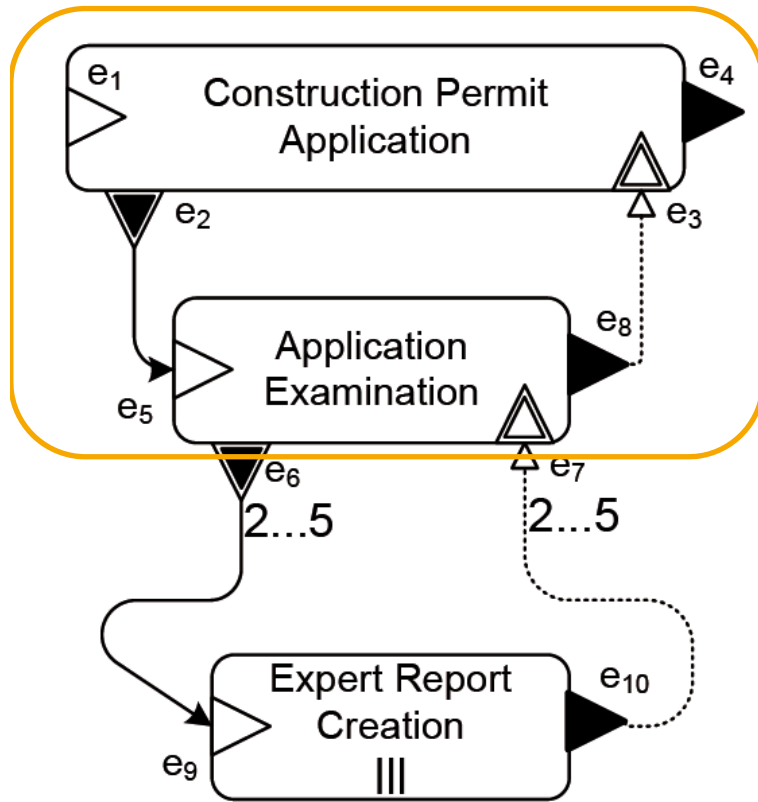
- BPA processes are considered sequences
- Correlation not addressed
- Very large BPAs may run into computational problems

Why not use BPMNchoreography Diagrams

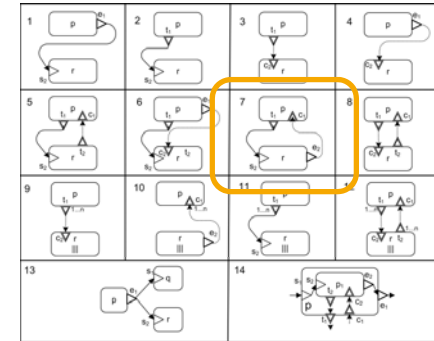
- **Choreographies**
 - Different Focus
 - Depict the interaction behavior
 - Do not provide overview of relations between processes
 - No Multi-Instance Concept
 - No Trigger Concept
 - Assumed that messages are sent
 - However not checked if process that supposed to send the message is triggered

Previous Work BPM2012

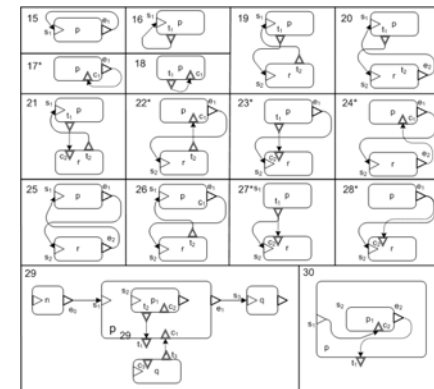
Analysis with use of Patterns



Normal Patterns



Anti-Patterns



Design is about removing the superfluous and adding the meaningful.

