Business Process Management

Paolo Bottoni

Dipartimento di Informatica

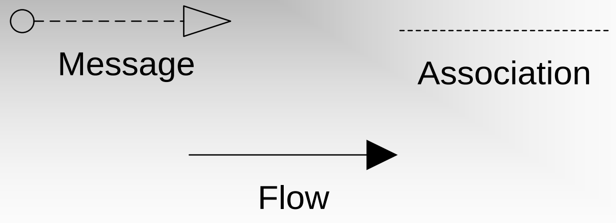
Sapienza Università di Roma

Block 5: AdvancedBPM

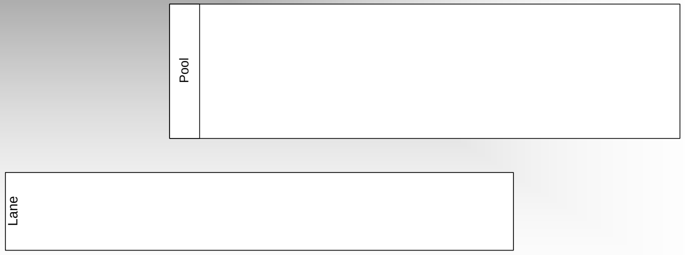
Adapted from the slides for the book : Dumas, La Rosa, Mendling & Reijers:
Fundamentals of Business Process Management, Springer 2013
http://courses.cs.ut.ee/2013/bpm/uploads/Main/ITlecture3.ppt

