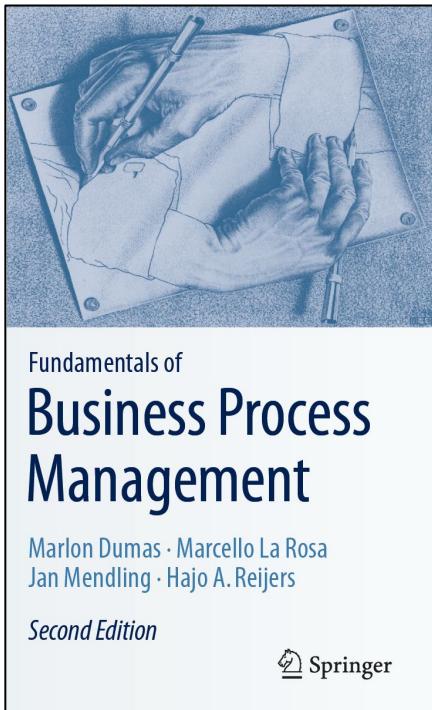




# Chapter 4: Advanced Process Modeling

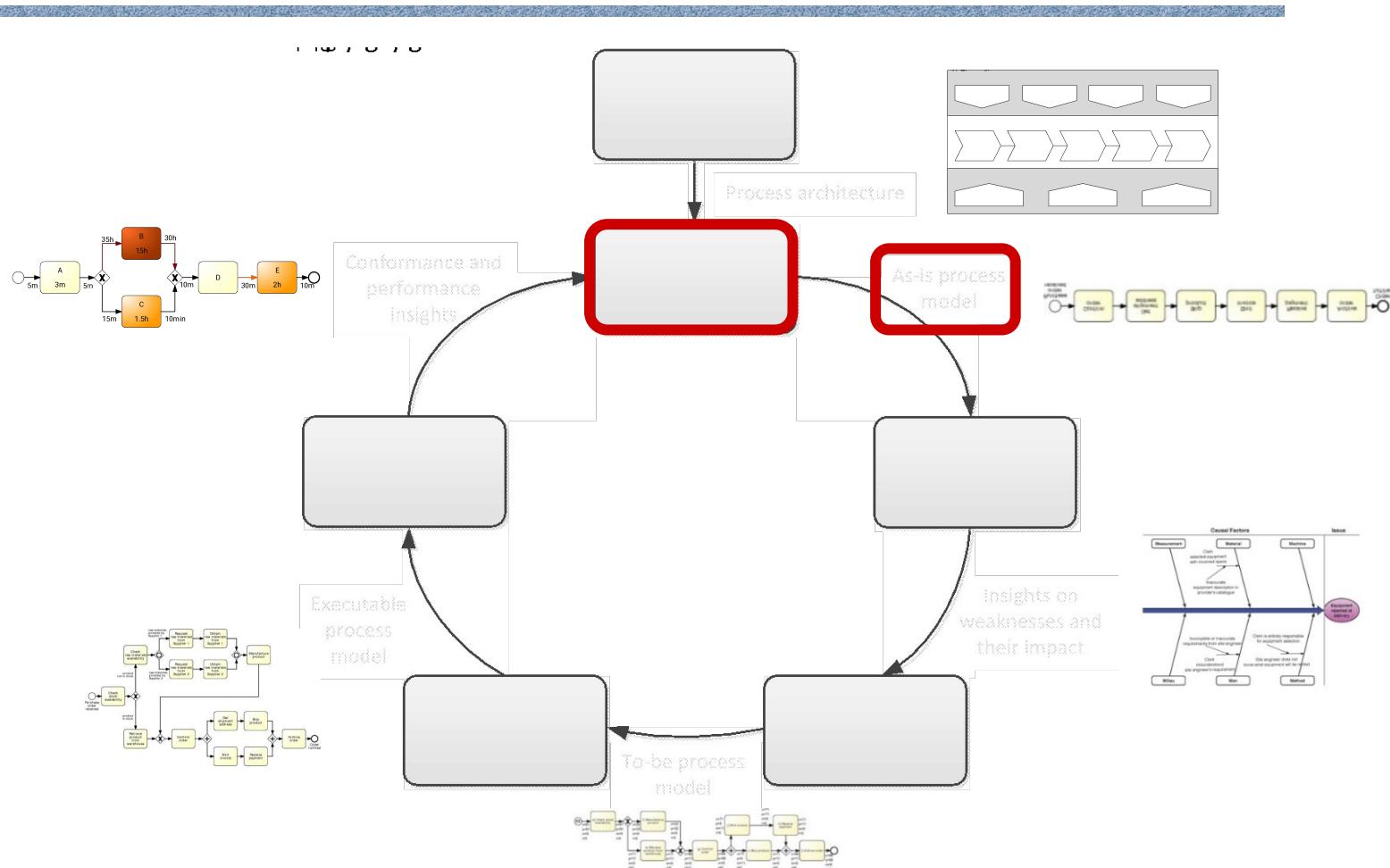


## Contents

1. More on Rework and Repetition
2. Handling Events
3. Handling Exceptions
4. Processes and Business Rules
5. Recap

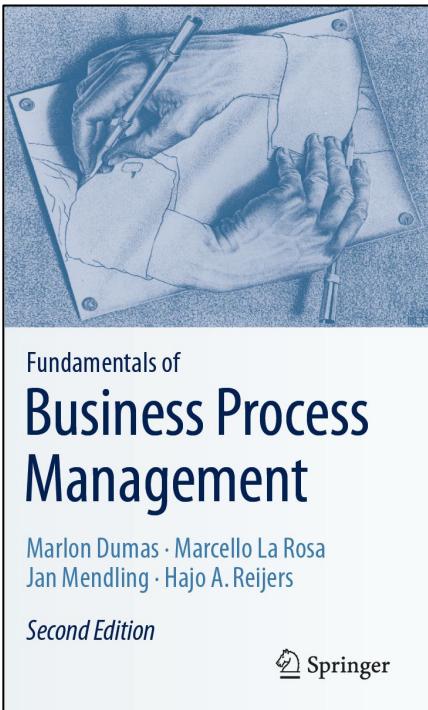


# Process Modeling in the BPM Lifecycle





# Chapter 4: Advanced Process Modeling

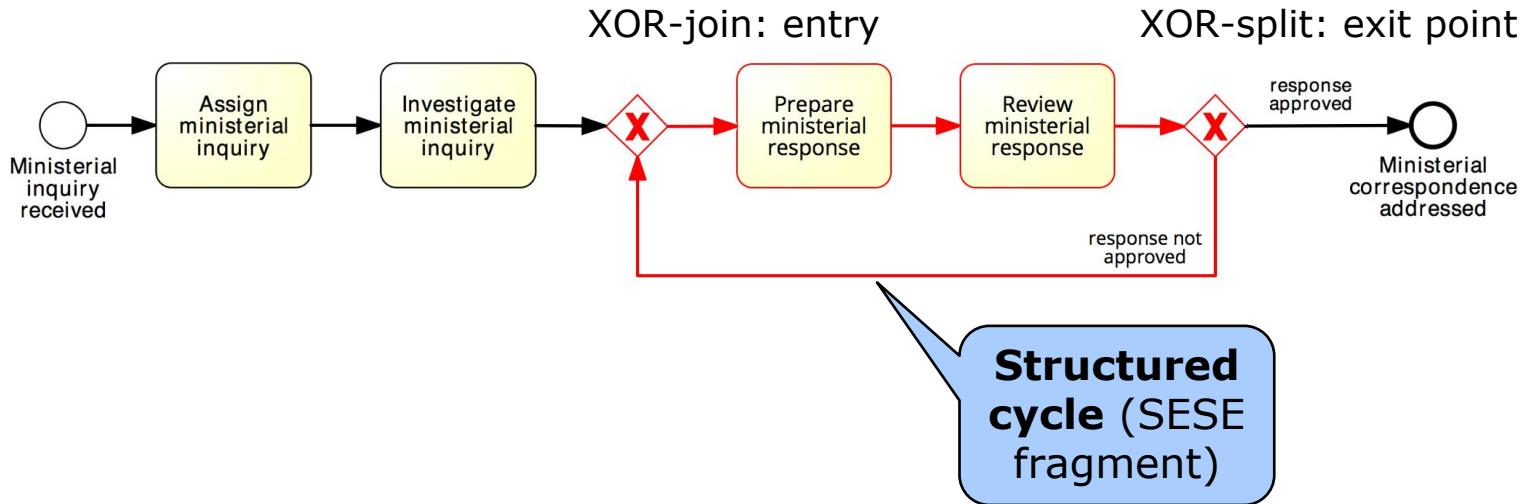


## Contents

1. More on Rework and Repetition
2. Handling Events
3. Handling Exceptions
4. Processes and Business Rules
5. Recap



# More on rework and repetition

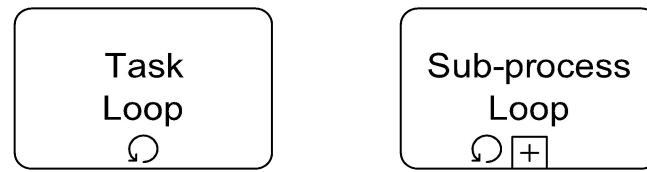


SESE = Single Entry Single Exit fragment, i.e. a fragment delimited by a single entry node and a single exit node (there are no other incoming arcs into the fragment or outgoing arcs from the fragment)



