Model Driven Engineering

Analysis of BPMN Models

Addis Gebremichael

22 January 2015

Universiteit
Antwerpen

Business Process Modeling Notations

Outline

- Background
- Experiment
  - BPMN formalism in AToMPM
  - Mapping using model transformation
- BPMN model analysis
- Conclusion

Universiteit Antwerpen

Background

Business Process Modeling Notations

- Common Standard Notation used to capture business processes
- Increase in Complexity of Business Processes
- Enable Process Analysis
- Creates a bridge for the gap between business process design and enactment
- Problem
  - Lack of unambiguous semantic definition
  - Prone to designer errors
  - Lack of tools supporting verification of error free BPMN models

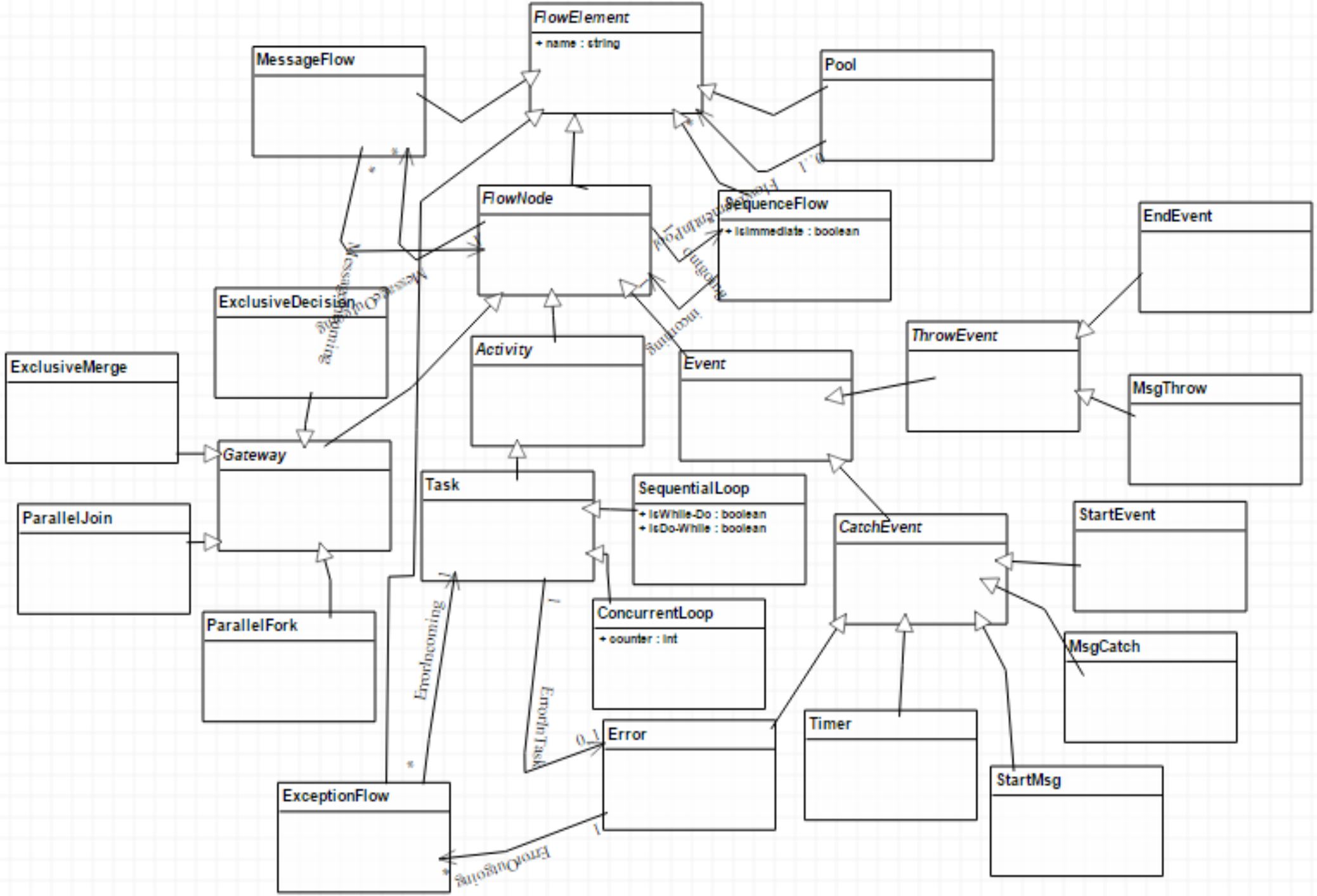
R. M. Dijkman, M. Dumas, C. Ouyang, Semantics and analysis of business process models in BPMN, Information and Software Technology 50 (12) (2008) 1281--1294.