BPMN Main Elements - Recap

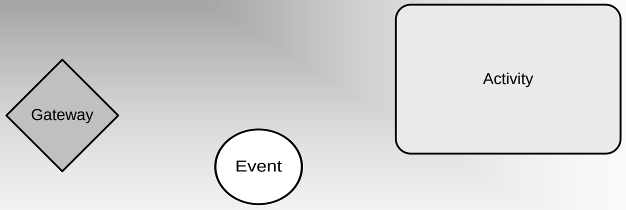
Connections



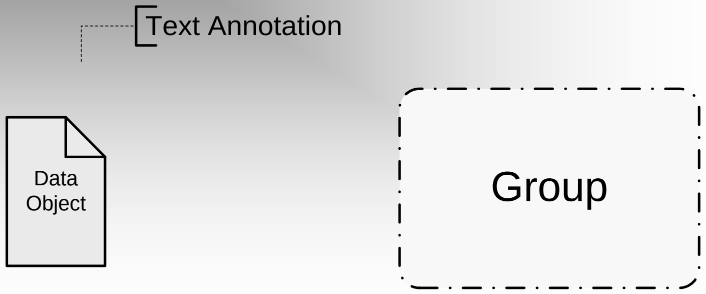
Swimlanes



Flow Objects



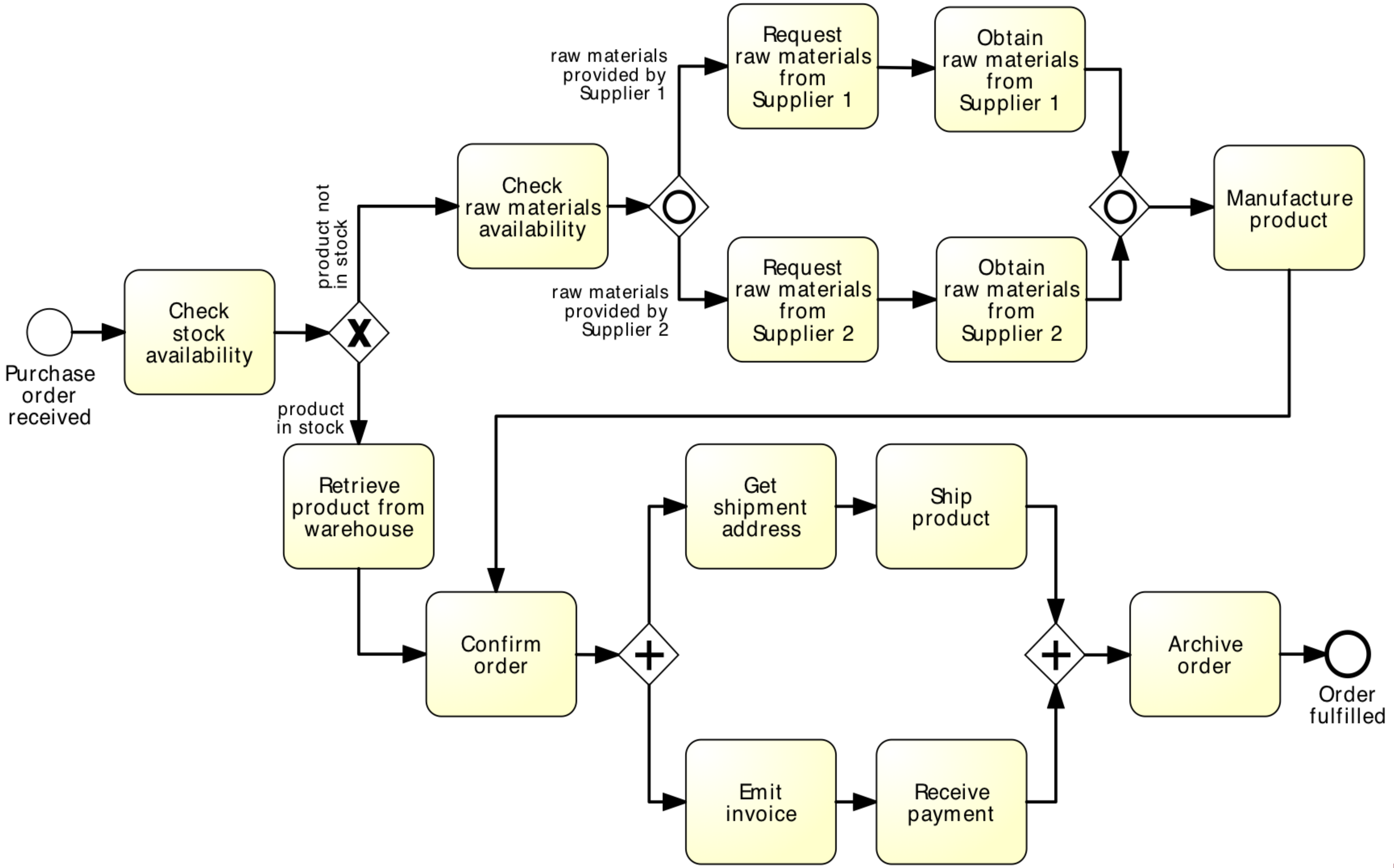
Artifacts



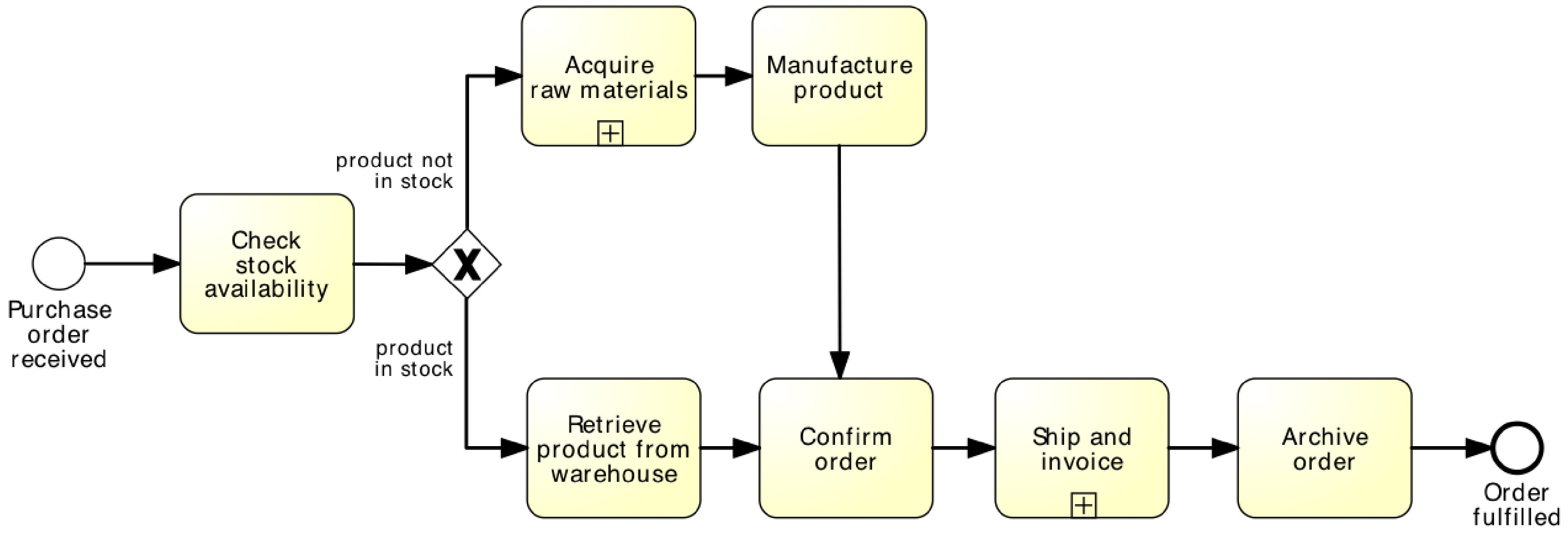
BPMN Gateways

Exclusive (XOR)	Parallel (AND)	Inclusive (OR)
• <u>Exclusive decision</u> take one branch • <u>Exclusive merge</u> Proceed when one branch has completed	• <u>Parallel split</u> take all branches • <u>Parallel join</u> proceed when all incoming branches have completed	• <u>Inclusive decision</u> take one or several branches depending on conditions • <u>Inclusive merge</u> proceed when all <u>active</u> incoming branches have completed

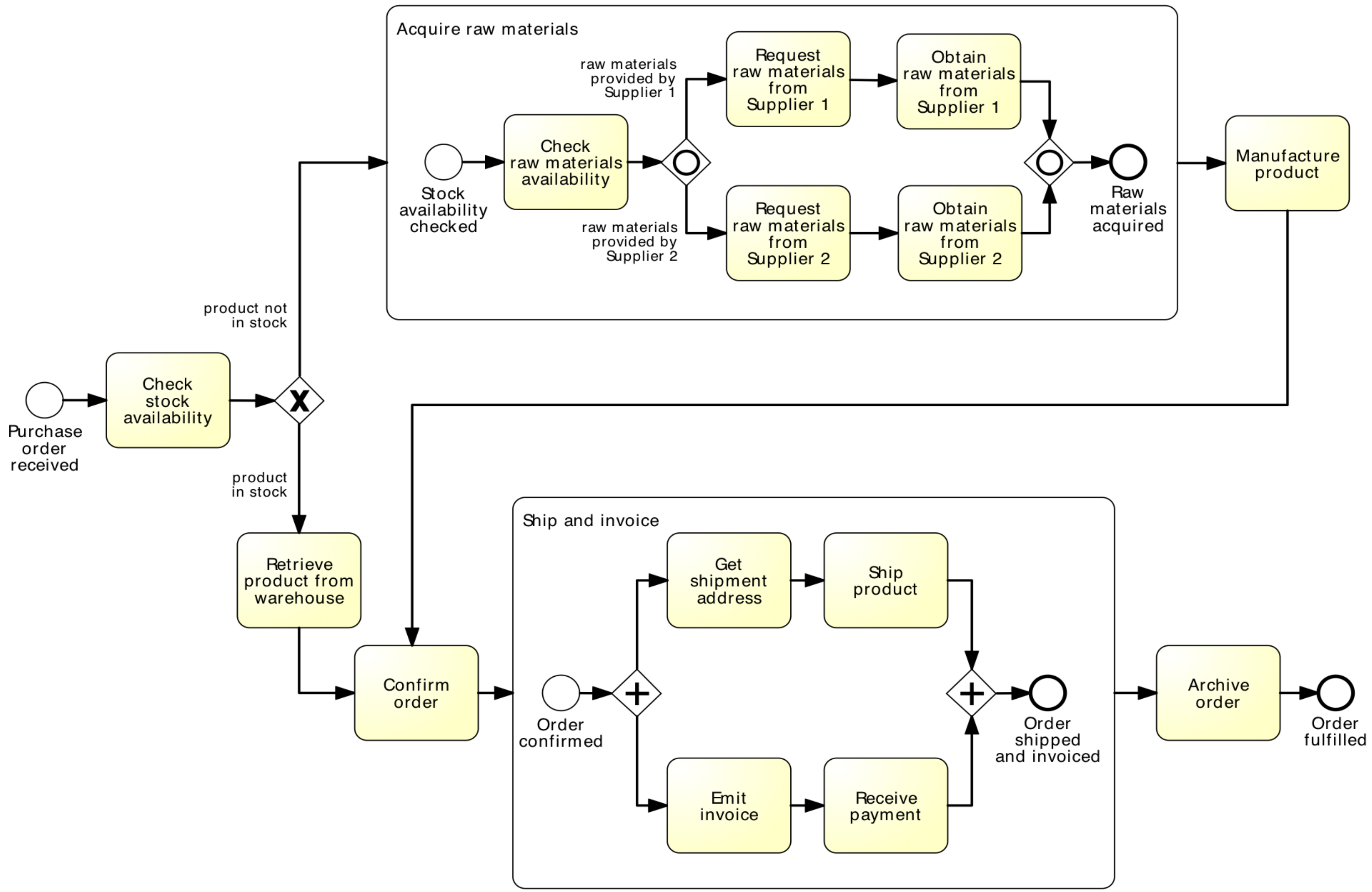
Anything wrong with this model?


Purchase order received
Check stock availability
product not in stock
product in stock
Check raw materials availability
raw materials provided by Supplier 1
raw materials provided by Supplier 2
Request raw materials from Supplier 1
Obtain raw materials from Supplier 1
Request raw materials from Supplier 2
Obtain raw materials from Supplier 2
Manufacture product
Retrieve product from warehouse
Confirm order
Get shipment address
Ship product
Emit invoice
Receive payment
Archive order
Order fulfilled

Is this better?