[Adapted from Garr Reynolds]

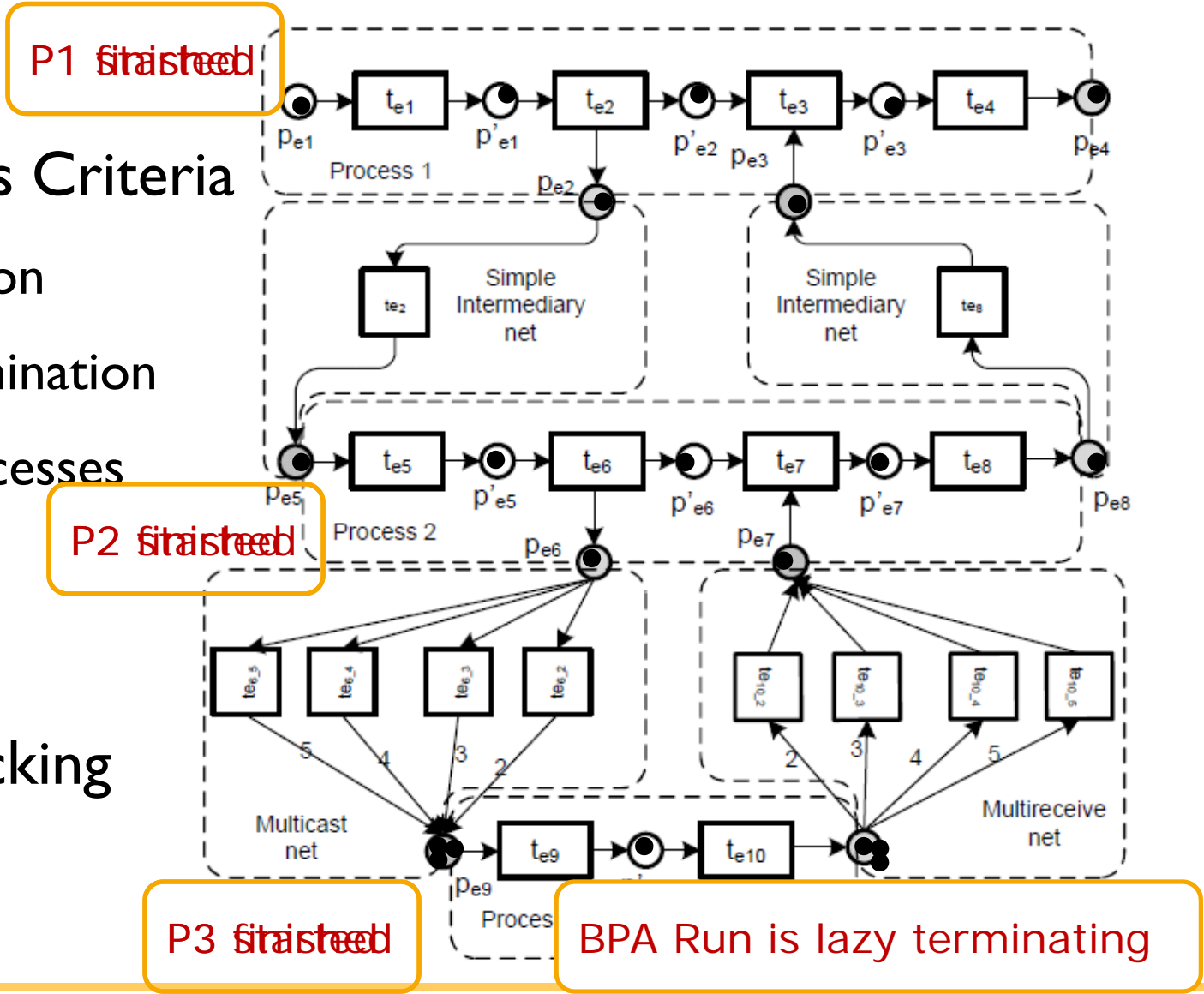
Contact:
Rami-Habib Eid-Sabbagh
rami.eidsabbagh@hpi.uni-potsdam.de