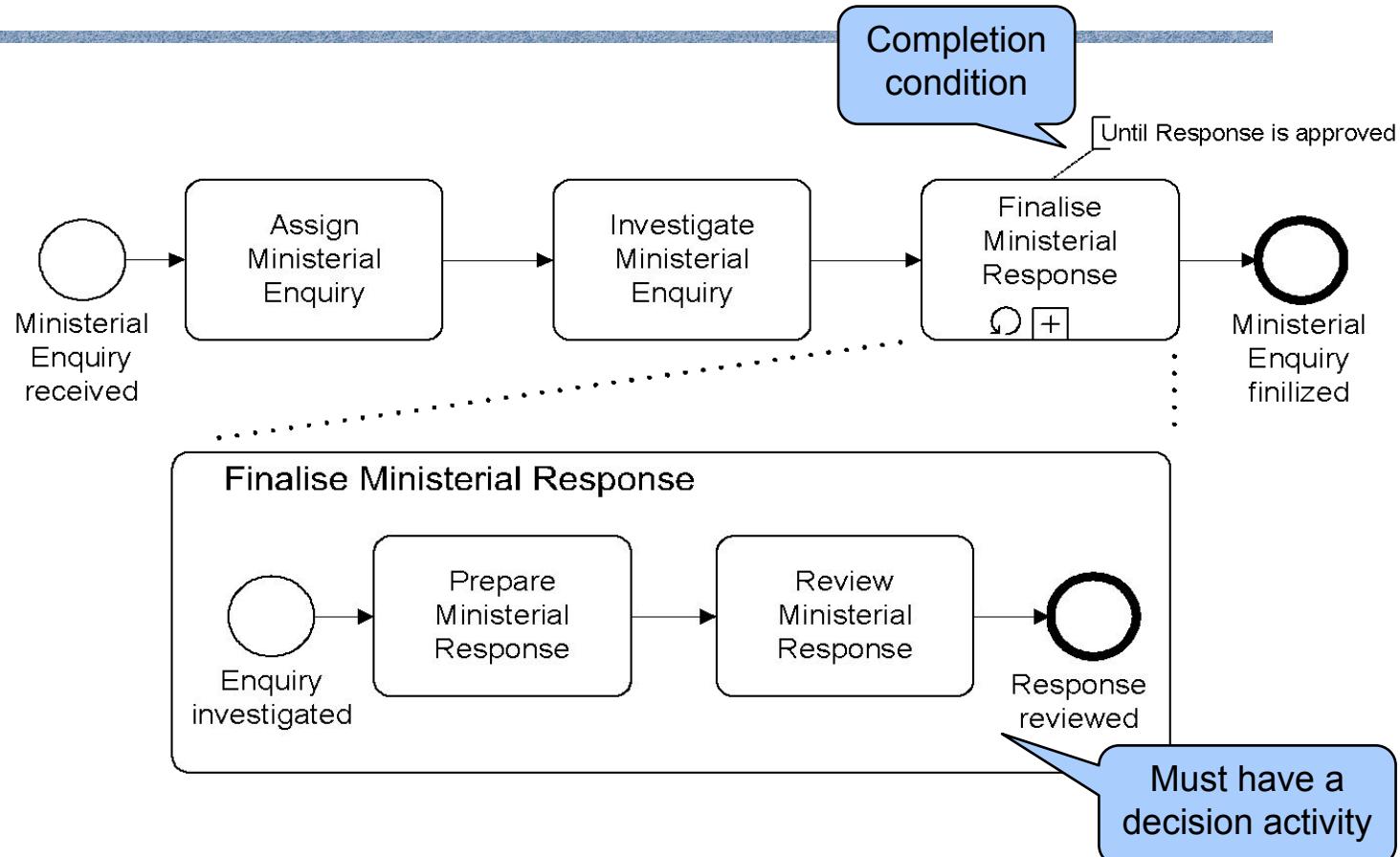
# Block-structured repetition: Loop Activity

BPMN also provides the *loop activity* construct to allow the repetition of a task or sub-process





# Example: block-structured repetition





# Example: multi-instance activity

---

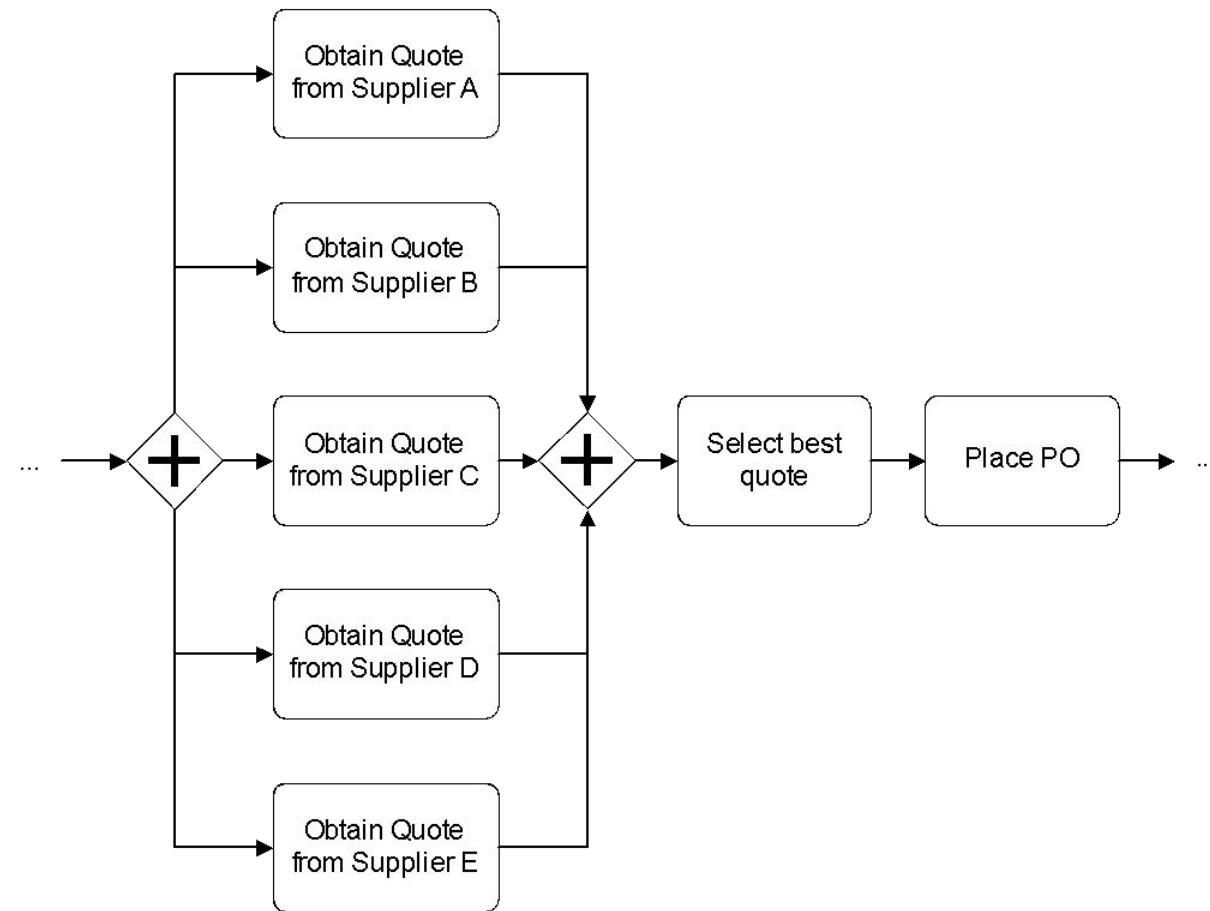
## Procurement

In procurement, typically a quote is to be obtained from all preferred suppliers (assumption: five preferred suppliers exist). After all quotes are received, they are evaluated and the best quote is selected. A corresponding purchase order is then placed.



# Solution: without multi-instance activity

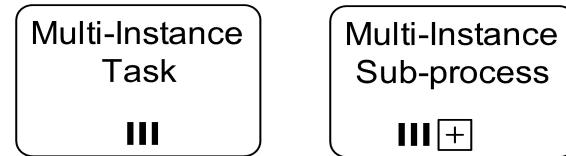
## Procurement





# Parallel repetition: multi-instance activity

Provides a mechanism to indicate that an activity is executed *multiple times concurrently*



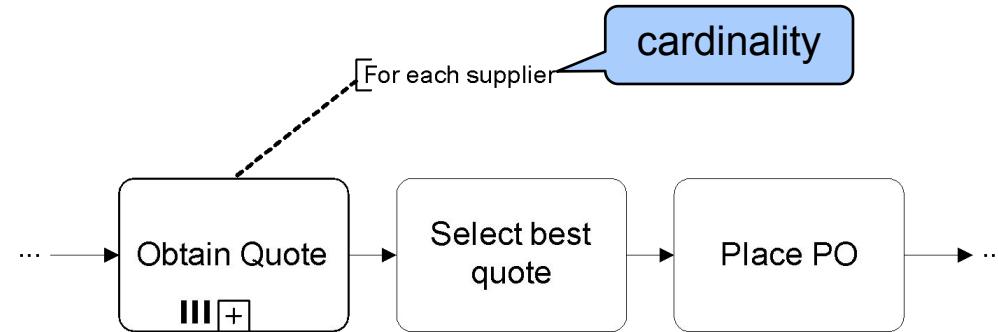
Useful when the same activity needs to be executed for multiple entities or data items, such as:

- Request quotes from multiple suppliers
- Check the availability for each line item in an order separately
- Send and gather questionnaires from multiple witnesses in the context of an insurance claim



# Solution: with multi-instance activity

## Procurement





# Further example: multi-instance activity

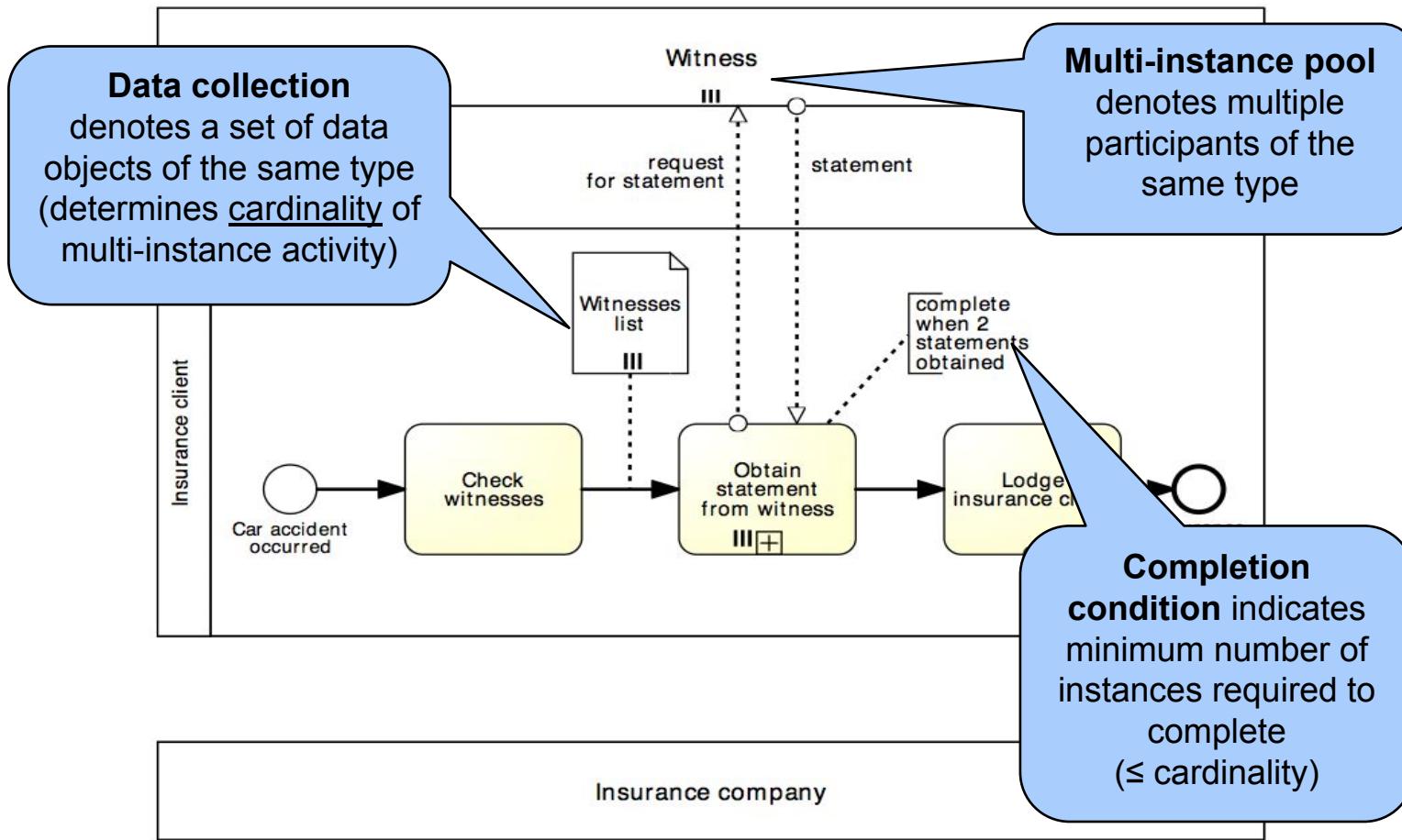
---

## Motor insurance claim lodgement

After a car accident, a statement is sought from the witnesses that were present, in order to lodge the insurance claim. As soon as the first two statements are received, the claim can be lodged to the insurance company without waiting for the other statements.