Expanded…


Acquire raw materials
Stock availability checked
Check raw materials availability
raw materials provided by Supplier 1
raw materials provided by Supplier 2
Request raw materials from Supplier 1
Obtain raw materials from Supplier 1
Request raw materials from Supplier 2
Obtain raw materials from Supplier 2
Raw materials acquired
Manufacture product
Purchase order received
Check stock availability
product not in stock
product in stock
Retrieve product from warehouse
Confirm order
Ship and invoice
Order confirmed
Get shipment address
Ship product
Emit invoice
Receive payment
Order shipped and invoiced
Archive order
Order fulfilled

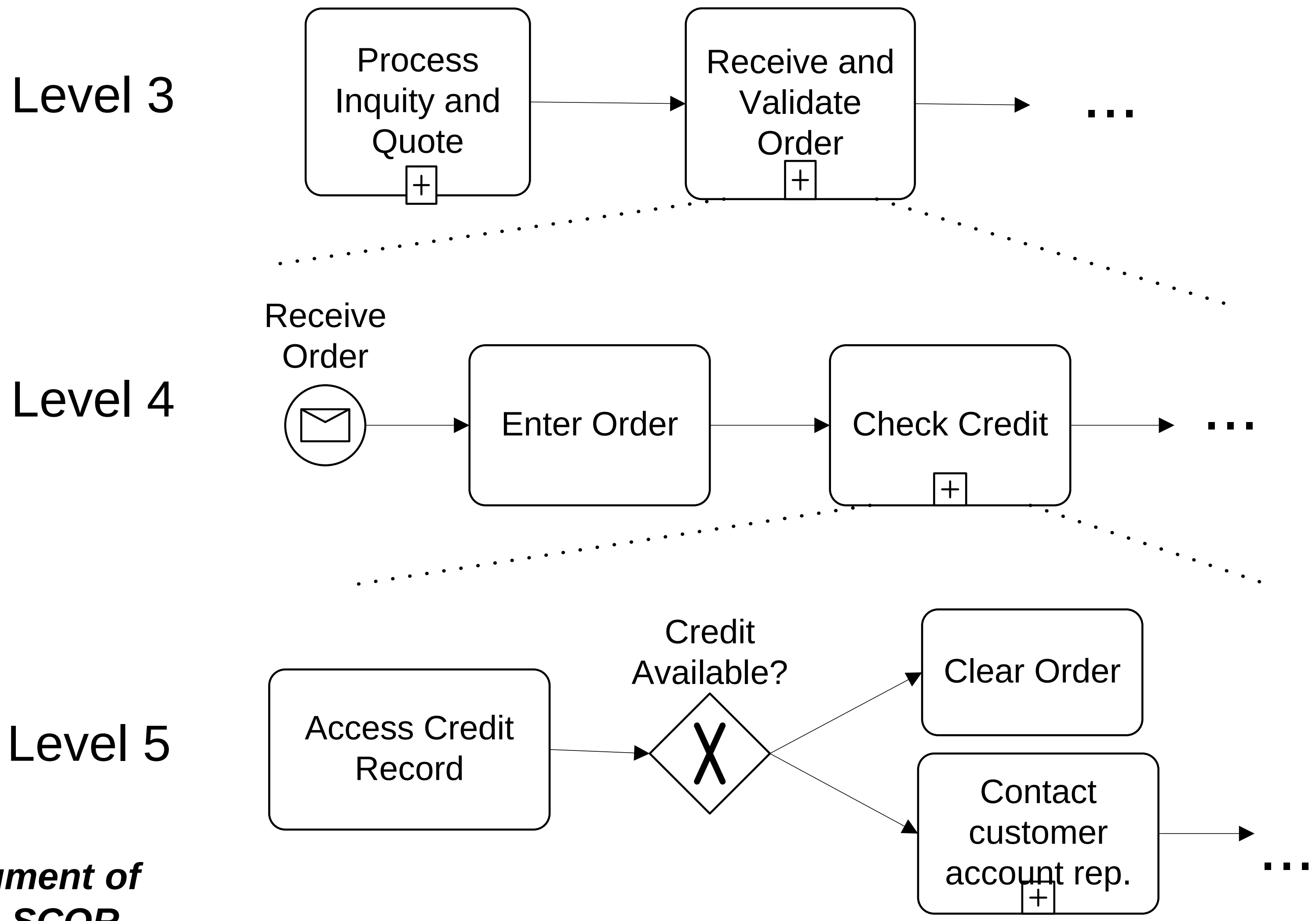
Sub-processes

- An activity in a process can “invoke” a separate (sub-)process
- Use this feature to:
  1. Break down large models into smaller ones, making them easier to understand and maintain

     → process hierarchies
  2. Share common fragments across multiple processes

     → shared subprocesses
  3. Identify parts of a process that should be:
     - repeated
     - executed multiple times in parallel
     - cancelled