Analysis of Transformed BPA

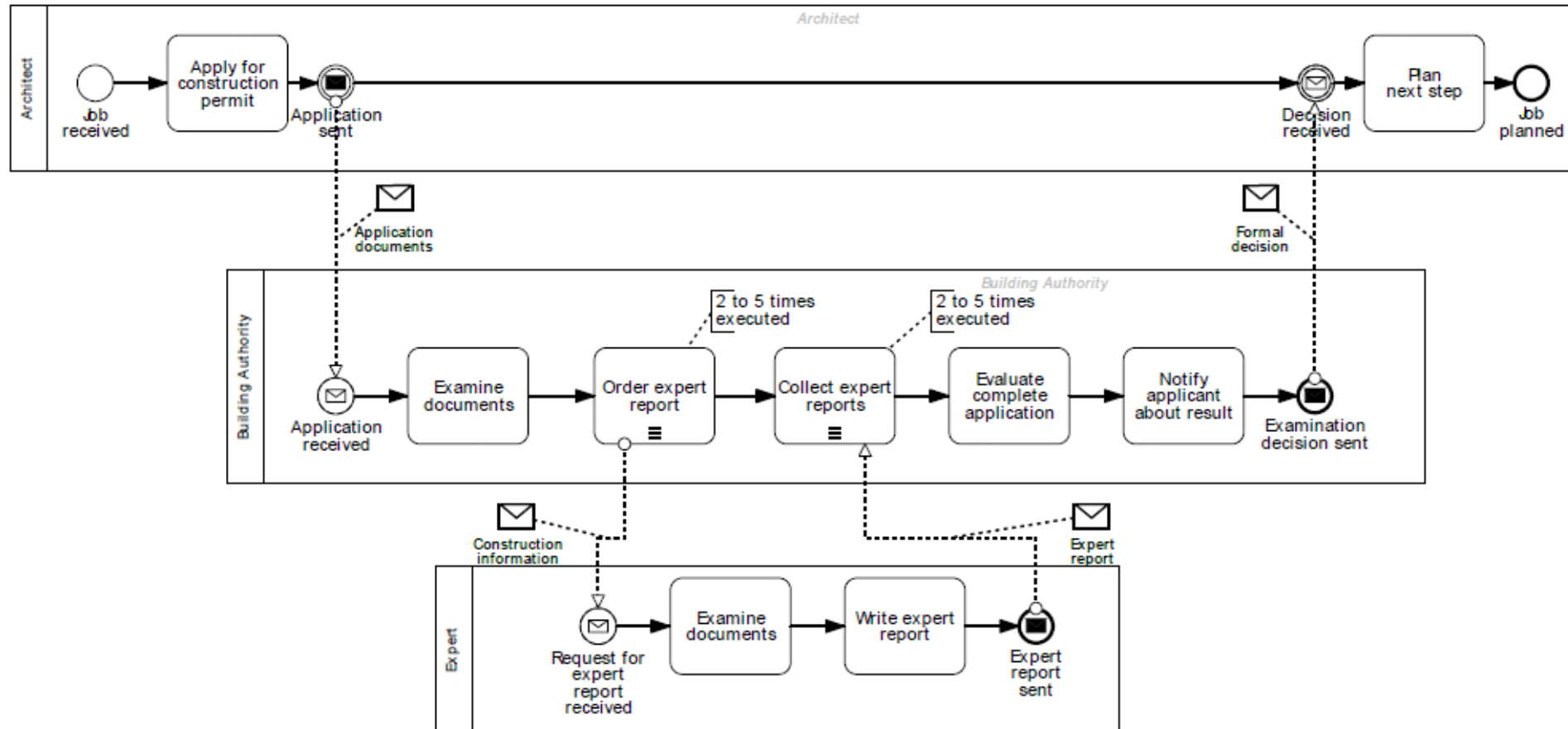
- Correctness Criteria

- Termination
- Lazy Termination