W. P. van der Aalst, Business process management: A comprehensive survey, ISRN Software Engineering 2013 (2012) 37.

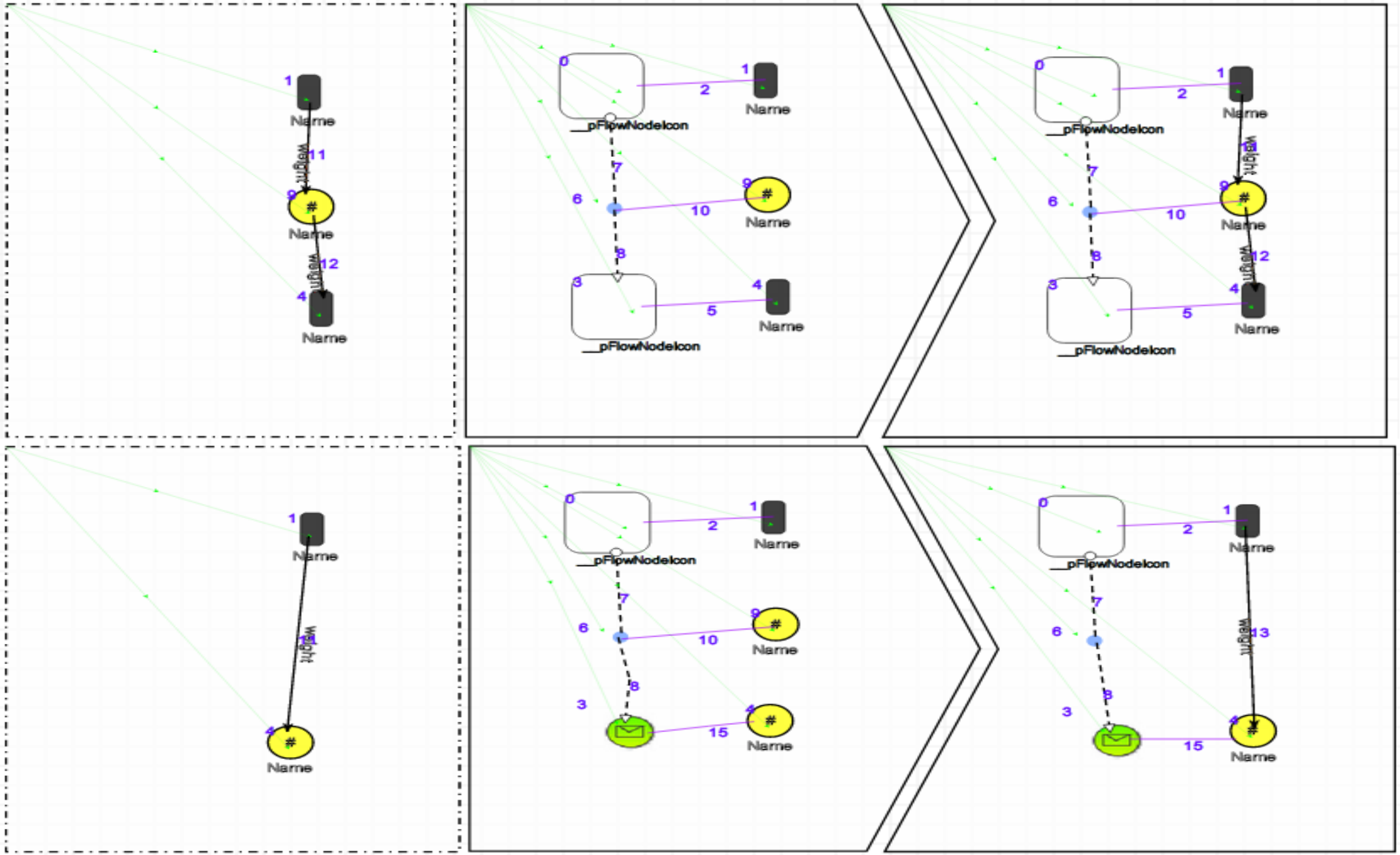
Experiment

BPMN formalism in AToMPM

- Abstract syntax meta-model and concrete visual syntax
- Control flow perspective of BPMN
- “Well formed BPMN Process”
- Syntactic constraints imposed
  - A start event has just one outgoing but no incoming sequence flow ;
  - An end event has just one incoming but no outgoing sequence flow;
  - Activities and intermediate events have exactly one incoming and outgoing sequence flows;
  - Fork or decision gateways have one incoming and more than one outgoing sequence flows;
  - Join or merge gateways have one outgoing and more than one incoming sequence flows; And
  - An error event has no incoming and outgoing sequence flow.
- Concrete visual syntax uses coloring, shapes and textual data - (Dual Coding)

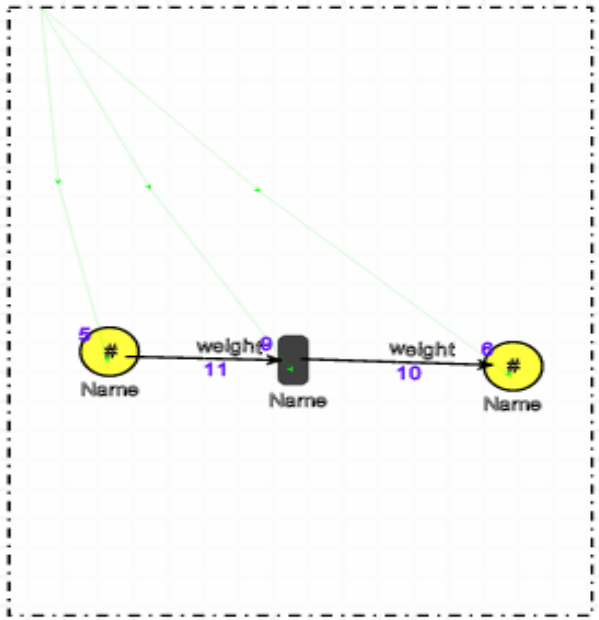

FlowElement
+ name : string
MessageFlow
Pool
FlowNode
SequenceFlow
+ isImmediate : boolean
EndEvent
ExclusiveDecision
ExclusiveMerge
Activity
Event
ThrowEvent
MsgThrow
Gateway
Task
SequentialLoop
+ isWhile-Do : boolean
+ isDo-While : boolean
CatchEvent
StartEvent
ParallelJoin
ParallelFork
ConcurrentLoop
+ counter : int
MsgCatch
Error
Timer
StartMsg
ExceptionFlow
Universiteit
Antwerpen