Process hierarchies

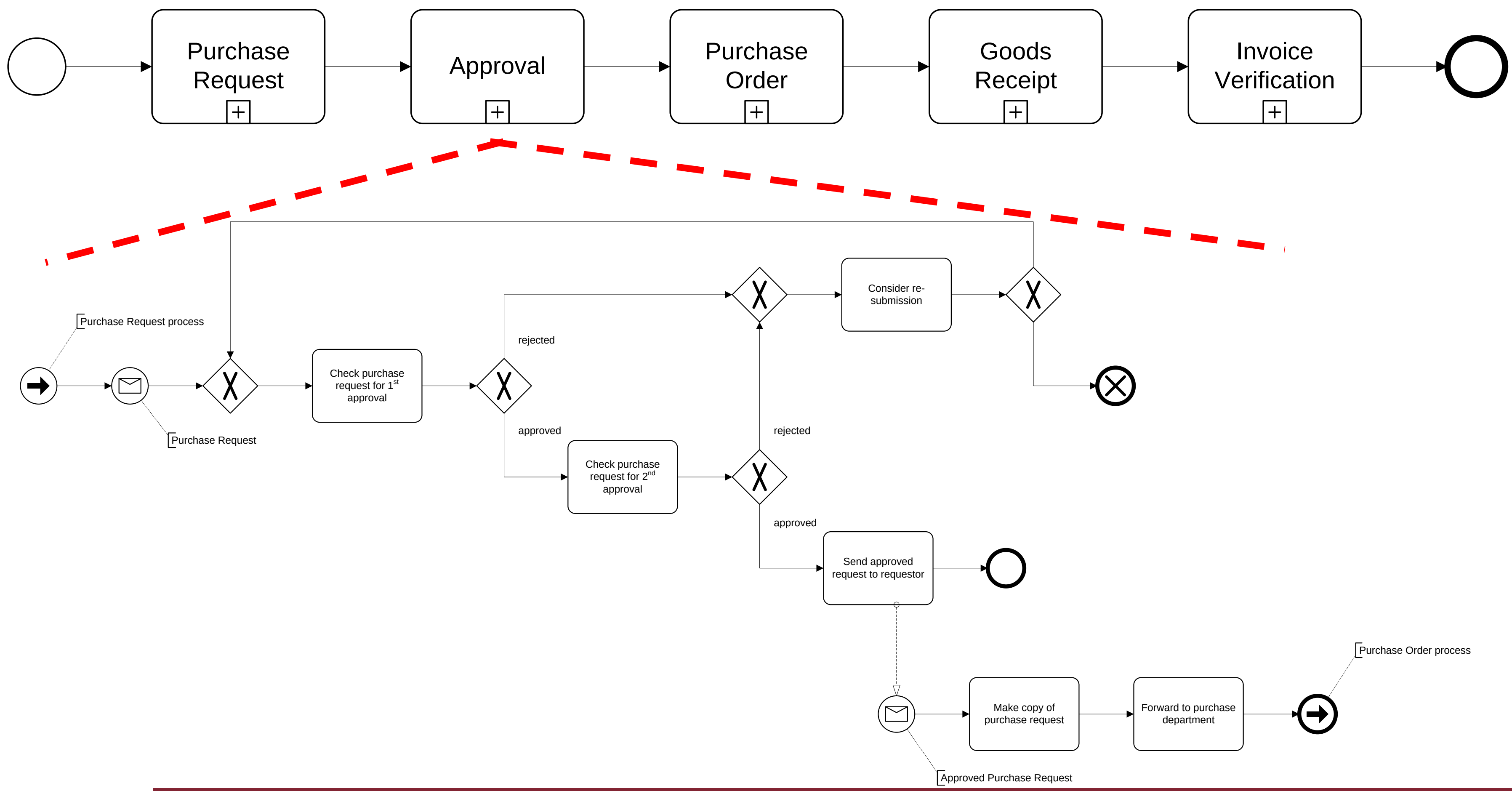


Fragment of the SCOR model

Modeling Guideline
Start with a value chain

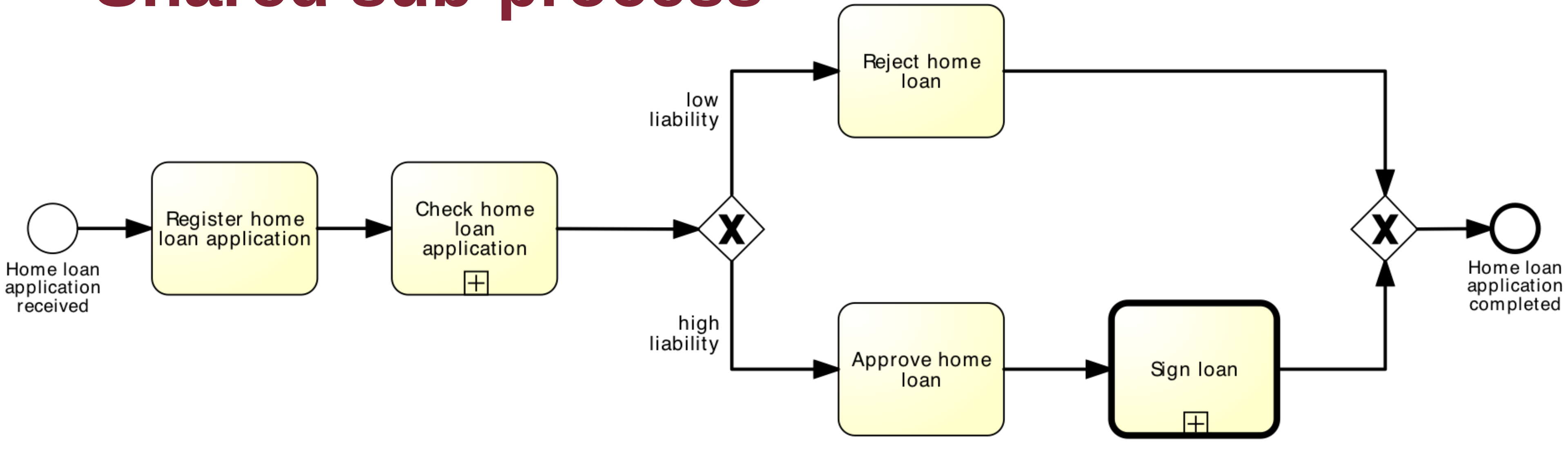
- Good practice is that the top-level process should be simple (no gateways) and should show the main phases of the process
  - Each phase then becomes a sub-process
  - This is sometimes called a “value chain”

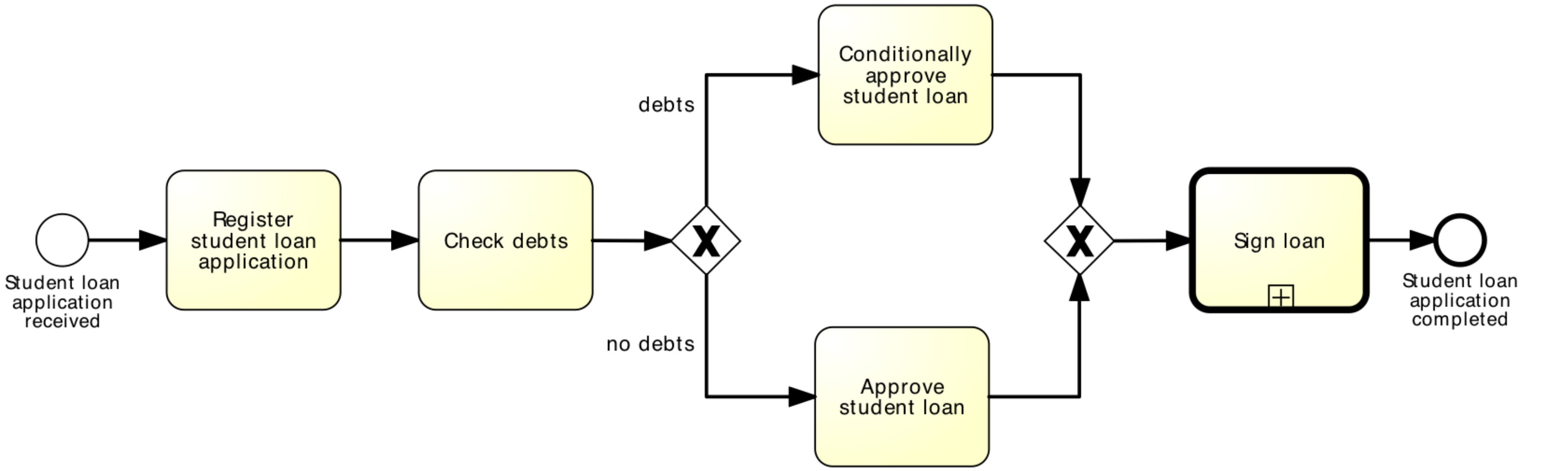
Showing the value chain with sub-processes



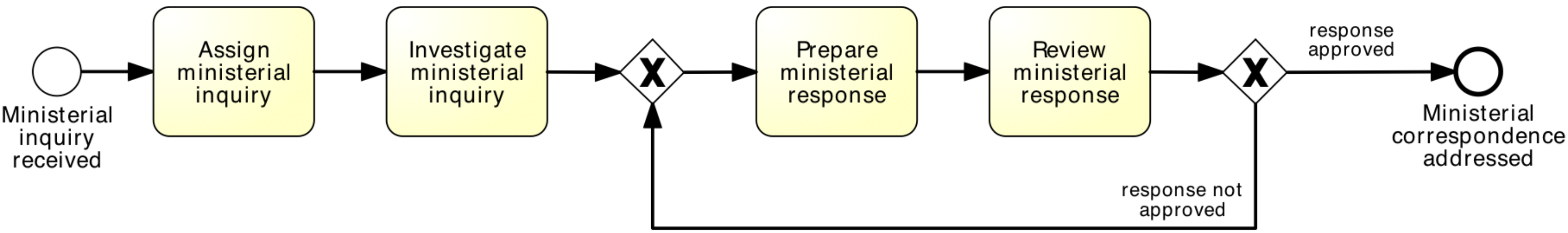
Purchase Request
Approval
Purchase Order
Goods Receipt
Invoice Verification
Purchase Request process
Purchase Request
Check purchase request for 1st approval
rejected
approved
Check purchase request for 2nd approval
rejected
approved
Consider re-submission
Send approved request to requestor
Make copy of purchase request
Forward to purchase department
Purchase Order process
Approved Purchase Request