- Dead Processes
- Livelocks

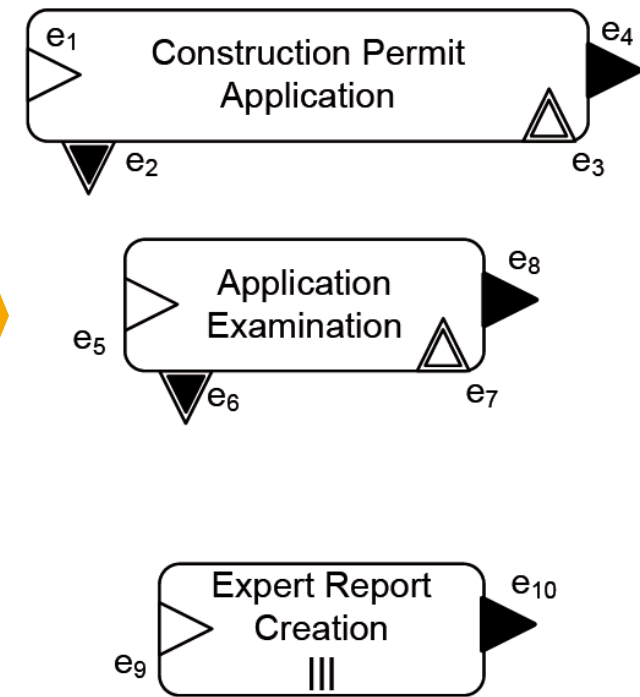
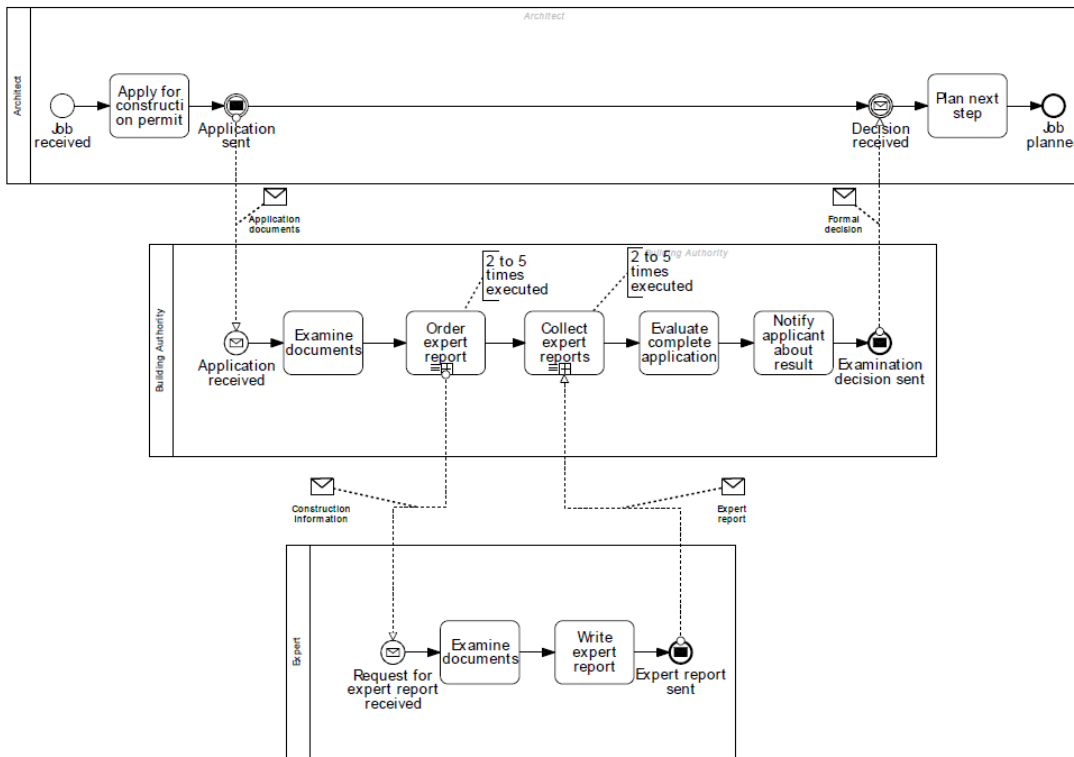
- Model Checking