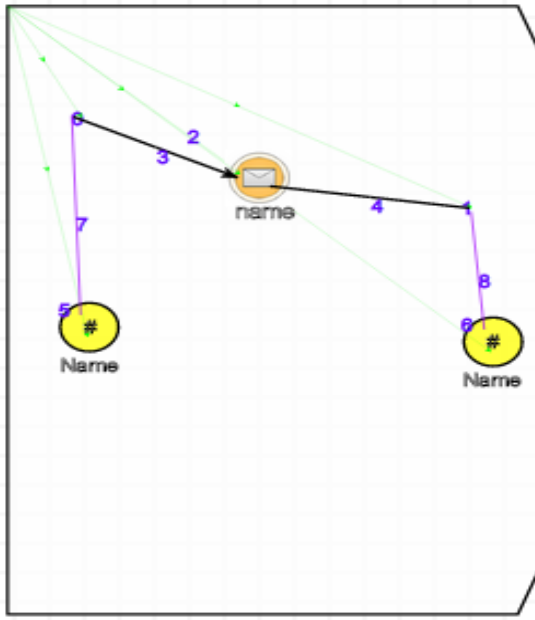
Experiment: *Mapping BPMN onto Petri Nets- CONNECTION FLOWS*

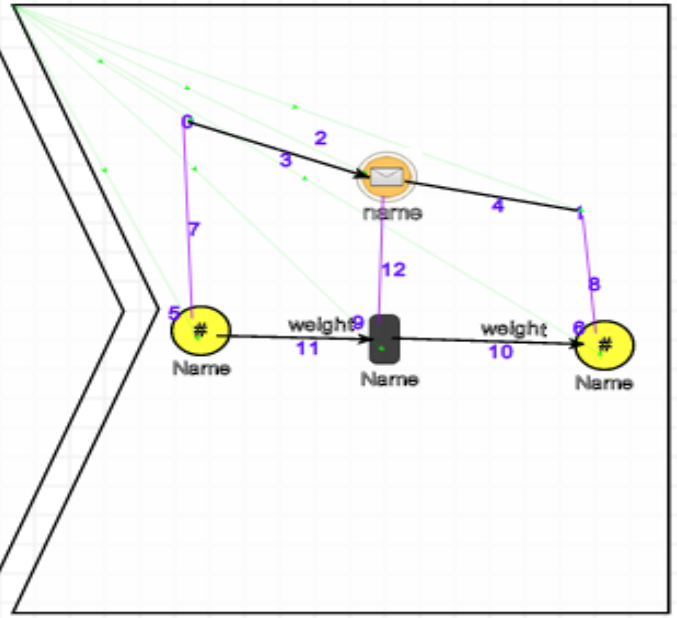


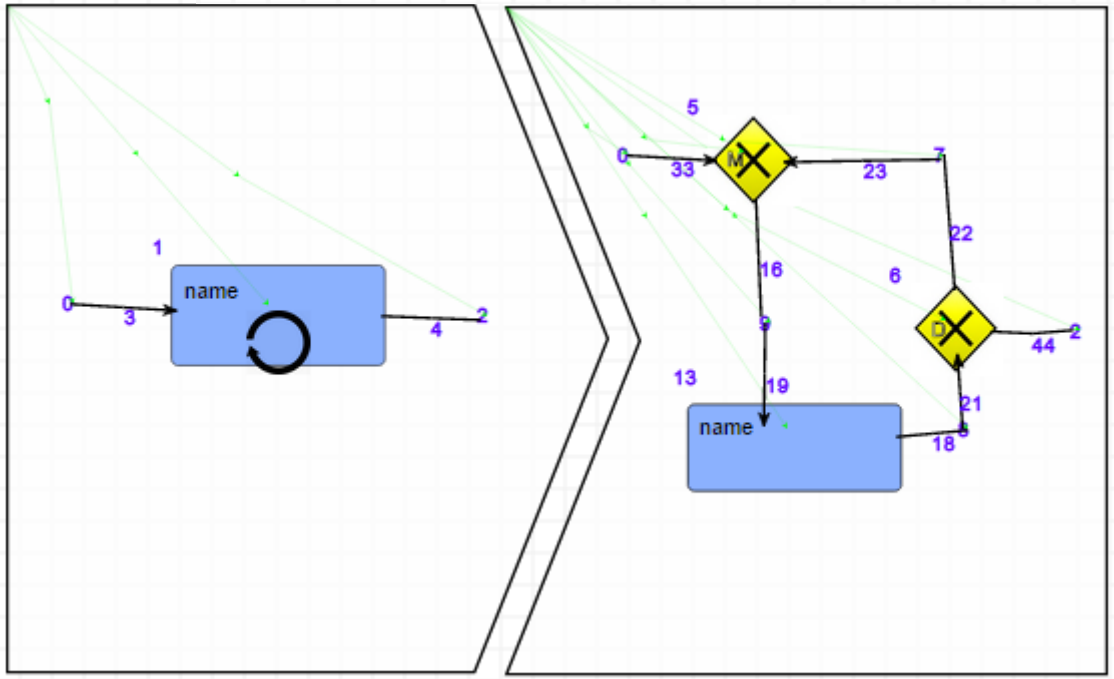
Universiteit Antwerpen

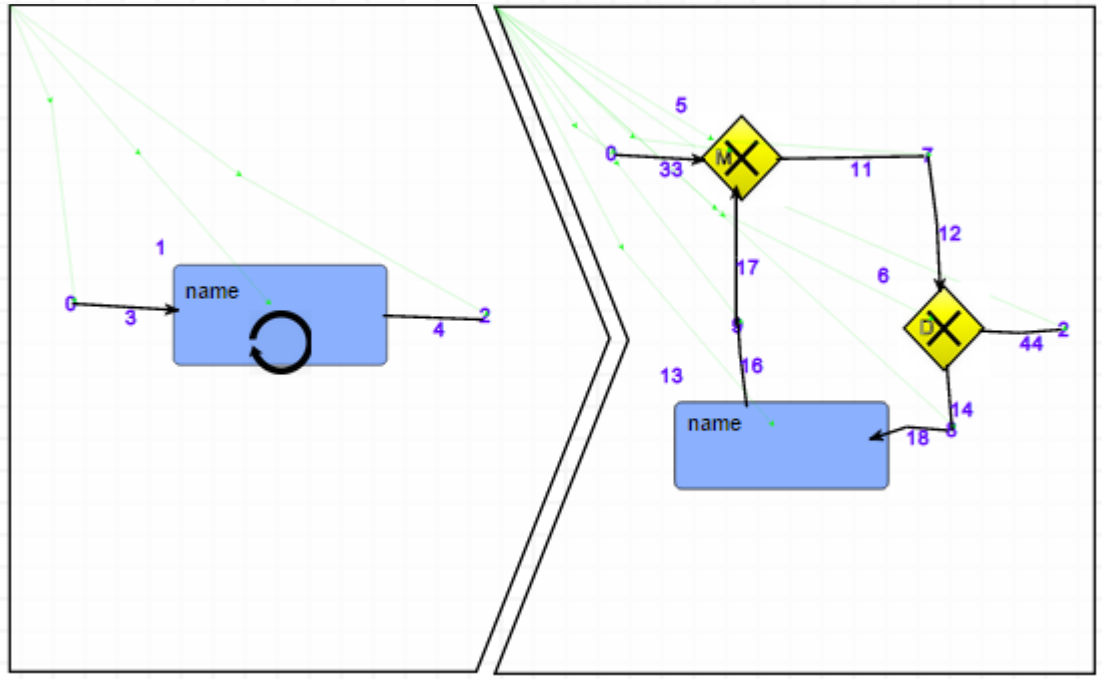
Experiment: *Mapping BPMN onto Petri Nets- EVENTS and TASKS*