Shared sub-process

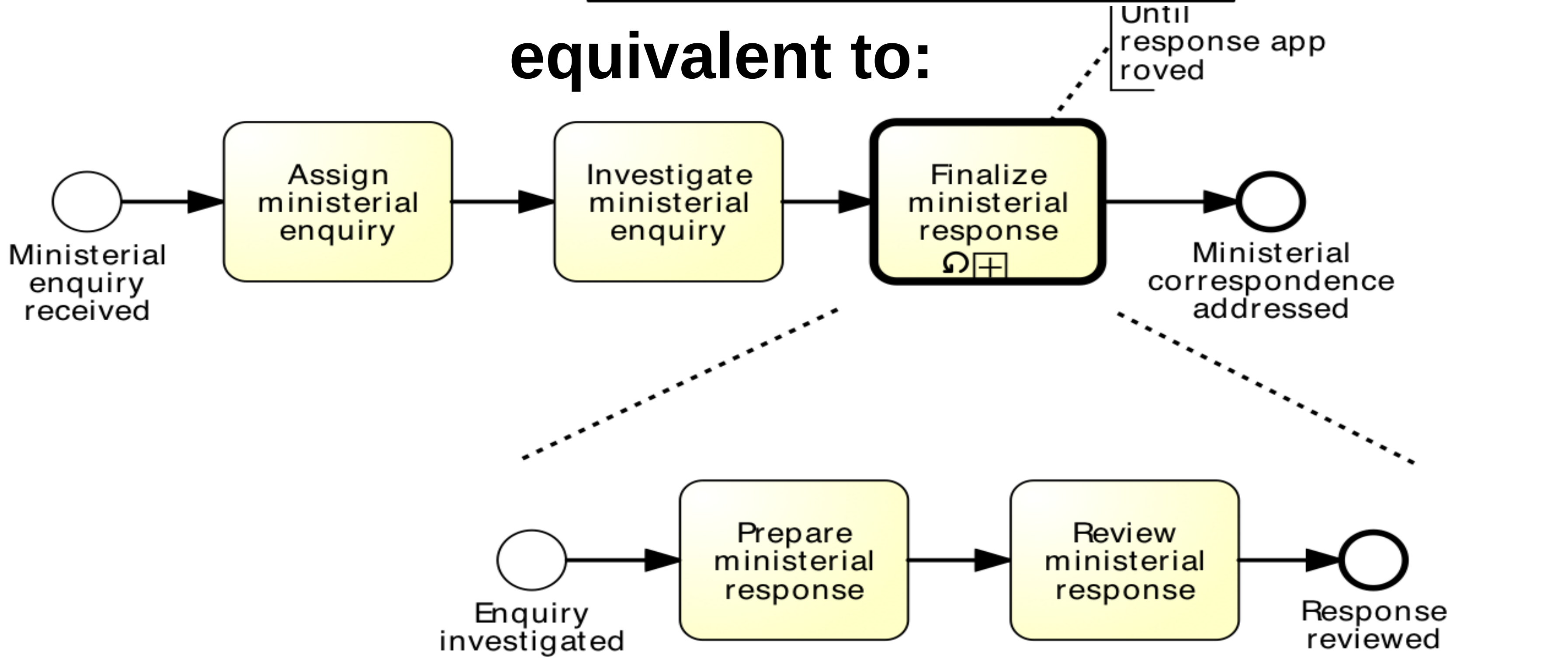

Home loan application received
Register home loan application
Check home loan application
low liability
high liability
Reject home loan
Approve home loan
Sign loan
Home loan application completed


Student loan application received
Register student loan application
Check debts
debts
no debts
Conditionally approve student loan
Approve student loan
Sign loan
Student loan application completed

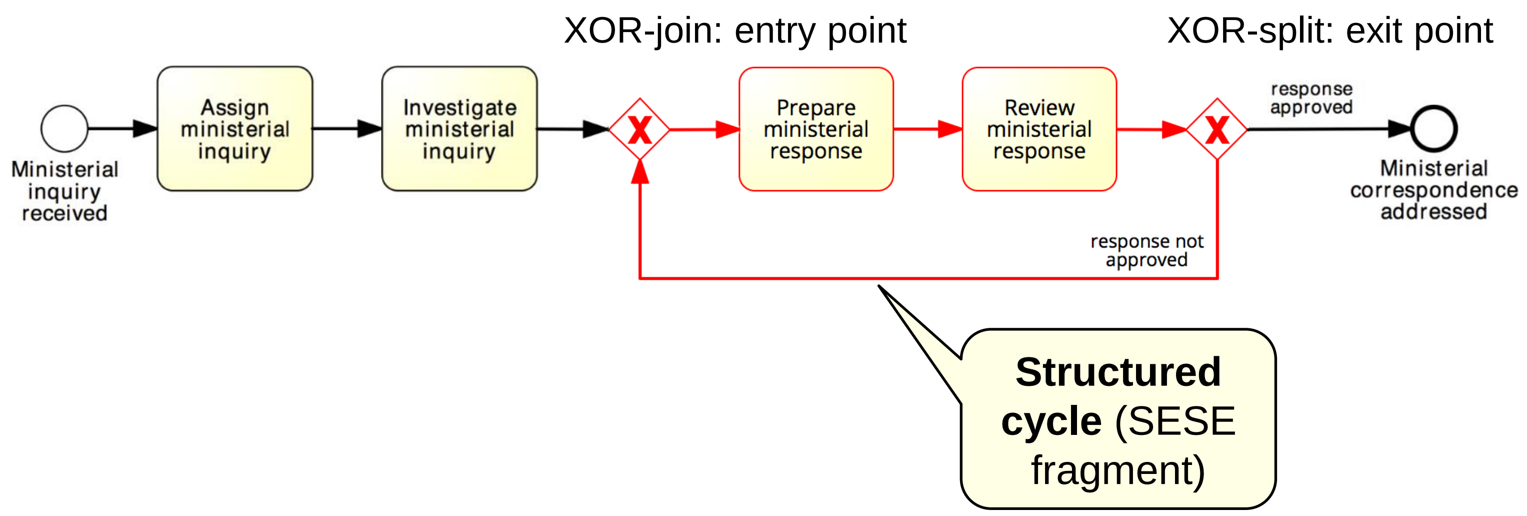
Sub-processes and loop marker


Ministerial inquiry received
Assign ministerial inquiry
Investigate ministerial inquiry
Prepare ministerial response
Review ministerial response
response approved
Ministerial correspondence addressed
response not approved

equivalent to:

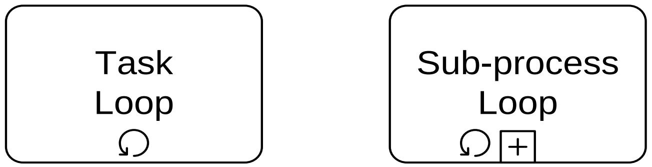

Until response app roved
Ministerial enquiry received
Assign ministerial enquiry
Investigate ministerial enquiry
Finalize ministerial response
Ministerial correspondence addressed
Enquiry investigated
Prepare ministerial response
Review ministerial response
Response reviewed