## Solution: multi-instance activity

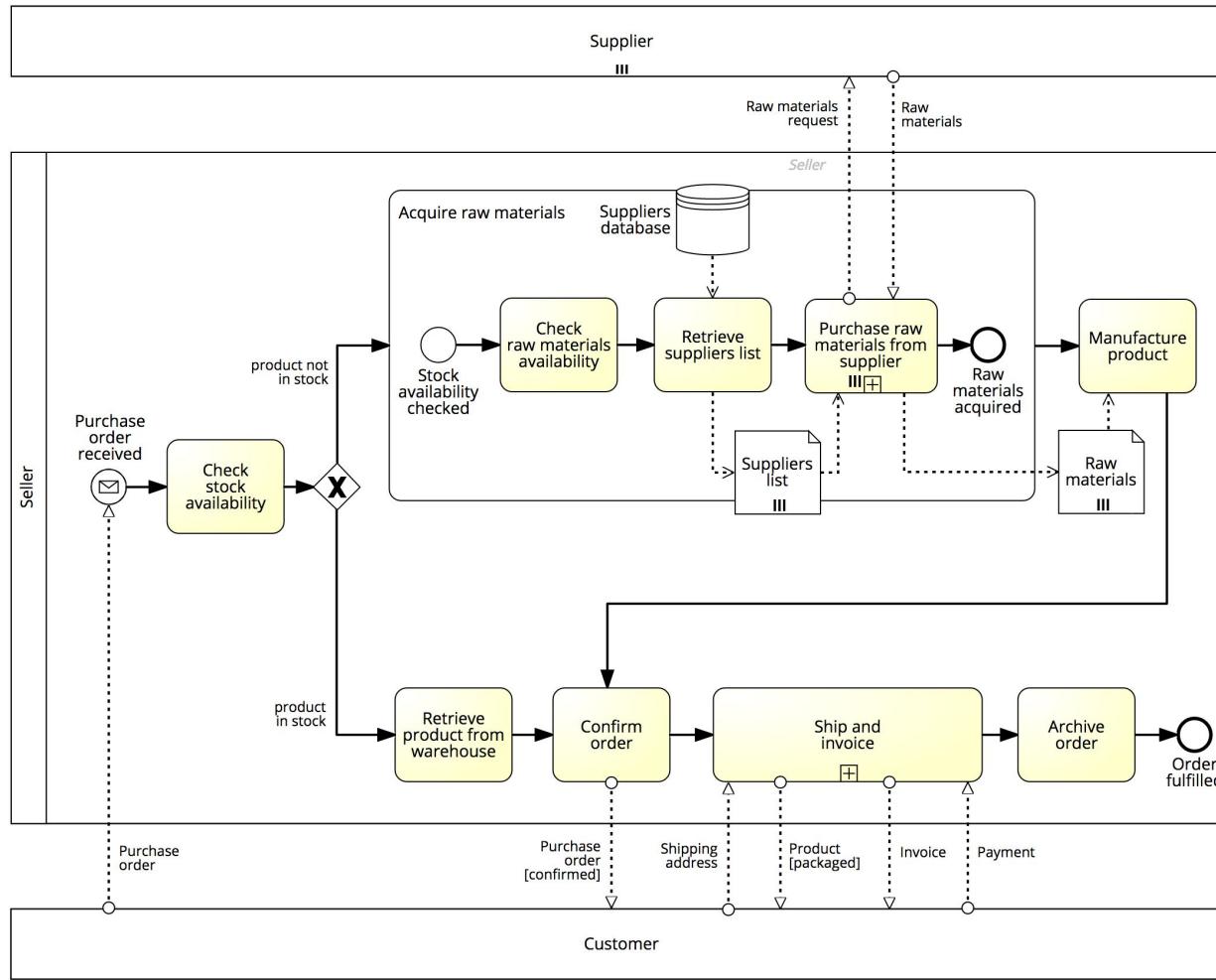
## **Motor insurance claim lodgement**





# Our order-to-cash example...

## now with pools, messages and MI markers



## Exercise 4.2

---

Model the following process fragment.



*After a car accident, a statement is sought from two witnesses out of the five that were present in order to lodge the insurance claim. As soon as the first two statements are received, the claim can be lodged with the insurance company without waiting for the other statements*



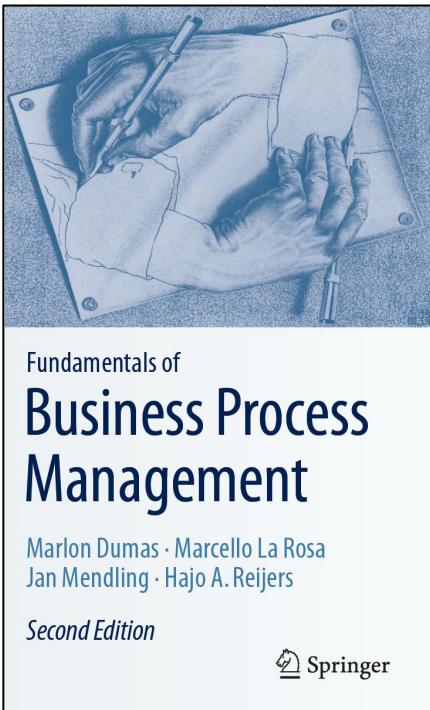
## Exercise 4.3: Army recruitment

Model the following process snippet.

*A typical army selection process starts by shortlisting all candidates' applications. Those shortlisted are then called to sit the following tests: drug and alcohol, eye, color vision, hearing, blood, urine, weight, fingerprinting and doctor examination. The color vision can only be done after the eye test, while the doctor examination can only be done after color vision, hearing, blood, urine and weight have been tested. Moreover, it may be required for some candidates to repeat some of these tests multiple times in order to get a correct assessment, e.g. the blood test may need be repeated if the candidate has taken too much sugar in the previous 24 hours. The candidates that pass all tests are asked to sit a mental exam and a physical exam, followed by an interview. Only those that also pass these two exams and perform well in the interview can be recruited in the army.*



# Chapter 4: Advanced Process Modeling



## Contents

1. More on Rework and Repetition
2. Handling Events
3. Handling Exceptions
4. Processes and Business Rules
5. Recap



# How do we model this scenario?

---

## PO handling

A Purchase Order (PO) handling process starts when a **PO is received**. The PO is first registered. If the current date is not **a working day**, the process waits until the **following working day** before proceeding. Otherwise, an availability check is performed and a **PO response** is sent back to the customer.



# Events handling

In BPMN, events model something instantaneous happening during the execution of a process

They affect the process flow:

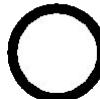
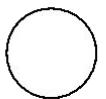
- Start
- Intermediate
- End





# BPMN event types

Start    Intermediate    End

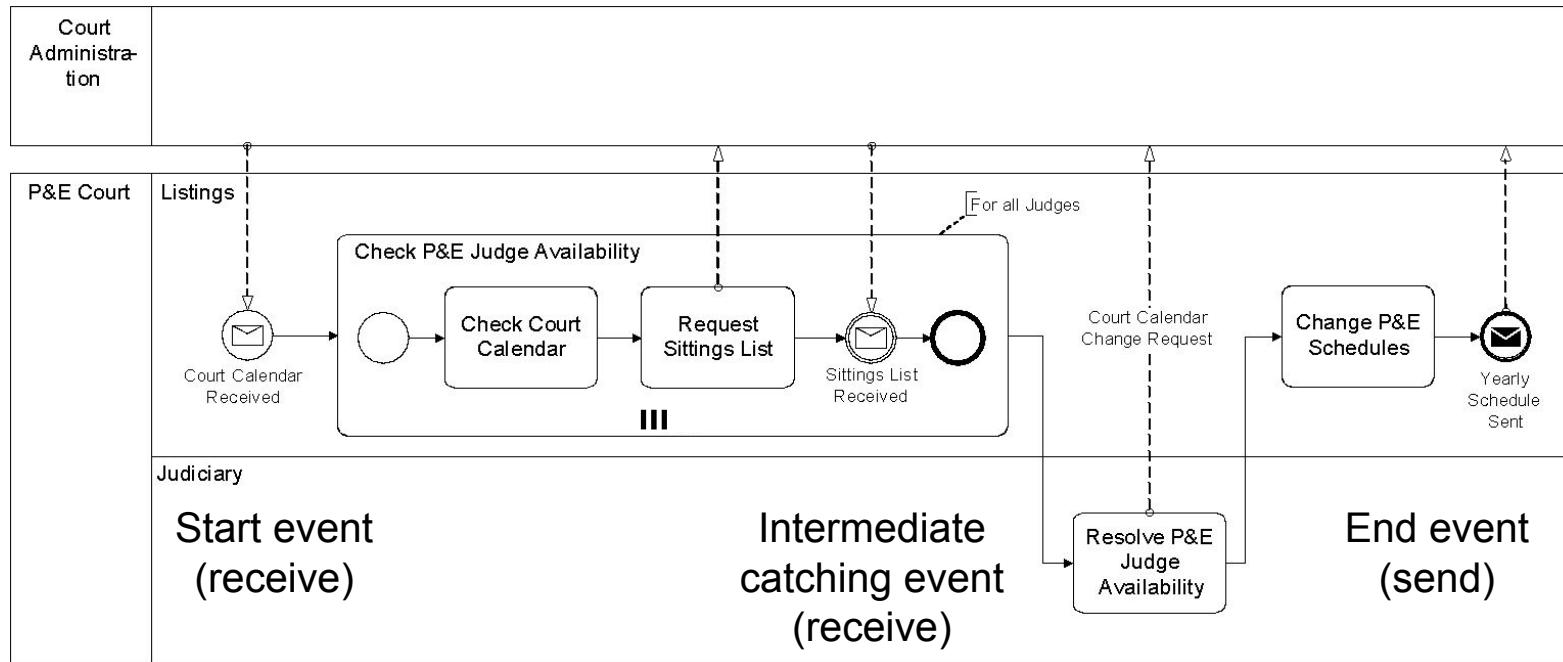


**Untyped Event** – Indicates that an instance of the process is created (start) or completed (end), without specifying the cause for creation/completion