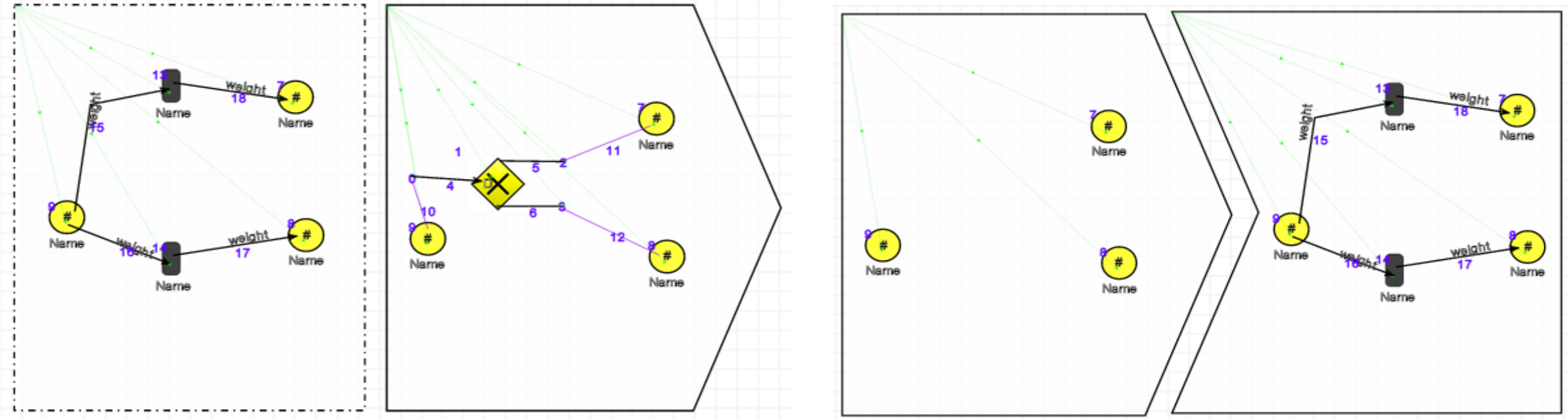






Universiteit Antwerpen

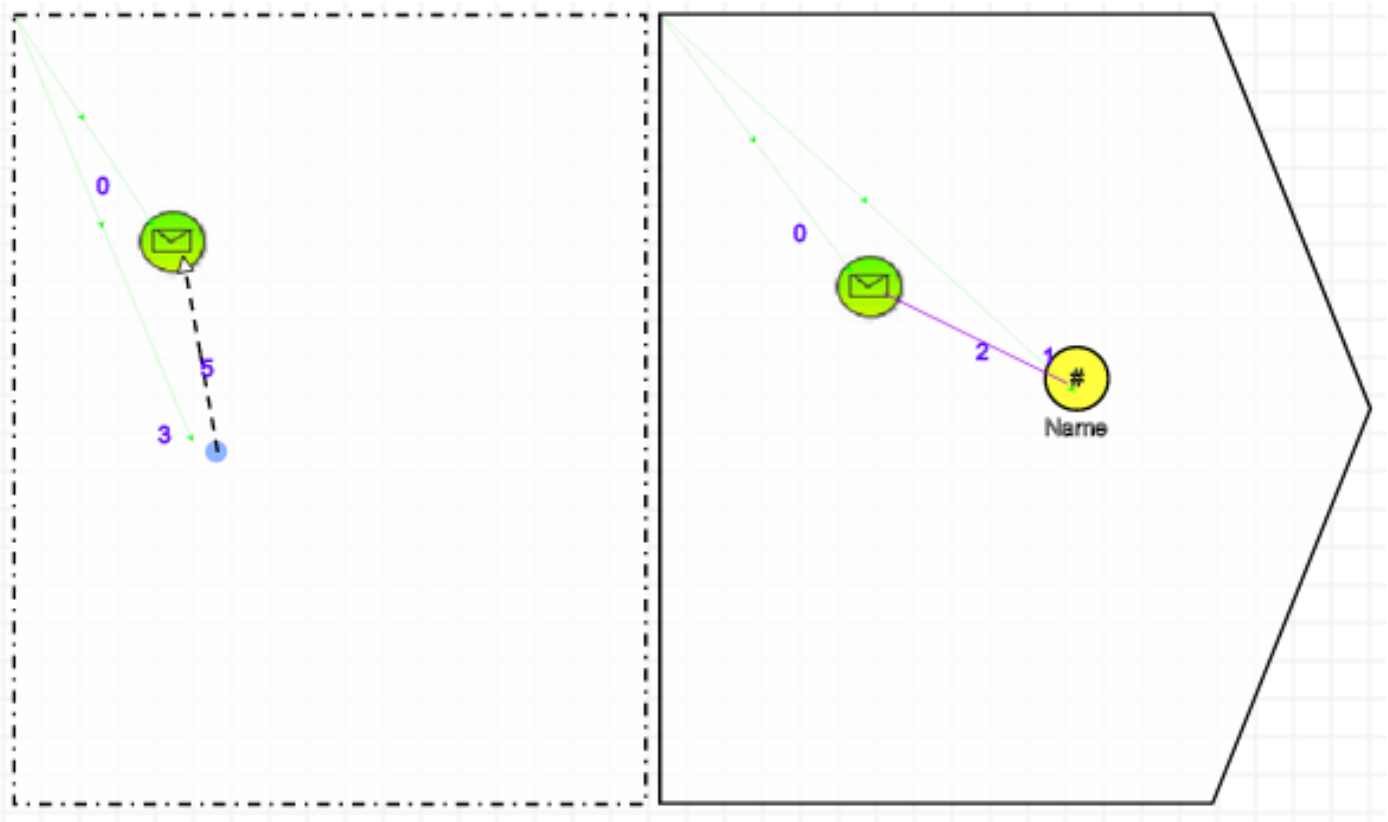