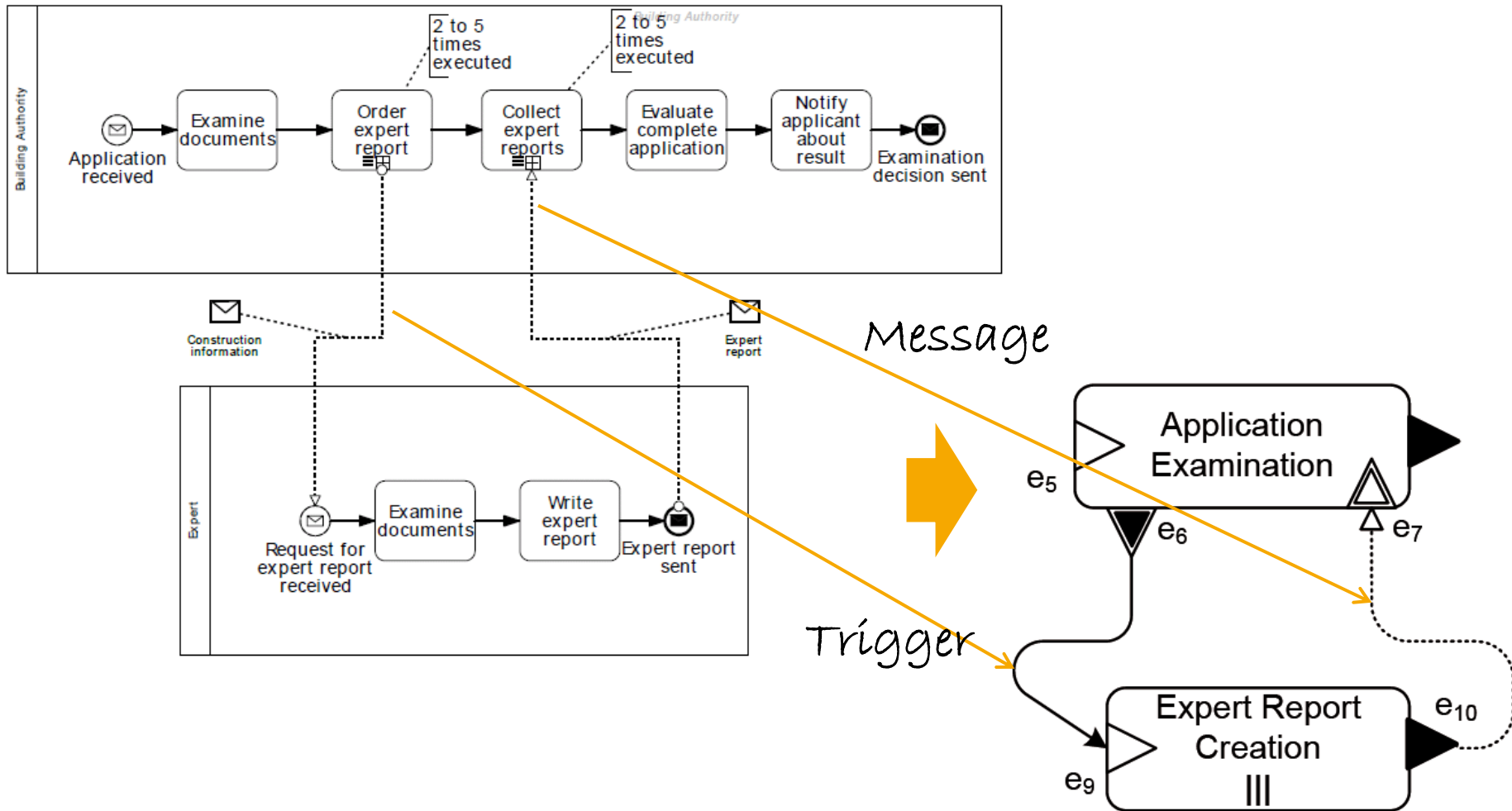
Construction Permit Application



Deriving BPA Processes

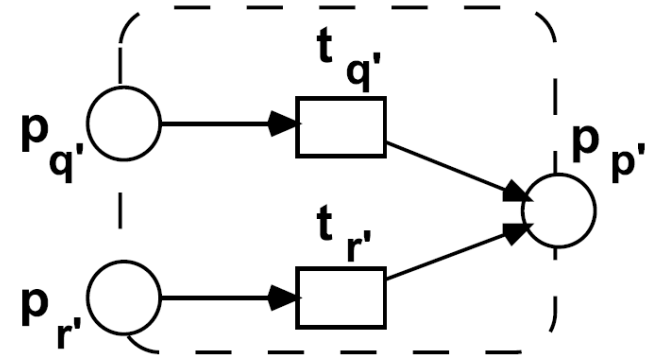