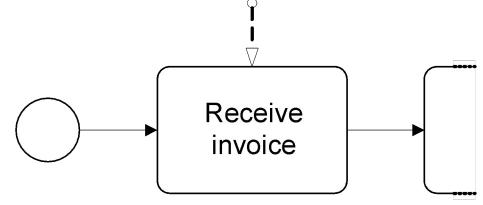
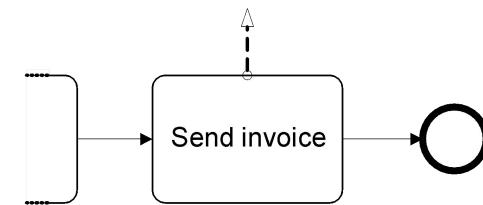
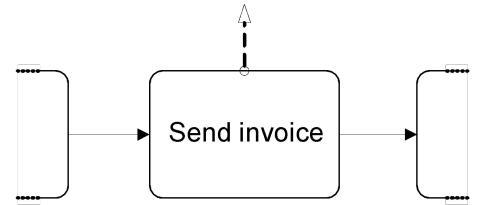
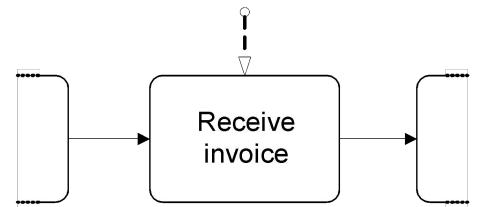
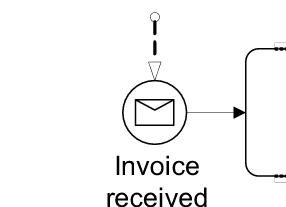
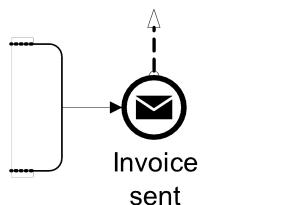
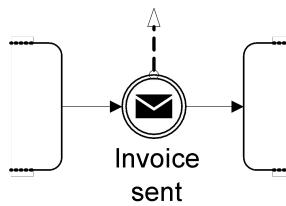
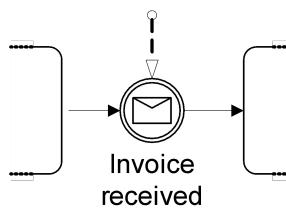
**Start Message Event** – Indicates that an instance of the process is created when a message is received



# Example: message events



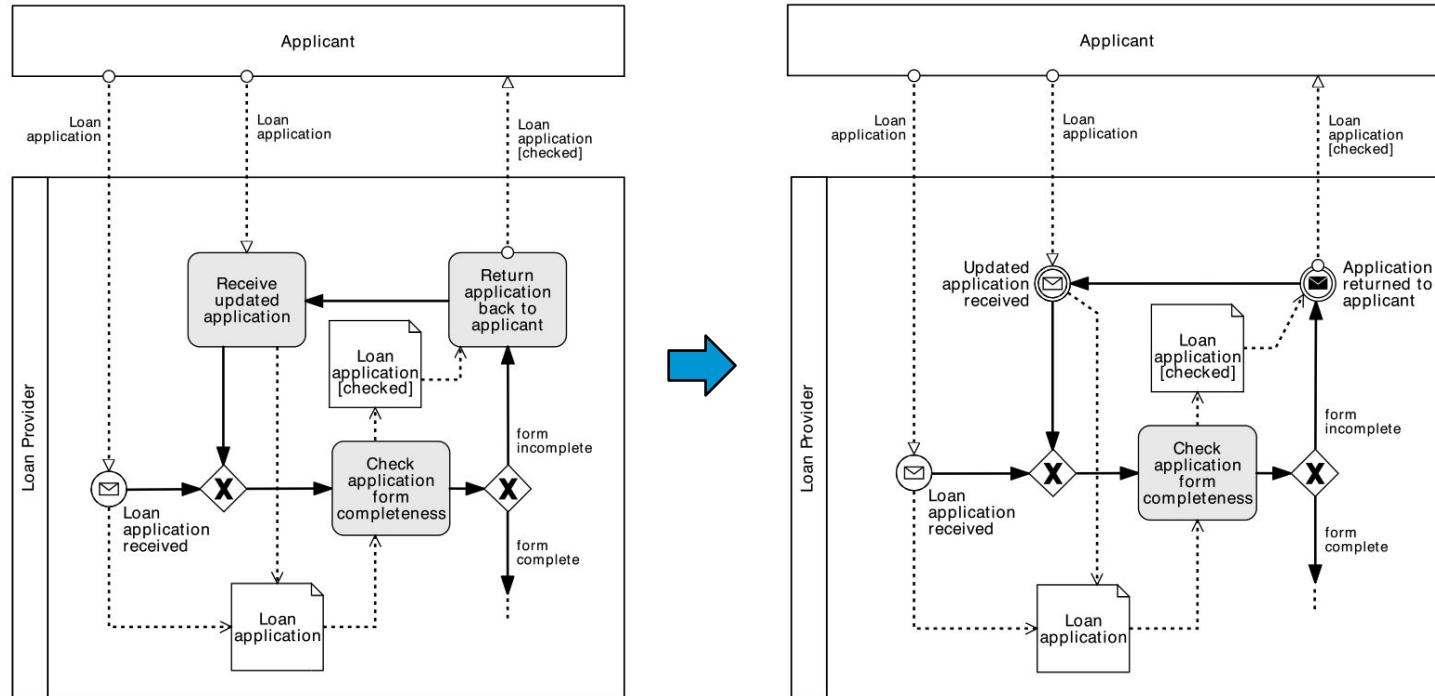
# Comparison with sending/receiving tasks





# So, when to use what?

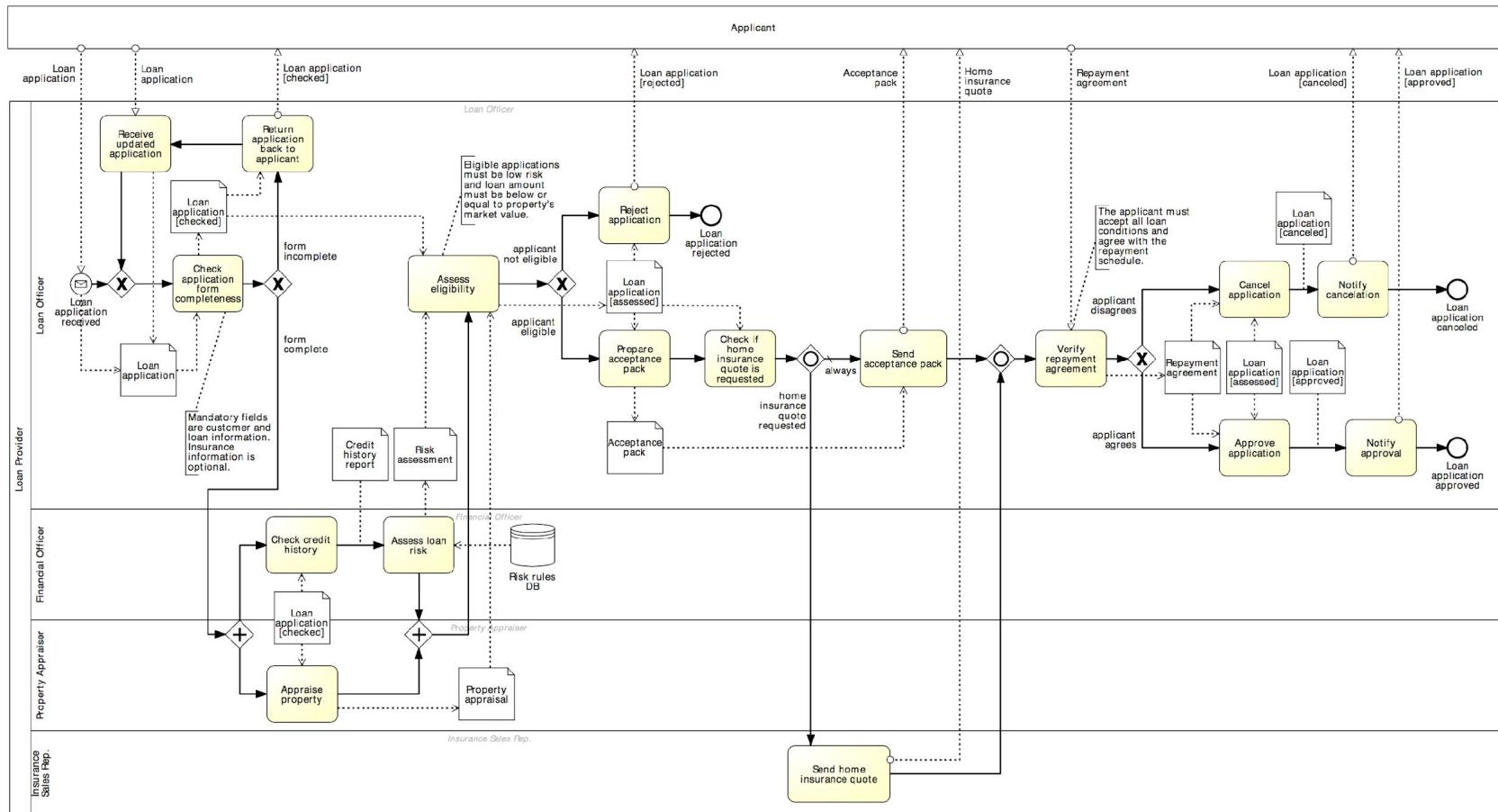
Use message events only when the corresponding activity would simply send or receive a message and do nothing else





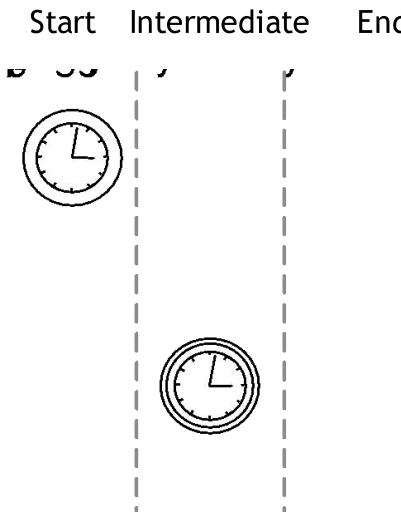
# Exercise 4.4

Is there any other activity in the loan assessment model below that can be replaced by a message event?





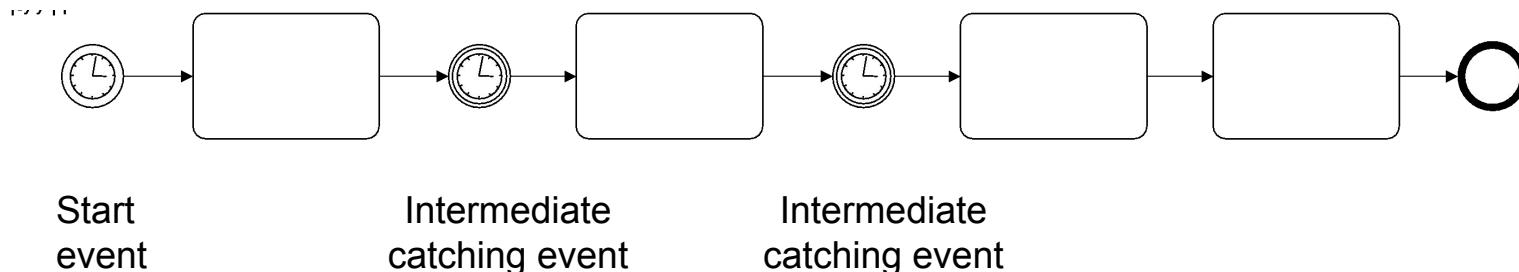
# Temporal events





# Example: temporal events

In a small claims tribunal, callovers occur once a month to set down the matter for the upcoming trials. The process for setting up a callover starts **three weeks prior** to the callover day, with the preparation of the callover list containing information such as contact details of the involved parties and estimated hearing date. **One week prior** to the callover, the involved parties are notified of the callover date. Finally, **on the callover day**, the callover material is prepared and the callover is held.