Experiment: *Mapping BPMN onto Petri Nets- GATEWAYS and EXCEPTION*

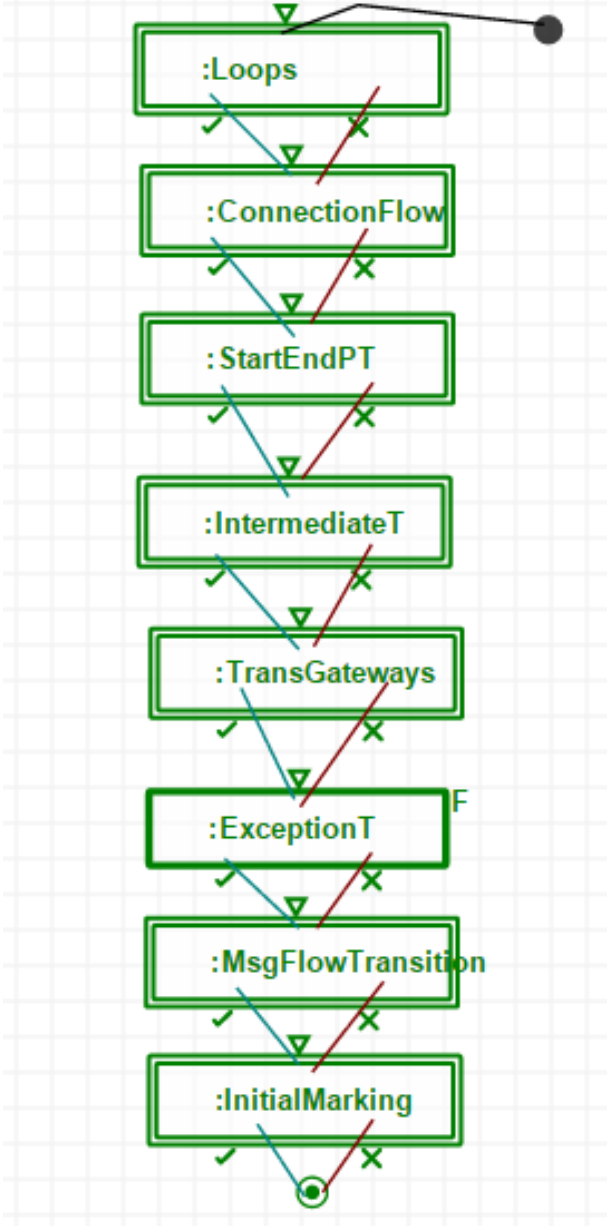


Universiteit Antwerpen

Experiment:

