

Automated Refinement of Business Processes through Model Transformations Specifying Business Rules

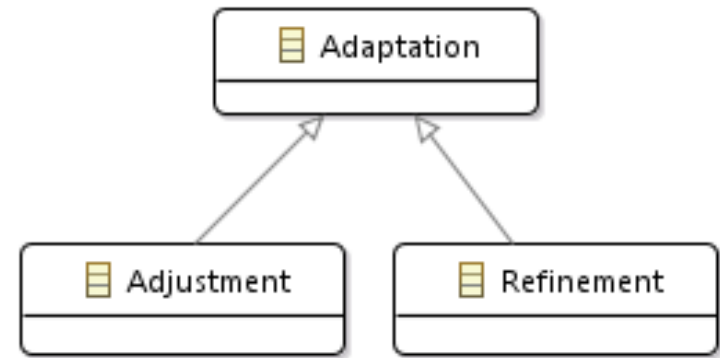
Roman Popp and **Hermann Kaindl**

Outline

- Introduction
- Business Rules Specified as Model Transformation Rules
- Automatically Adapting Reference Processes Through Model Transformations
- Discussion
- Conclusion and Future Work

Introduction

- Changes of business processes
- Business rules for defining variability
- Automated adaptation of reference processes
 - *Adjustment* of concrete reference processes
 - *Refinement* of high-level reference processes
- *Adaptation* of business process models (represented in BPMN 2.0) based on model transformations
- Business rules as model-transformation rules



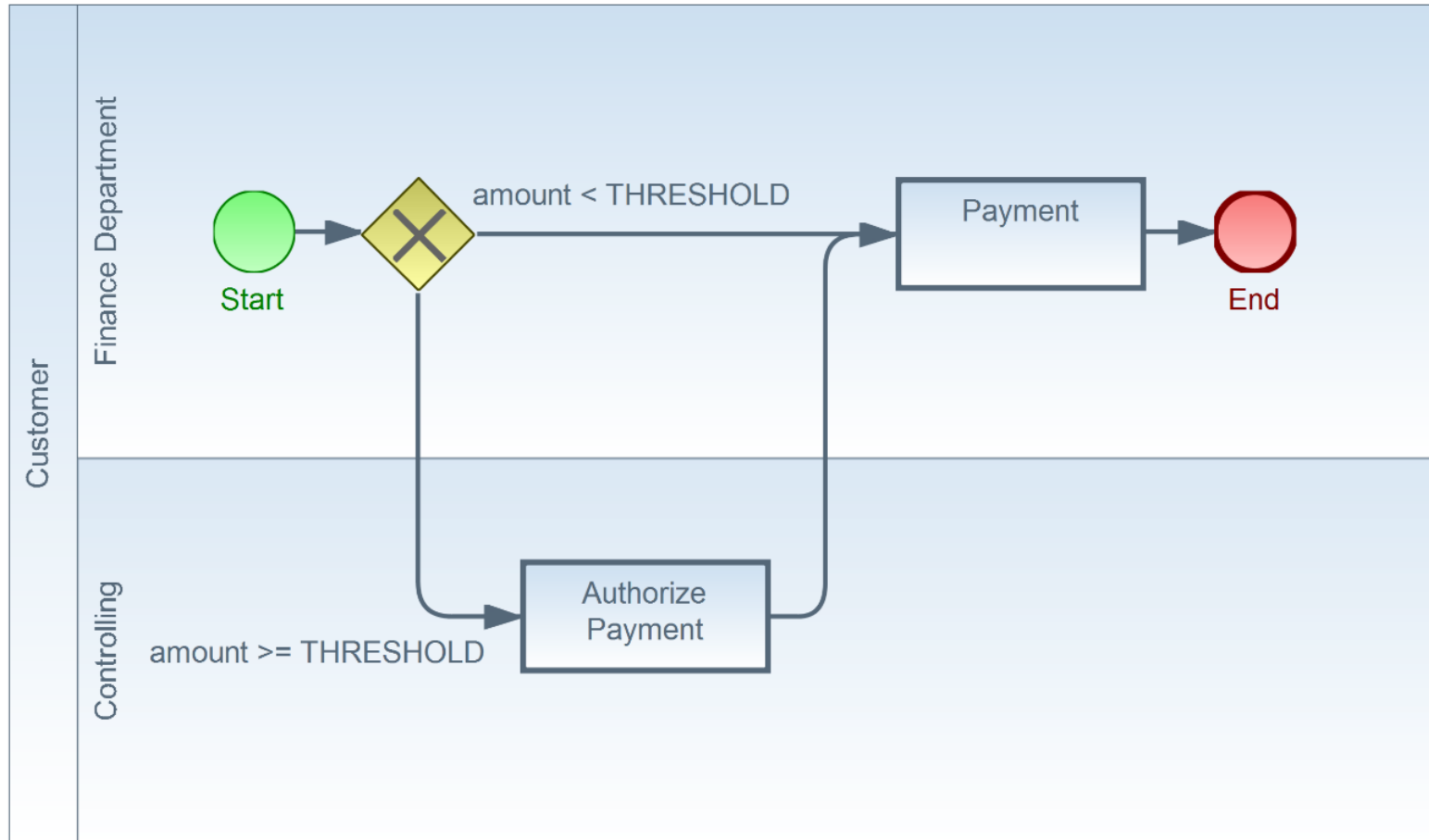
Business Rules Specified as Model Transformation Rules

- As independent from any reference process as possible
- Example:
 - If the amount to be paid exceeds a defined threshold, another business actor than the one primarily responsible for executing a payment activity needs to authorize the payment before its execution.
 - Can be formalized in many ways