Start event

Intermediate catching event

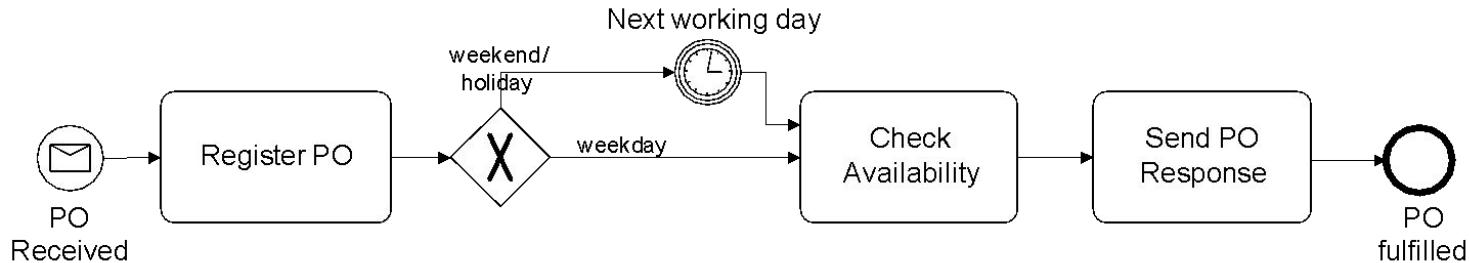
Intermediate catching event



# Coming back to our scenario...

## PO handling

A Purchase Order (PO) handling process starts when a PO is received. The PO is first registered. If the current date is not a working day, the process waits until the following working day before proceeding. Otherwise, an availability check is performed and a PO response is sent back to the customer.





## Exercise 4.4: Internet Service Provider

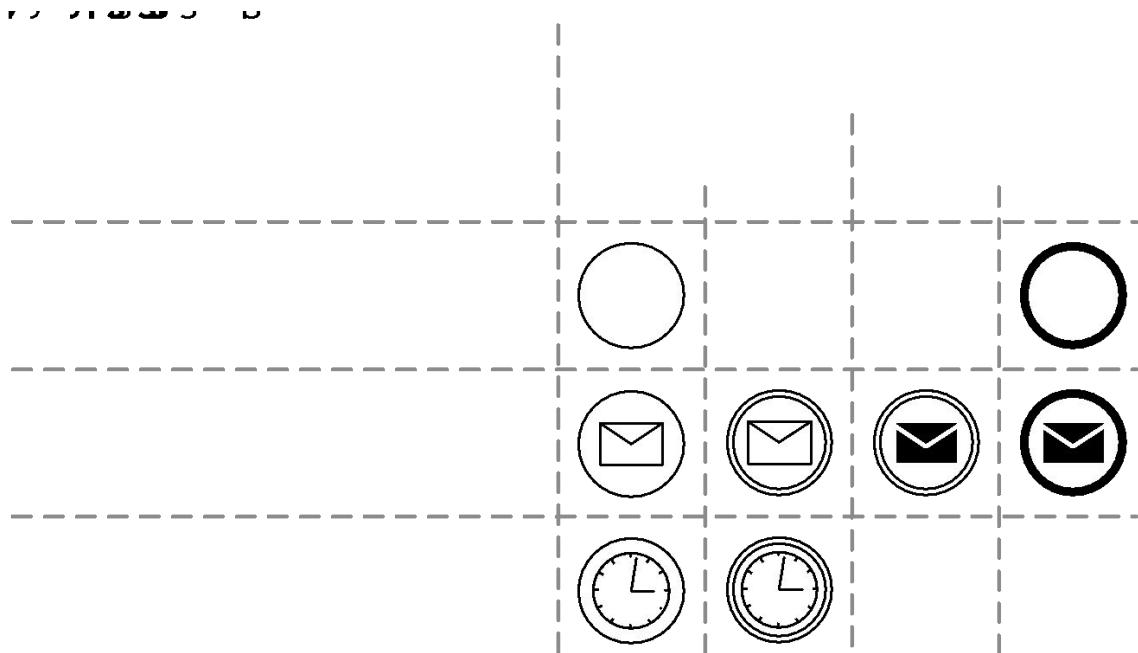
---

Model the billing process of an Internet Service Provider (ISP).

*The ISP sends an invoice by email to the customer on the first working day of each month (Day 1). On Day 7, the customer has the full outstanding amount automatically debited from its bank account. If an automatic transaction fails for any reason, the customer is notified on Day 8. On Day 9, the transaction that failed on Day 7 is re-attempted. If it fails again, on Day 10 a late fee is charged to the customer's bank account. At this stage, the automatic payment is no longer attempted. On Day 14, the Internet service is suspended until payment is received. If the payment is still outstanding on Day 30, the account is closed and a disconnection fee is applied. A debt-recovery procedure is then started.*

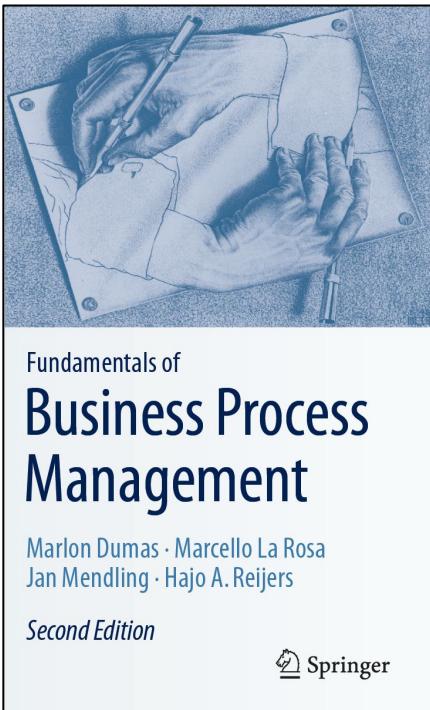


# Recap: message and timer events





# Chapter 4: Advanced Process Modeling



## Contents

1. More on Rework and Repetition
2. Handling Events
3. Handling Exceptions
4. Processes and Business Rules
5. Recap



# Process abortion

Exceptions are events that deviate a process from its “normal” course

The simplest form of exception is to notify that there is an exception (negative outcome)

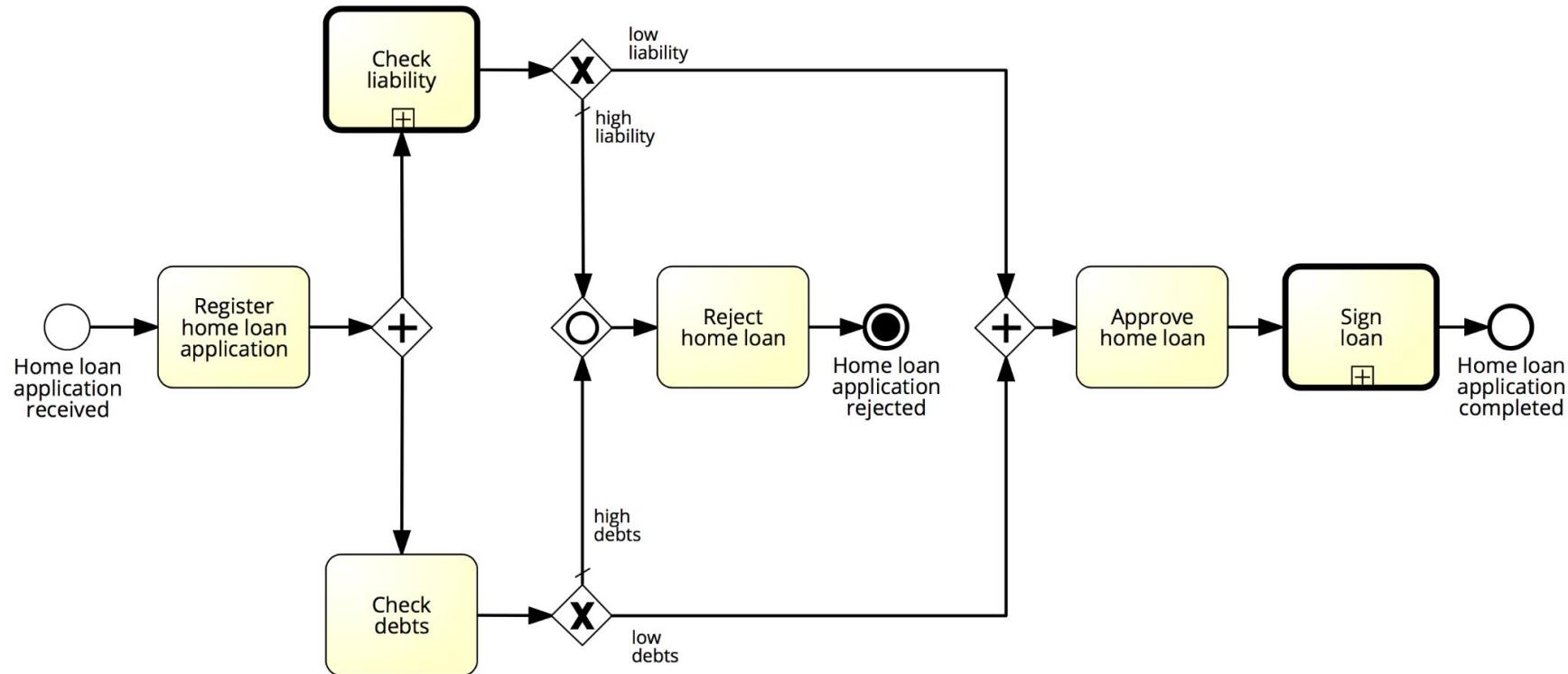
This can be done via the Terminate end event: it forces the whole process to *abort* (“wipes off” all tokens left behind, if any)





## Example 2: terminate event

Abort the process by removing all tokens...

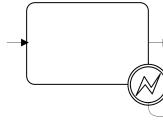




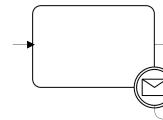
# Exception handling

Handling exceptions often involves stopping a sub-process and performing a special activity

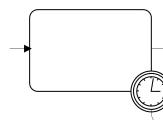
Types of exceptions for an activity (task/sub-process):



**Internal:** something goes wrong inside an activity, whose execution must thus be interrupted. Handled with the Error event



**External:** something goes wrong outside the process, and the execution of the current activity must be interrupted. Handled with the Message event

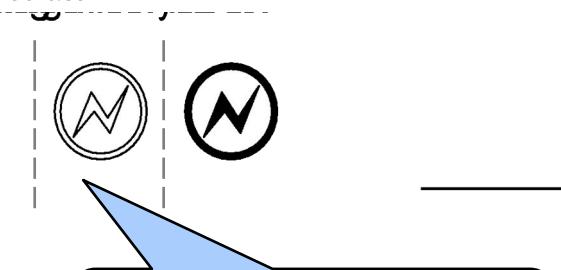


**Timeout:** an activity takes too long and must be interrupted. Handled with the Timer event

All these events are catching intermediate events. They stop the enclosing activity and start an exception handling routine.