Mapping BPMN onto Petri Nets- Initial place configuration and Schedule





Universiteit Antwerpen

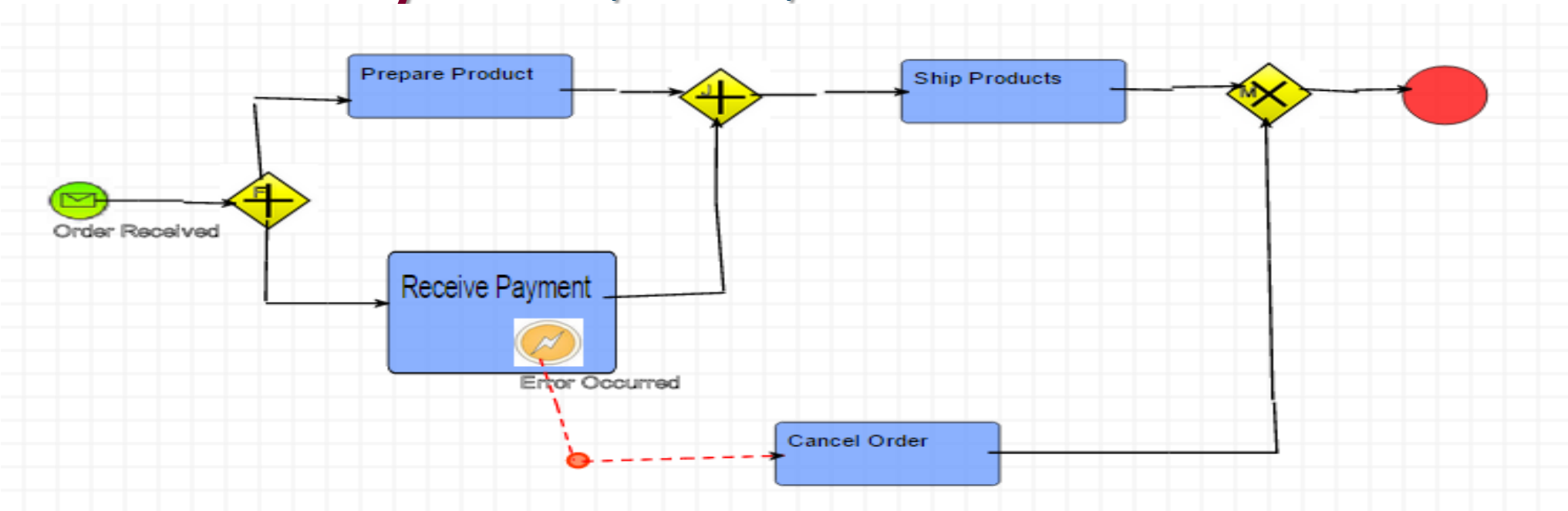
BPMN Model Analysis

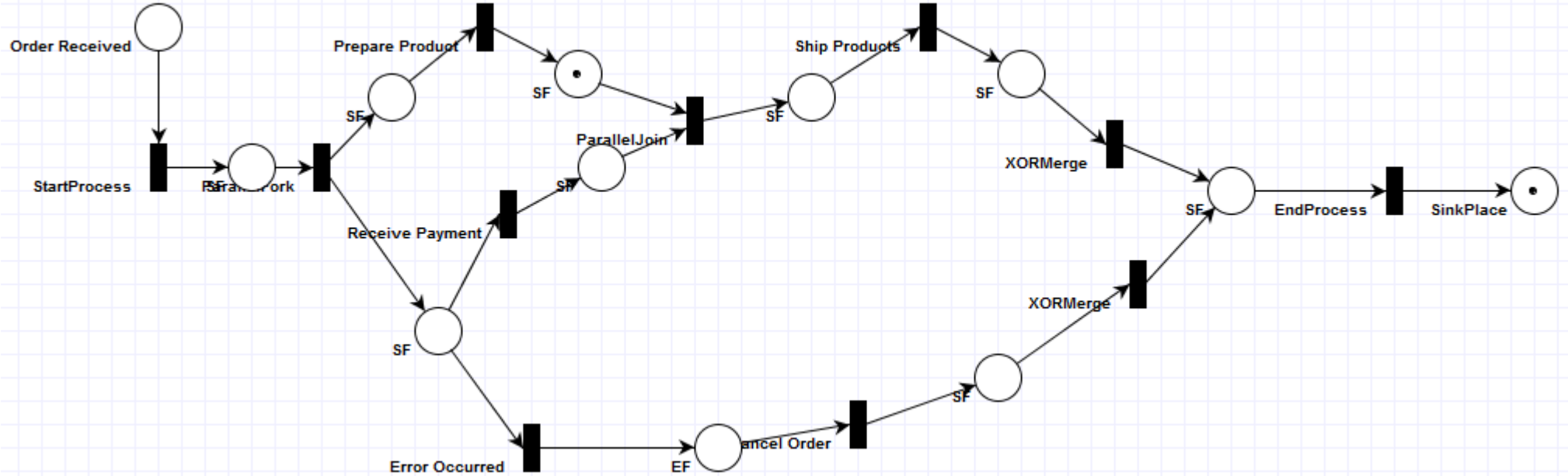
Reachability Analysis

- **Absence of Dead Tasks**: There are no tasks that can never be performed within a model.
- **Proper Completion**: At least one of the end tasks has been executed at least once, and there is no other enabled task for that process instance.