Business rule example with Sub-Process creation

Sub-Process Created by this Rule



Sub-Process Authorized Payment

Sketch of the Sub-Process Business Rule in ATL

```
rule callActivityPayment {  
  from  
    p : BPMN2!CallActivity (p.name ='Payment')  
  to  
    end : BPMN2!SubProcess(  
      incoming <- p.incoming,  
      outgoing <- p.outgoing,  
      flowElement <- Set{start,a,payment,f1,f2,f3,f4,end}),  
    start : BPMN2!StartEvent(  
      outgoing <- Set{f1},  
      name <- 'start'),  
    gateway : BPMN2!ExclusiveGateway(  
      categoryValueRef<- p.categoryValueRef,  
      default <- p.default,  
      gatewayDirection<- p.gatewayDirection,  
      id <- p.id,  
      name <- p.name,  
      incoming <- Set{f1},  
      outgoing <- Set{f2,f3}),  
  ...  
}
```

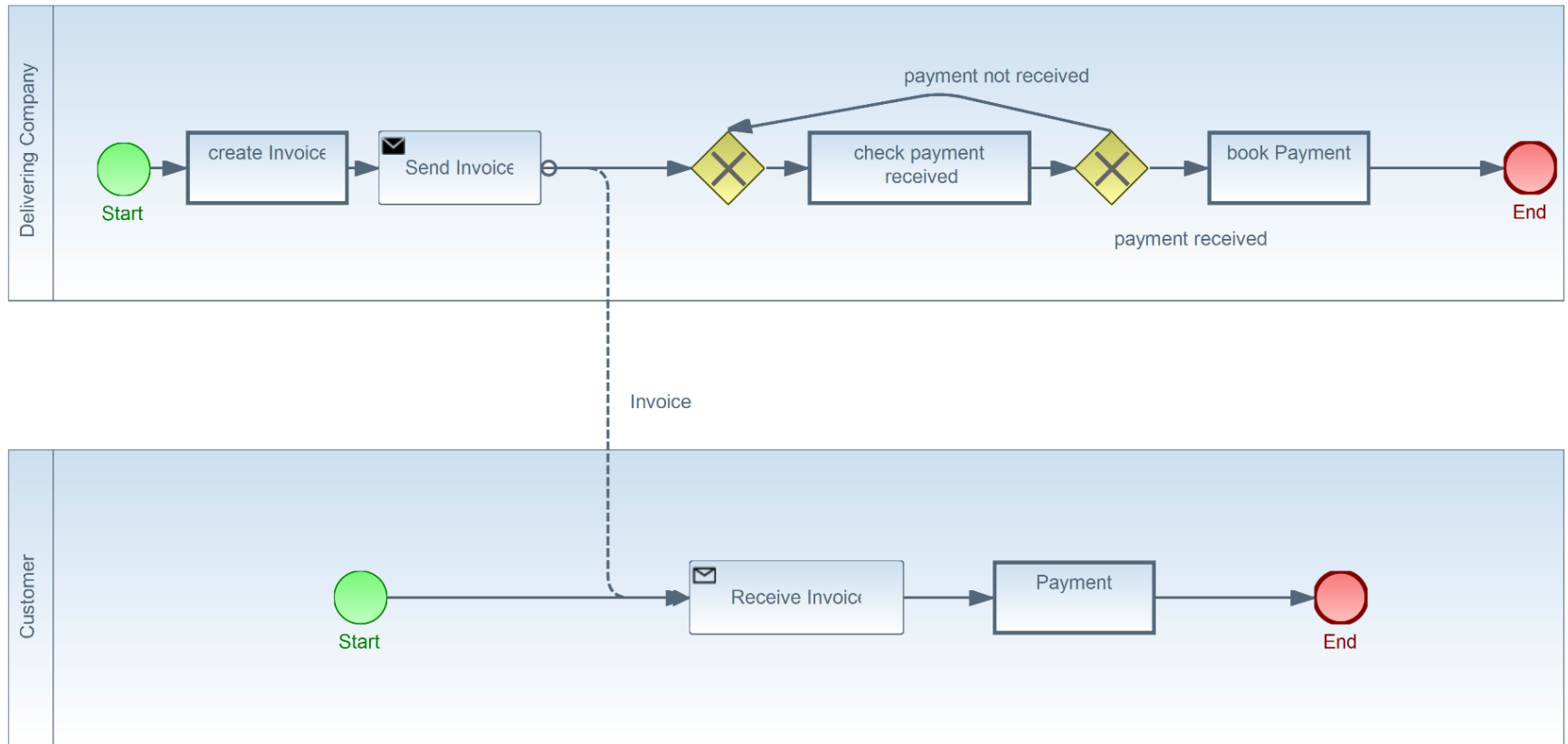


Automatically Adapting Reference Processes Through Model Transformations

- Reference process represented in BPMN
- Technical problems
 - Creation of new model rather than change of existing one
 - ATL, e.g., only allows triggering of exactly one rule per element.
- Adjustment or refinement?



Reference Process



Discussion

- *Abstraction* as the inverse of refinement
- Graph-matching
- ‘Ports’: one entry point and one exit point
- *Decoupling* of (new) business rules from existing business processes
- Design-time vs. run-time
 - Verification
 - Complementary



Conclusion and Future Work

- Adaptability of business process models
- Novel contributions:
 - Focus on refinement of high-level processes rather than adjustment of specific ones
 - Representation of business rules as model transformations
- Automatic execution of these transformations
- New form of representing and managing variability of business process models
- Future work on studies with people actually working on adaptable business processes

Thank you for your attention!

???

A 'generic' transformation rule

rule Activity_to_Activity

from

s: BPMN2!callActivity

to

last: BPMN2!callActivity(
 name <- s.name,
)