# Internal exception: error event

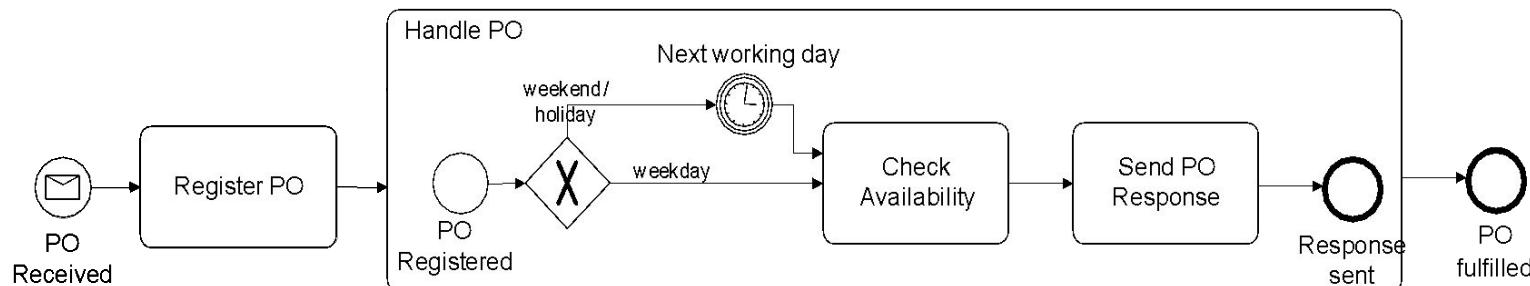




# Example: internal exception

## PO handling

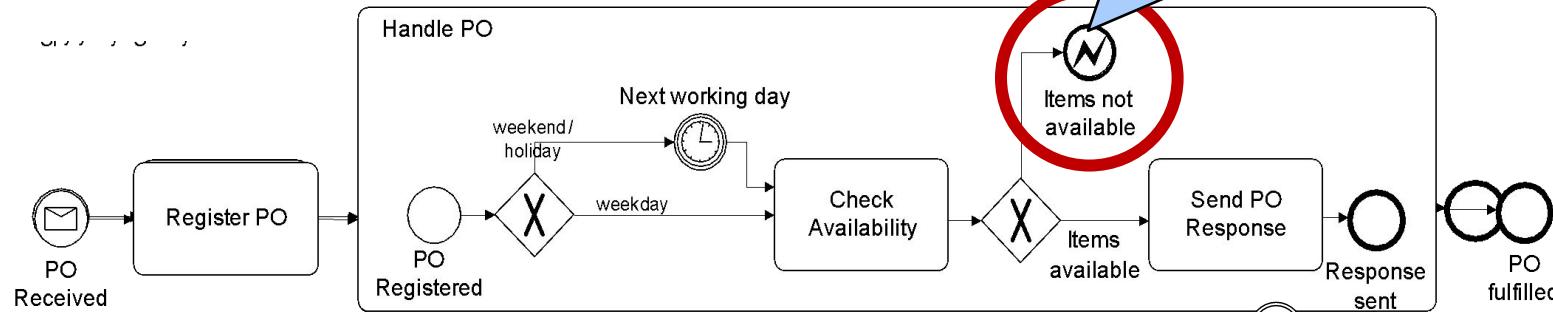
Consider again our “PO Handling process” example with the following extension: **if an item is not available**, any processing related to the PO **must be stopped**. Thereafter, the client needs to be notified that the PO cannot be further processed.





# Solution: internal exception

## PO handling



Throwing and catching error events must have the **same** label

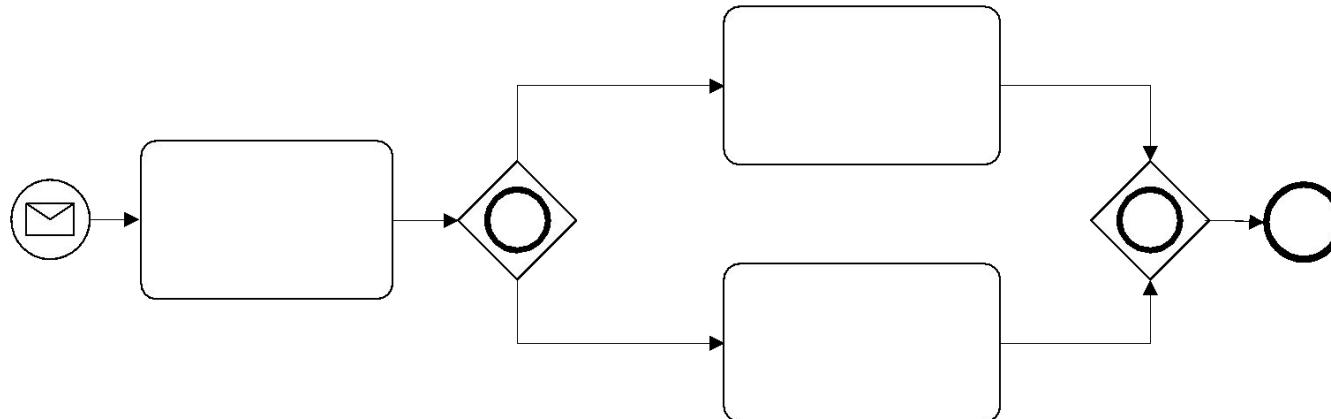
Must catch an error event thrown from **within** the same activity



# One more example: internal exception

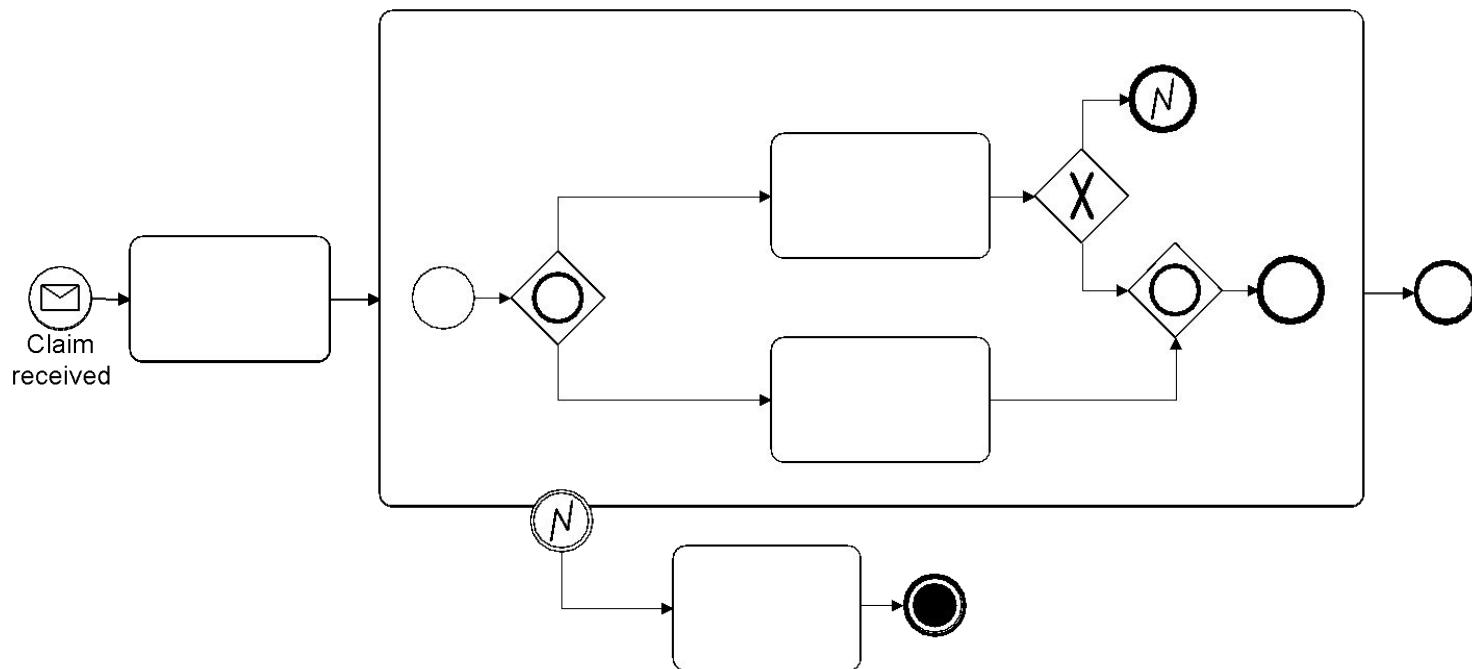
Extend the claim handling process shown below as follows:

After checking the insurance policy, a possible outcome is that the insurance is invalid. In this case, any processing is cancelled and a letter is sent to the customer. In the case of a complex claim, this implies that the damage checking is cancelled if it has not yet been completed.