Universiteit Antwerpen

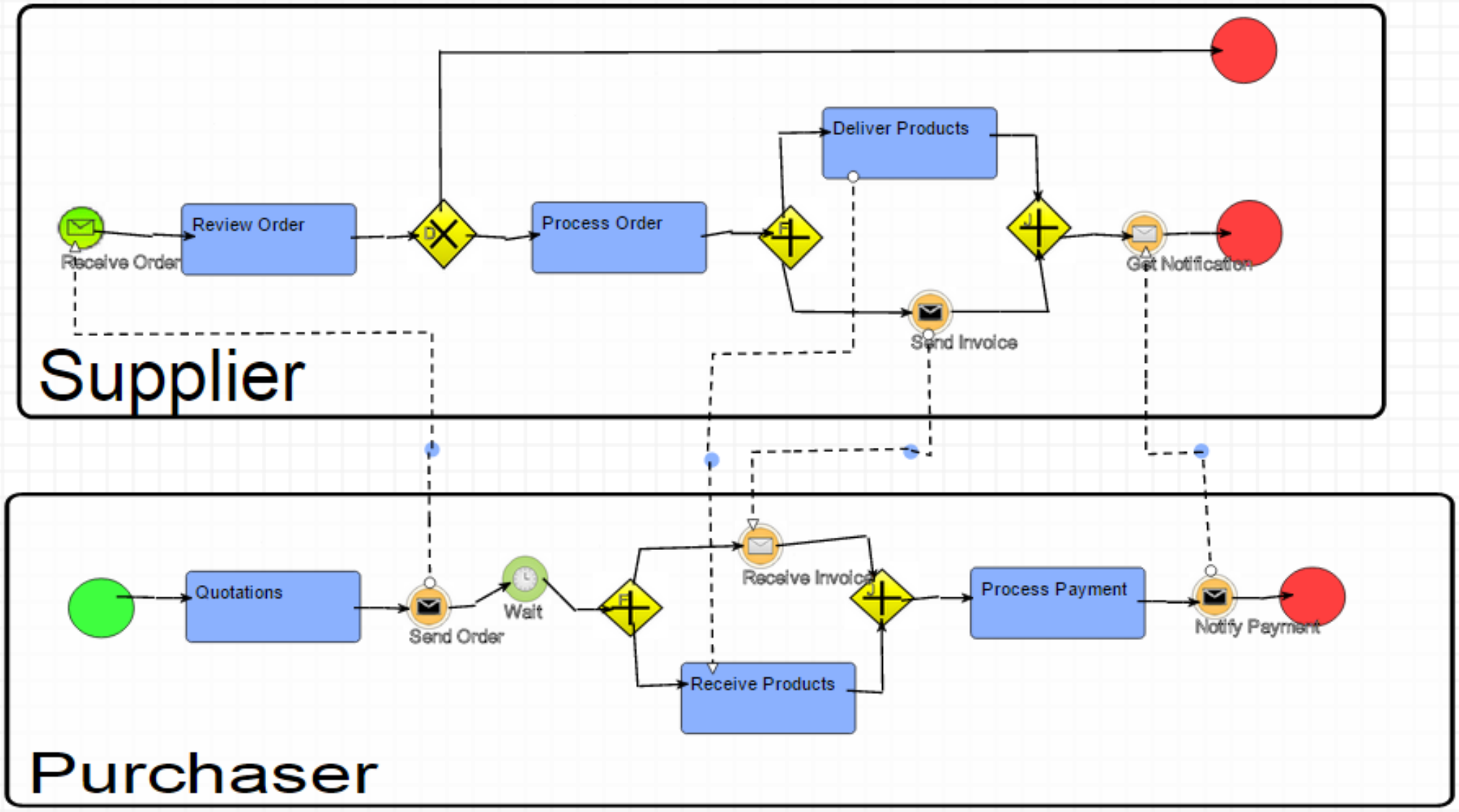
Model Analysis: *Proper Completion*





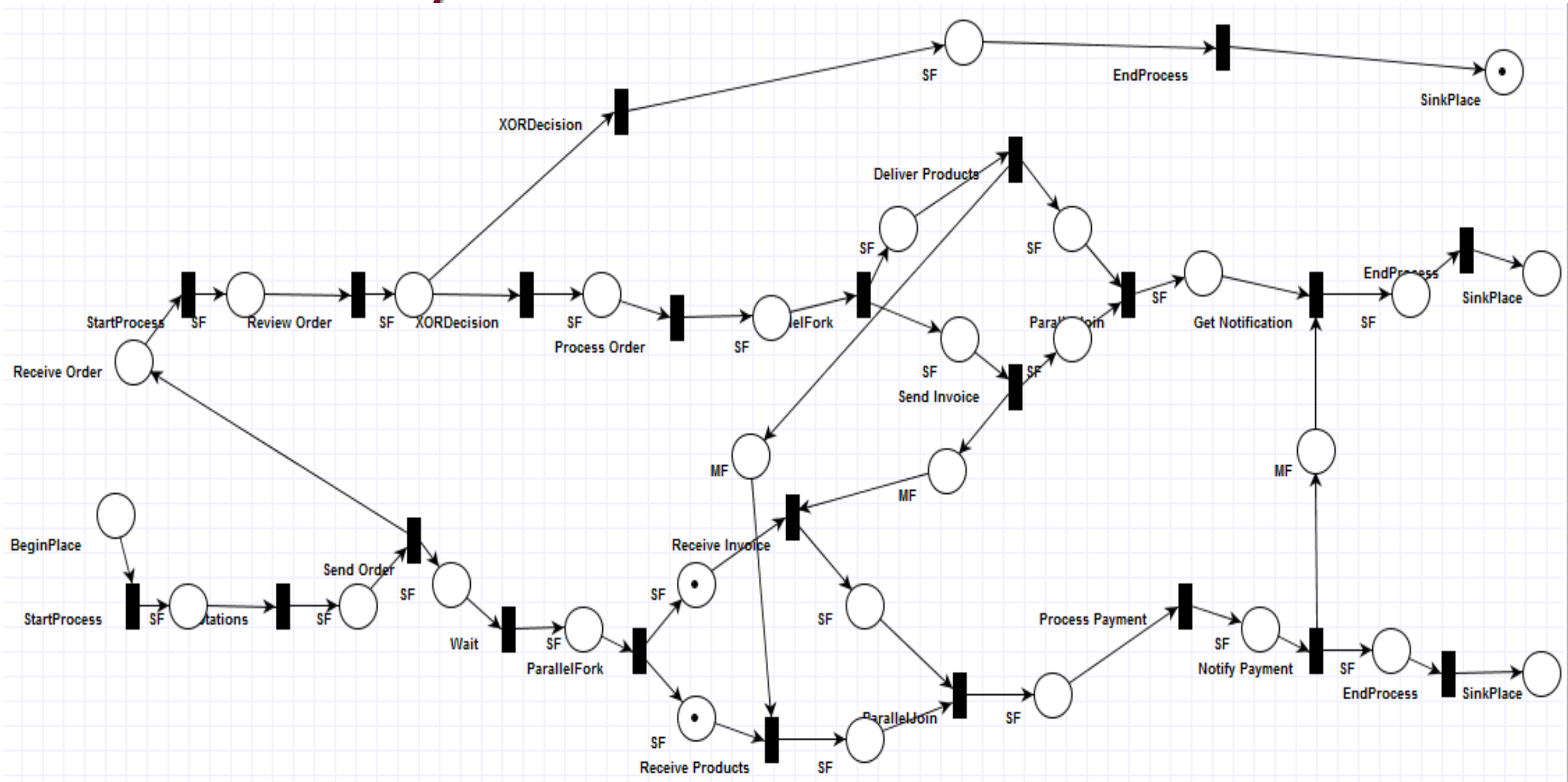
Universiteit Antwerpen

Model Analysis: *Dead Tasks*

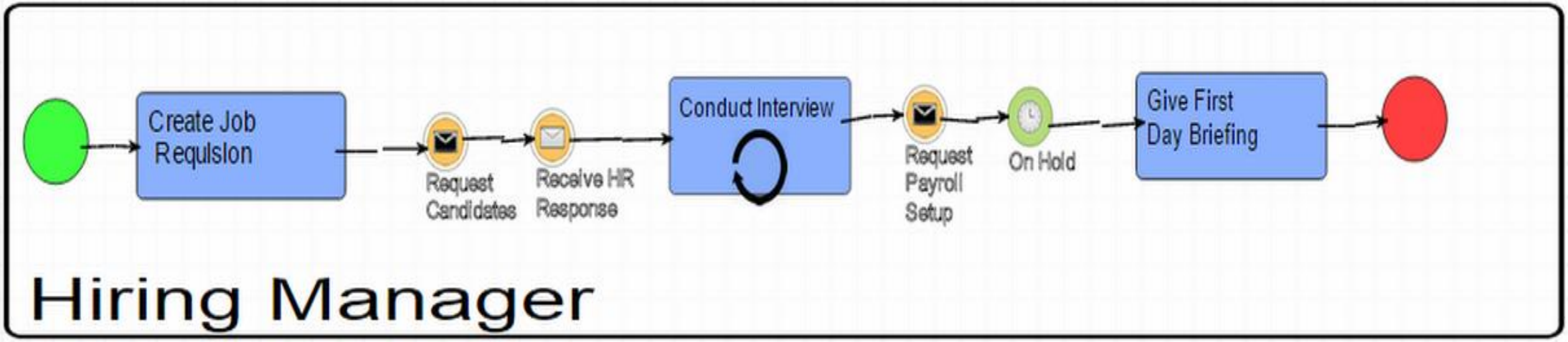


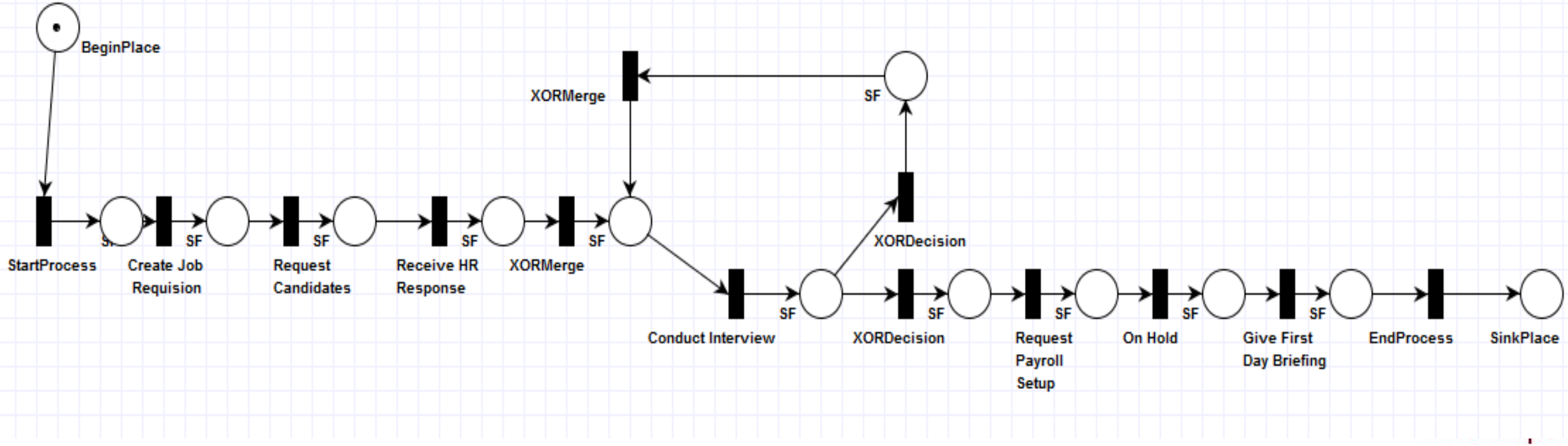
Universiteit Antwerpen

Model Analysis: *Dead Tasks*



Model Analysis: *Sequential loop with "Do-While"*





Conclusion

- Model driven engineering approach to
  - Unveil semantic definition of BPMN
  - Verify semantic correctness of BPMN models
- Efficient in performing model mapping using AToMPM
- Issues with BPMN Specification
  - Process model with multiple start events
  - When should an execution of a process model be considered to be completed?

R. M. Dijkman, M. Dumas, C. Ouyang, Semantics and analysis of business process models in BPMN, Information and Software Technology 50 (12) (2008) 1281--1294.

--END--

Universiteit Antwerpen