More on rework and repetition

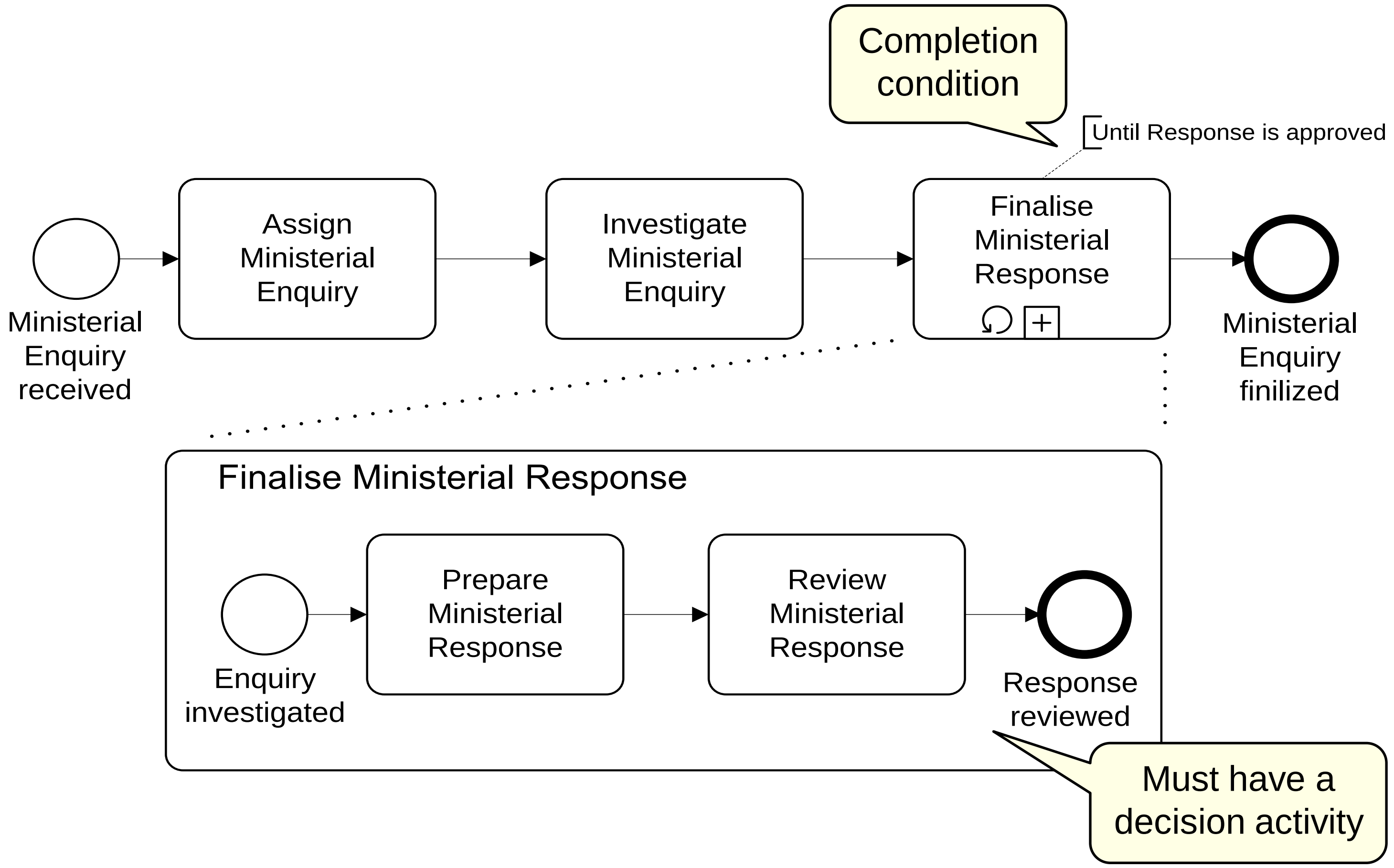

XOR-join: entry point
XOR-split: exit point
Ministerial inquiry received
Assign ministerial inquiry
Investigate ministerial inquiry
Prepare ministerial response
Review ministerial response
response approved
response not approved
Ministerial correspondence addressed
Structured cycle (SESE fragment)

Block-structured repetition: Loop Activity

In BPMN also *loop activity* to allow repetition of a task or sub-process



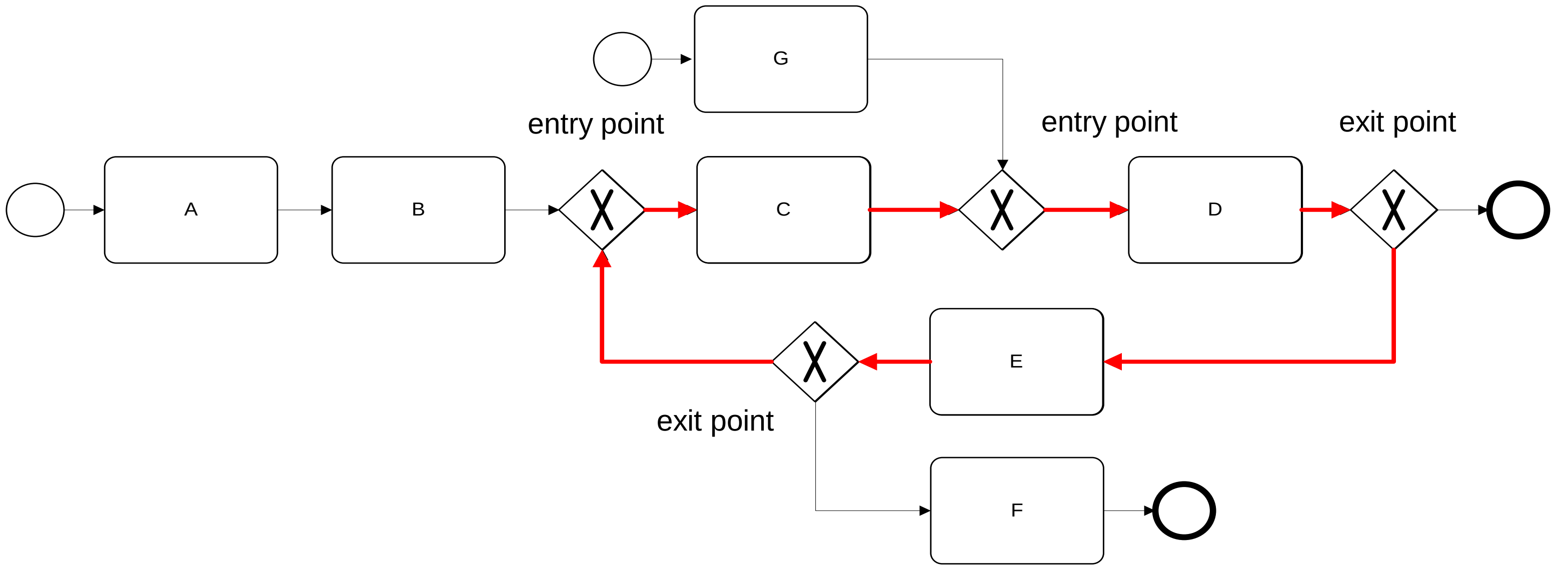
Example: block-structured repetition


Completion condition
Until Response is approved
Ministerial Enquiry received
Assign Ministerial Enquiry
Investigate Ministerial Enquiry
Finalise Ministerial Response
Ministerial Enquiry finilized
Finalise Ministerial Response
Enquiry investigated
Prepare Ministerial Response
Review Ministerial Response
Response reviewed
Must have a decision activity

Loop Activity or (Arbitrary) Cycle?