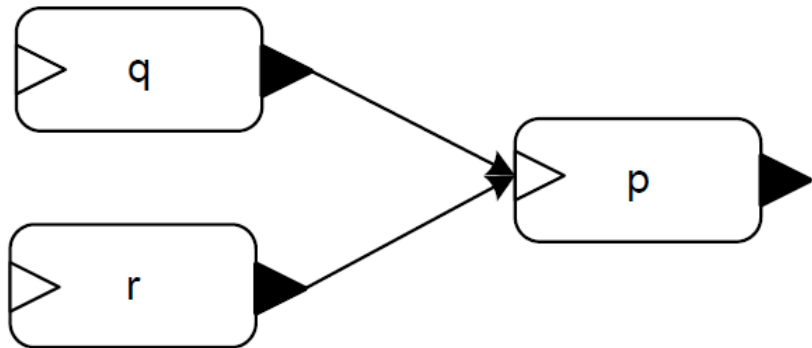
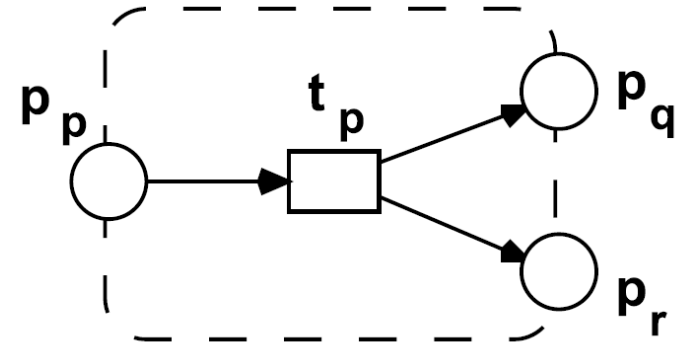
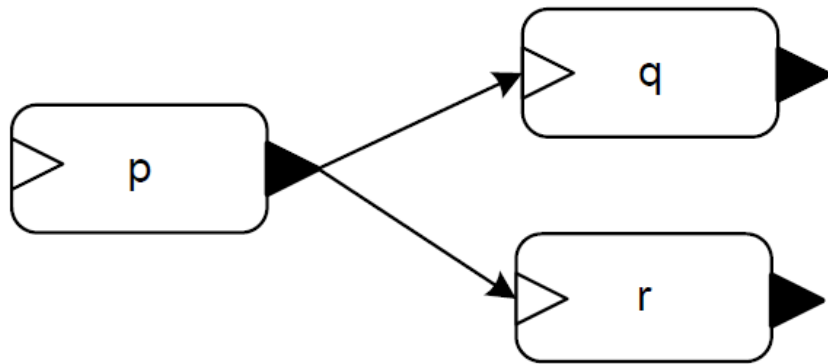