# Solution: internal exception

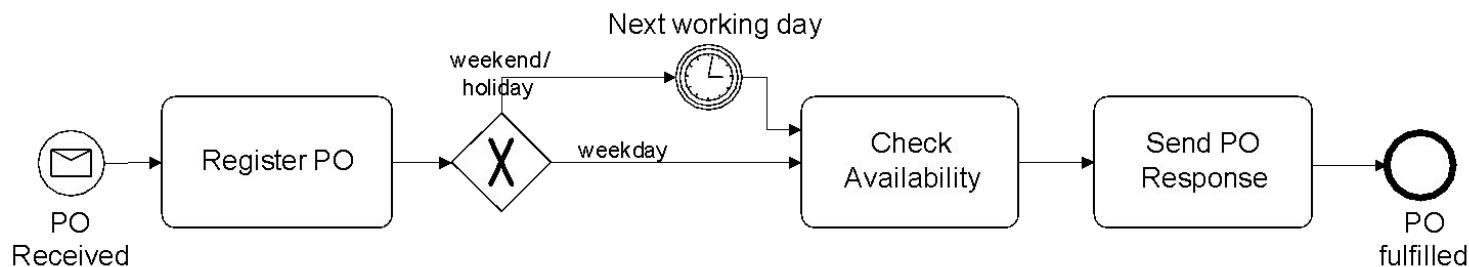




# Example: external exception

## PO handling

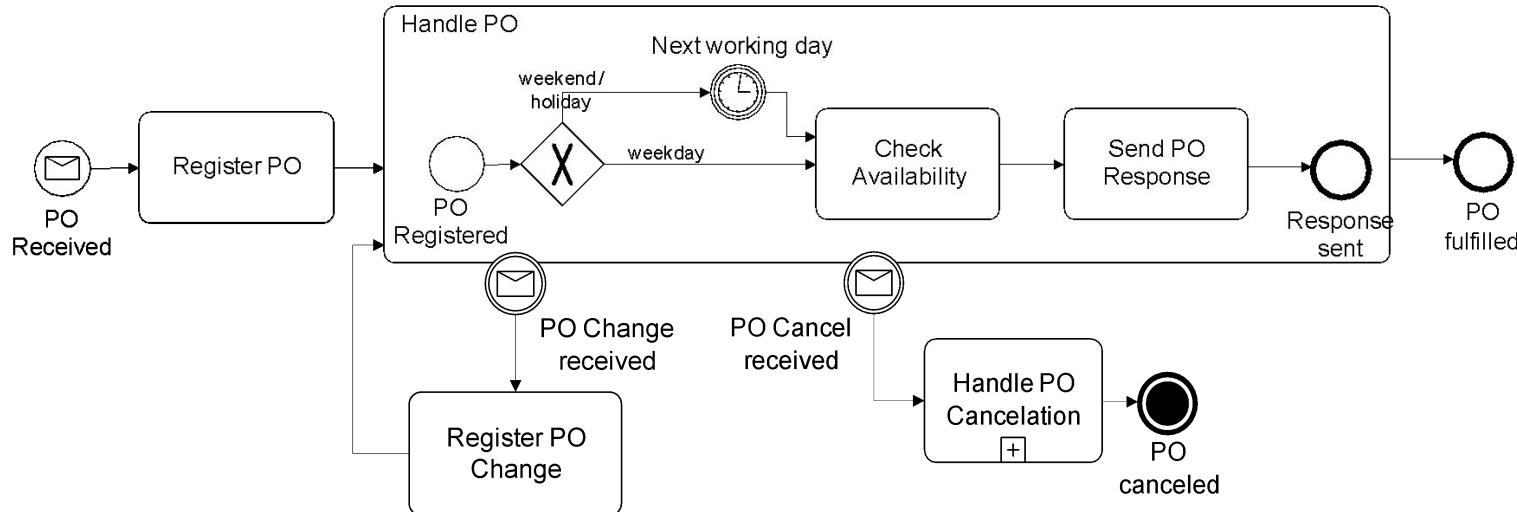
A PO change request may be **received anytime after the PO is registered**. This request includes a change in quantity or line items. When such a request is received, **any processing related to the PO must be stopped**. The PO change request is then registered. Thereafter, the process proceeds as it would do after a normal PO is registered. Further, if the customer **sends a PO cancelation request after the PO registration**, the PO processing **must be stopped** and the cancelation request must be handled.





# Solution: external exception

## PO handling





## Exercise 4.9

---

Model the following routine for accessing an Internet bank service.

*The routine for logging into an Internet bank account starts once the credentials entered from the user have been retrieved. First, the username is validated. If the username is not valid, the routine is interrupted and the invalid username is logged. If the username is valid, the number of password trials is set to zero. Then, the password is validated. If this is not valid, the counter for the number of trials is incremented and if lower than three, the user is asked to enter the password again, this time together with a CAPTCHA test to increase the security level. If the number of failed attempts reaches three times, the routine is interrupted and the account is frozen. Moreover, the username and password validation may be interrupted should the validation server not be available. Similarly, the server to test the CAPTCHA may not be available at the time of log in. In these cases, the procedure is interrupted after notifying the user to try again later. At any time during the log in routine, the customer may close the web page, resulting in the interruption of the routine.*



# Exercise 4.10: activity timeout

## Order-to-transportation quote

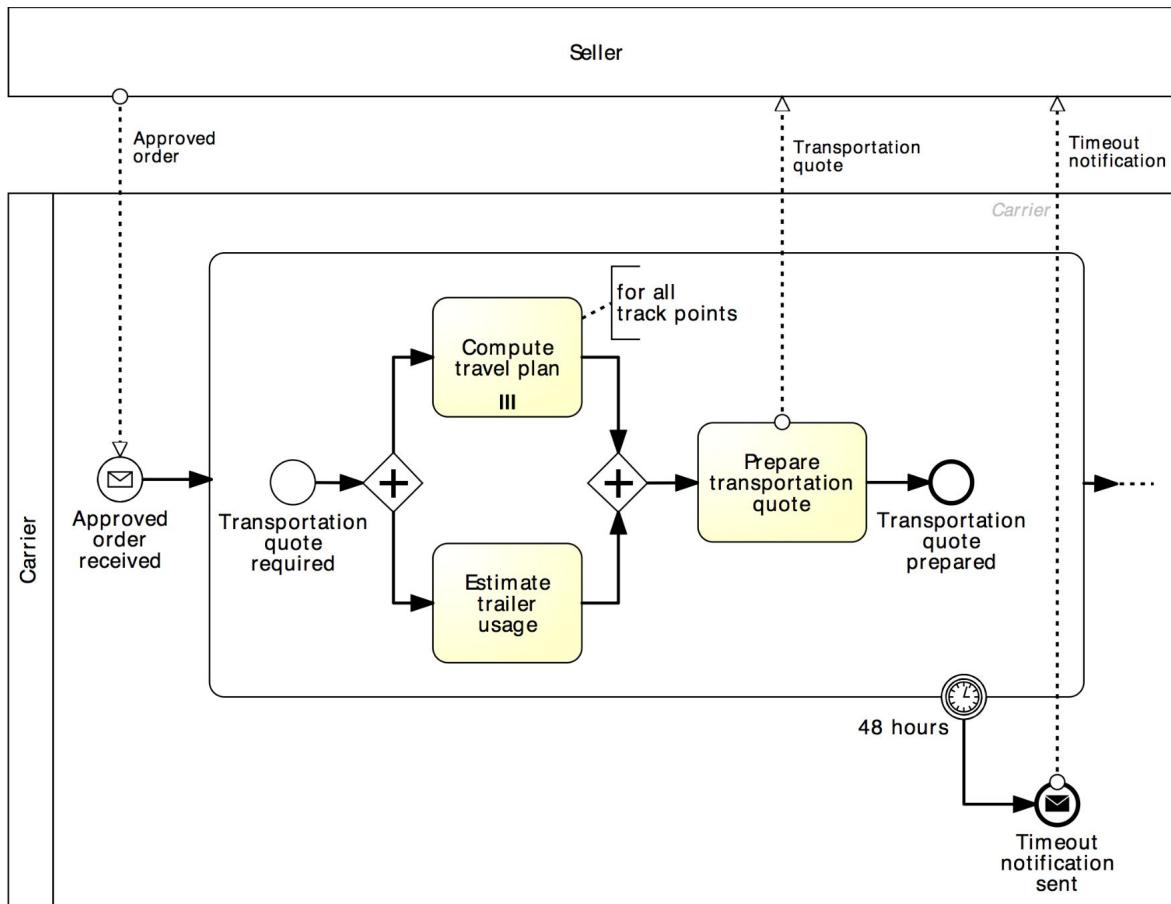
*Once a wholesale order has been confirmed, the supplier transmits this order to the carrier for the preparation of the transportation quote. In order to prepare the quote, the carrier needs to compute the route plan (including all track points that need to be traversed during the travel) and estimate the trailer usage.*

*By contract, wholesale orders have to be dispatched within four days from the receipt of the order. This implies that transportation quotes have to be prepared within 48 hours from the receipt of the order to remain within the terms of the contract.*



# Solution: activity timeout

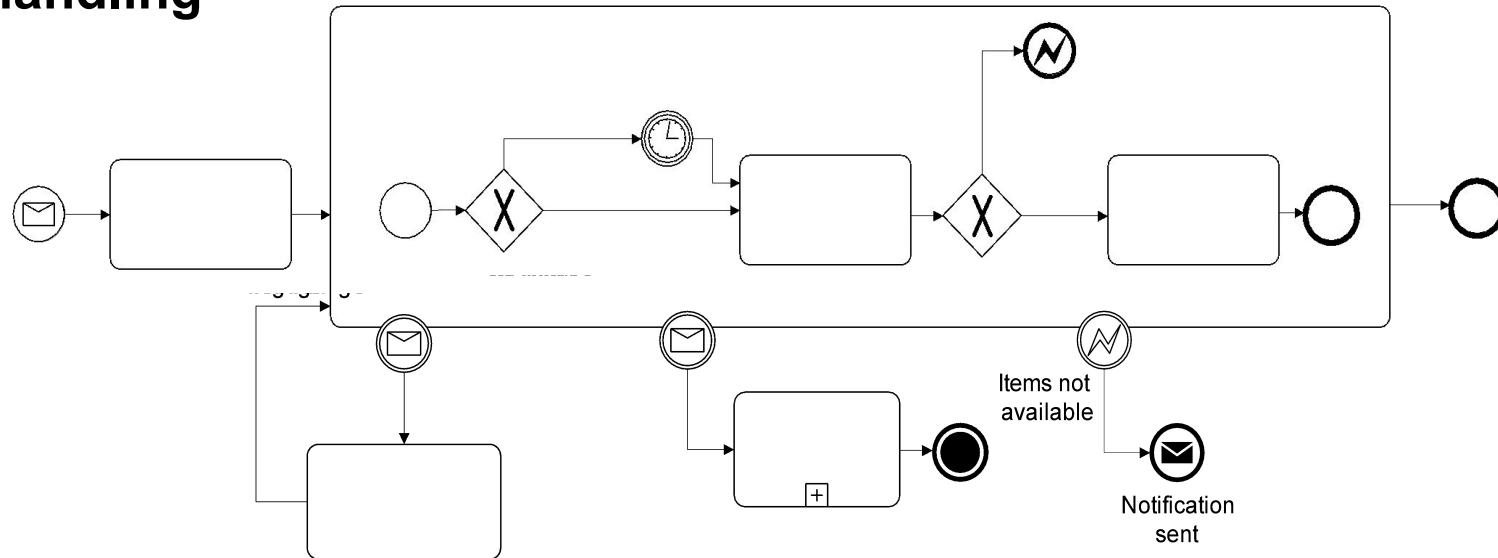
## Order-to-transportation quote





# Example: Non-interrupting boundary events

## PO handling



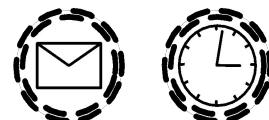
The customer may send a request for address change after the PO registration. When such a request is received, **it is just registered**, without further action.



# Non-interrupting boundary events

Sometimes we may need to trigger an activity **in parallel** to the normal flow, i.e. without interrupting the normal flow.