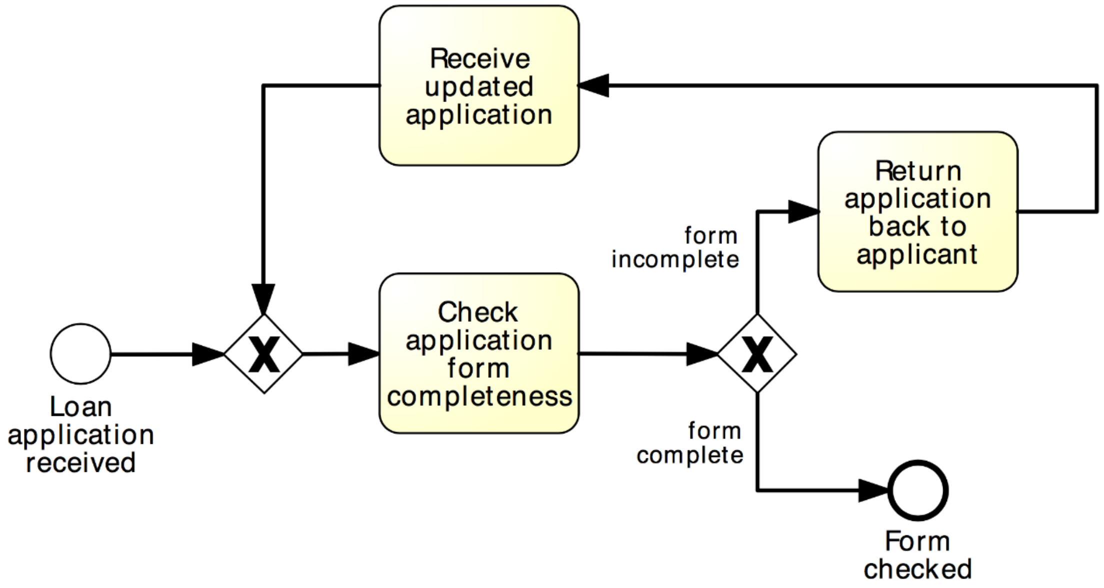
Arbitrary Cycles

Arbitrary = unstructured, i.e. it can have multiple entry and exit nodes (non-SESE).



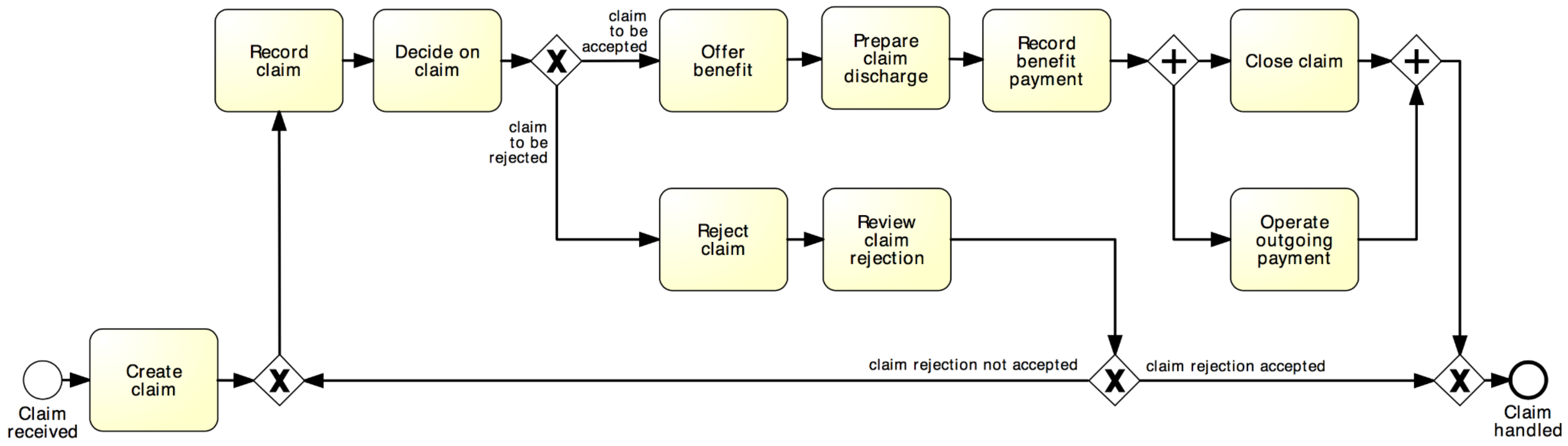
Exercise 4.1

- Identify the entry and exit points that delimit the unstructured cycles in the process model shown below. What are the repetition blocks?
- Model the business process shown below using a loop activity.


Receive updated application
Return application back to applicant
form incomplete
Check application form completeness
Loan application received
form complete
Form checked

Exercise 4.2

- Identify the entry and exit points that delimit the unstructured cycles in the process model shown below. What are the repetition blocks?



Exercise

After a claim is registered, it is examined by a claims officer. The claims officer then writes a “settlement recommendation”. This recommendation is checked by a senior claims officer who may mark the claim as “OK” or “Not OK”. If the claim is marked as “Not OK”, it is sent back to the claims officer and examination is repeated. If the claim is marked as “OK”, the claims officer notifies the settlement to the customer.

Exercise (name-verb analysis)

After a claim is registered, it is examined by a claims officer. The claims officer then writes a “settlement recommendation”. This recommendation is checked by a senior claims officer who may mark the claim as “OK” or “Not OK”. If the claim is marked as “Not OK”, it is sent back to the claims officer and examination is repeated. If the claim is marked as “OK”, the claims officer notifies the settlement to the customer.