Deriving BPA Trigger and Flow Relations

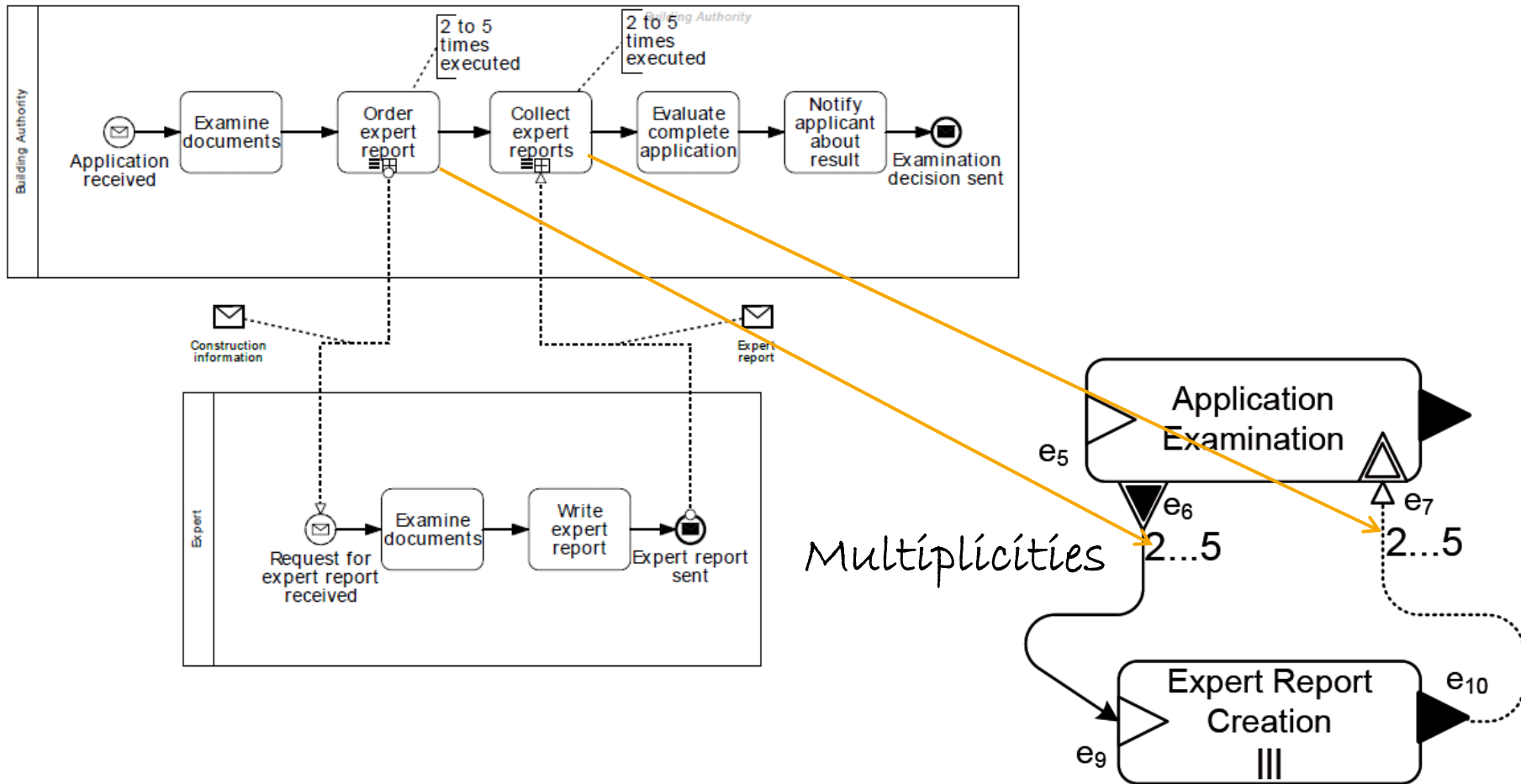


Intermediary Nets

Splitter net/ Collector net



Adding Multiplicities



Checking BPA Correctness

BPA run is terminating

P1 finished

BPA Correctness

- (Lazy-) termination
- No dead processes
- No livelocks

P2 finished

- Model Checking

All states of P3 finished