This can be achieved by using *non-interrupting* boundary events

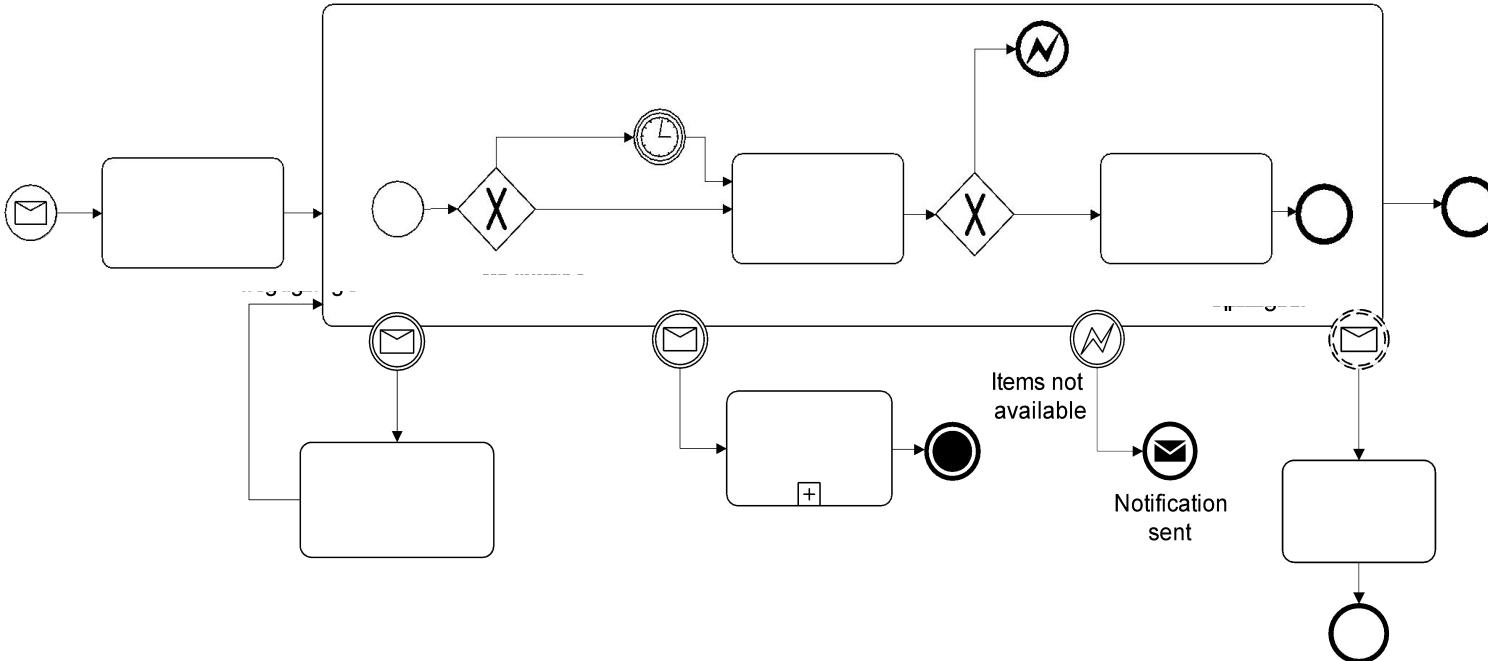


Must be attached to  
the activity's boundary



# Solution: non-interrupting boundary events

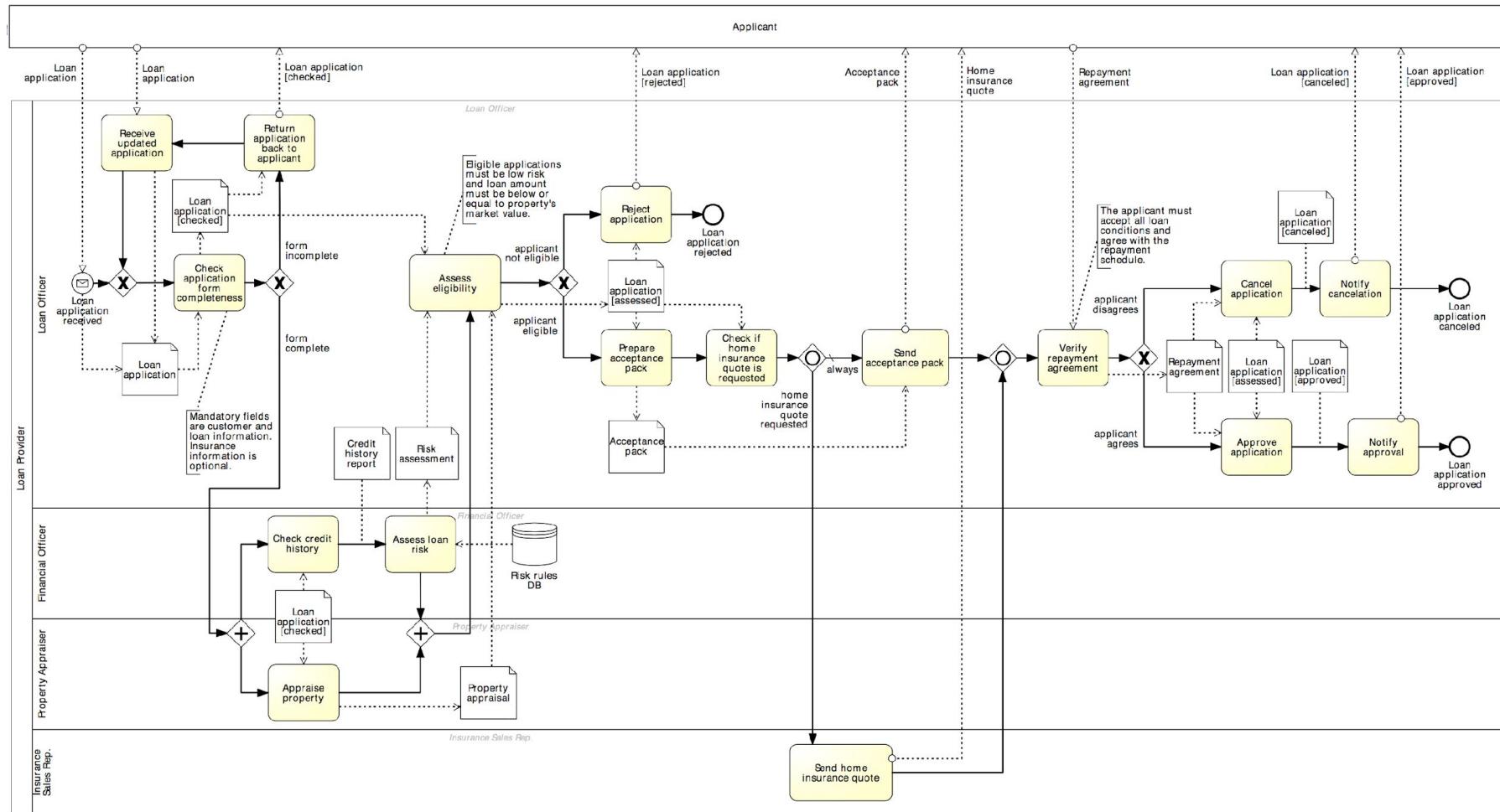
## PO handling





# Exercise 4.11

Consider the process for assessing loan applications below.





## Exercise 4.11

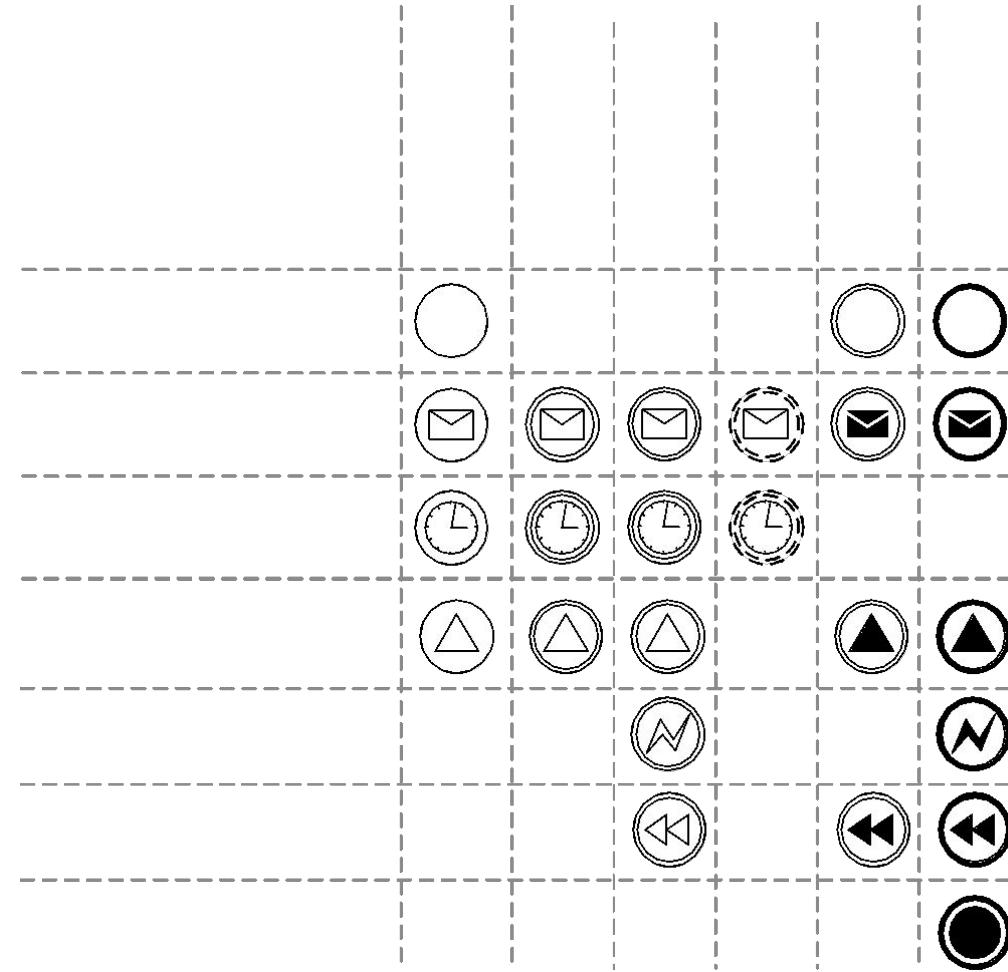
---

Extend this process as follows.

*An applicant who has decided not to combine the loan with a home insurance plan may change its mind any time before the eligibility assessment has been completed. If a request for adding an insurance plan is received during this period, the loan provider will simply update the loan application with this request.*

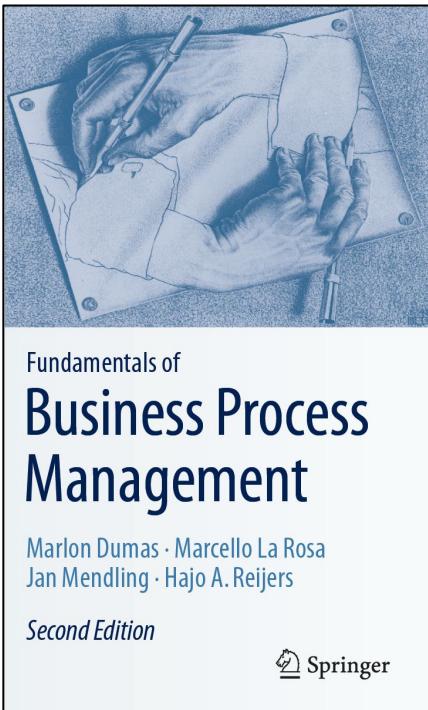


# Recap: Events





# Chapter 4: Advanced Process Modeling



## Contents

1. More on Rework and Repetition
2. Handling Events
3. Handling Exceptions
4. Processes and Business Rules
5. Recap



# Processes and Business Rules

- Business rules implement organizational policies or practices
- Example: in an online shop platinum customers have a 20% discount for purchases over EUR 250.
- In BPMN, business rules can be captured via:
  - Decision activities
  - Conditions on outgoing arcs of (X)OR-split
  - Conditional events



# Example: conditional event

## Replenishment order