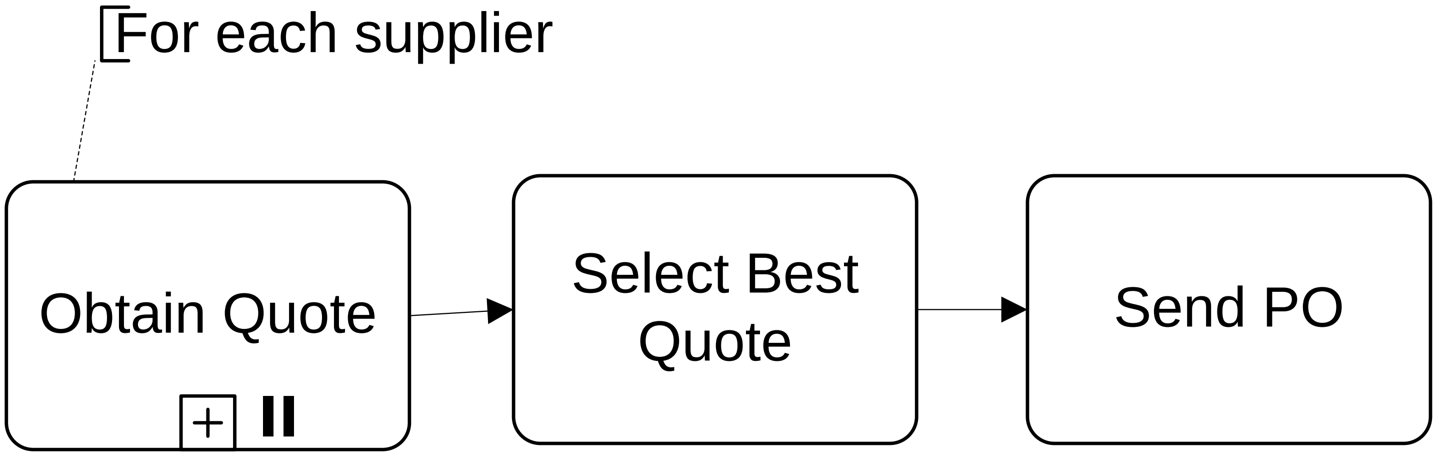
• Objects	🟨
• Tasks	🟩
• Paths	🟧
• Iterations	🟦
• Actors	🟥

- $(Acts, \leq)$
- $record \leq close, record \leq operate$
- $prepare \leq record$
- Control flow-induced partial order
- Resource-induced partial order

Multiple instance marker

- “Multiple instance” marker ||| means “parallel repetition” of an activity/sub-process
- Useful when the same activity should be executed for multiple entities or data items, e.g.
  - Request quotes from multiple suppliers
  - Check availability for each line item in an order separately
  - Send and gather questionnaires for multiple witnesses in the context of an insurance claim

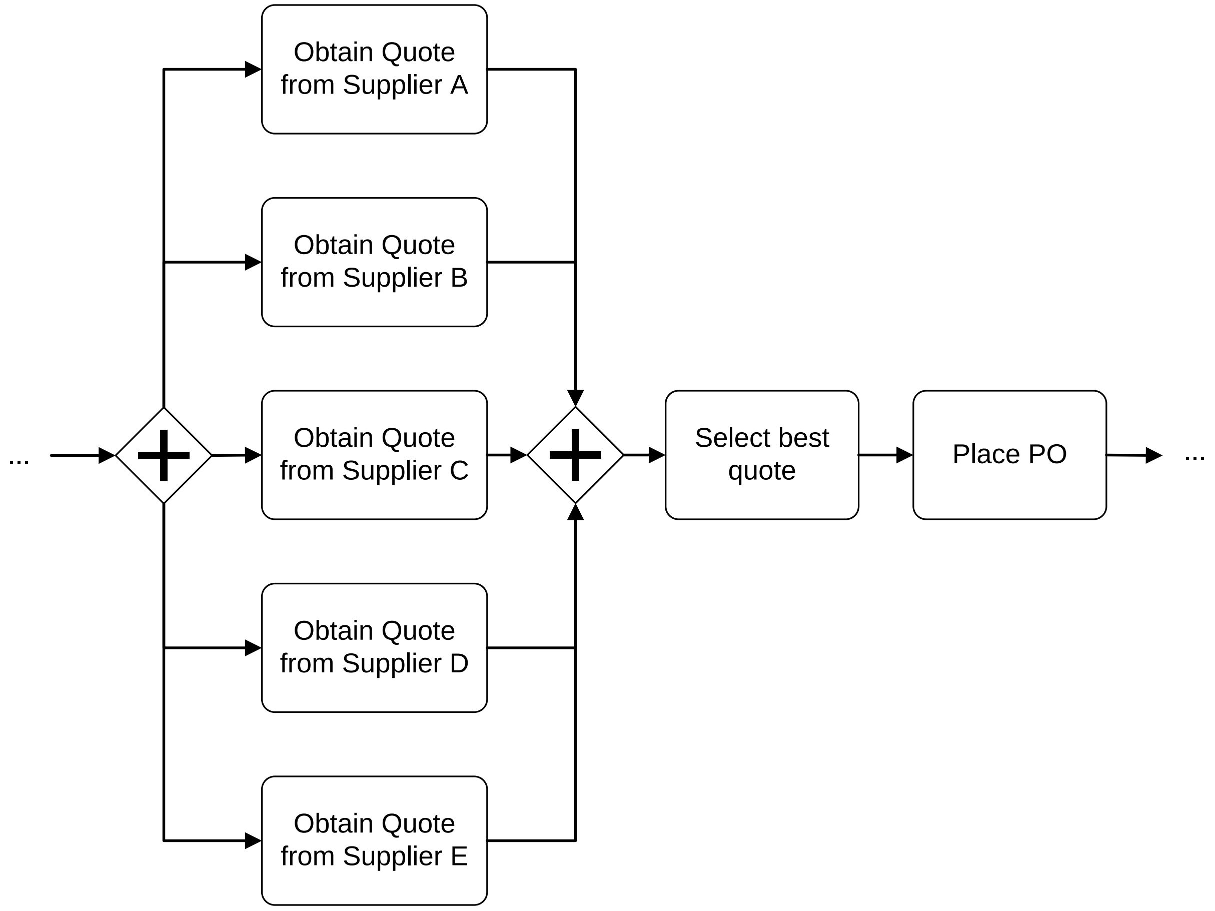
Multiple instance activity - example



Example: multi-instance activity

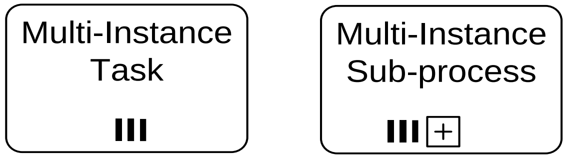
In procurement, typically a quote is to be obtained from all preferred suppliers (assumption: five preferred suppliers exist). After all quotes are received, they are evaluated and the best quote is selected. A corresponding purchase order is then placed.

Solution: without multi-instance activity


...
Obtain Quote from Supplier A
Obtain Quote from Supplier B
Obtain Quote from Supplier C
Obtain Quote from Supplier D
Obtain Quote from Supplier E
Select best quote
Place PO
...

Parallel repetition: multi-instance activity

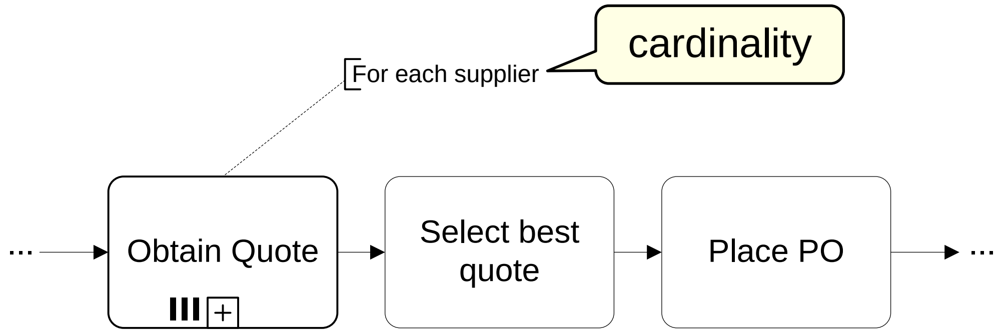
Provides a mechanism to indicate that an activity is executed *multiple times concurrently*



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

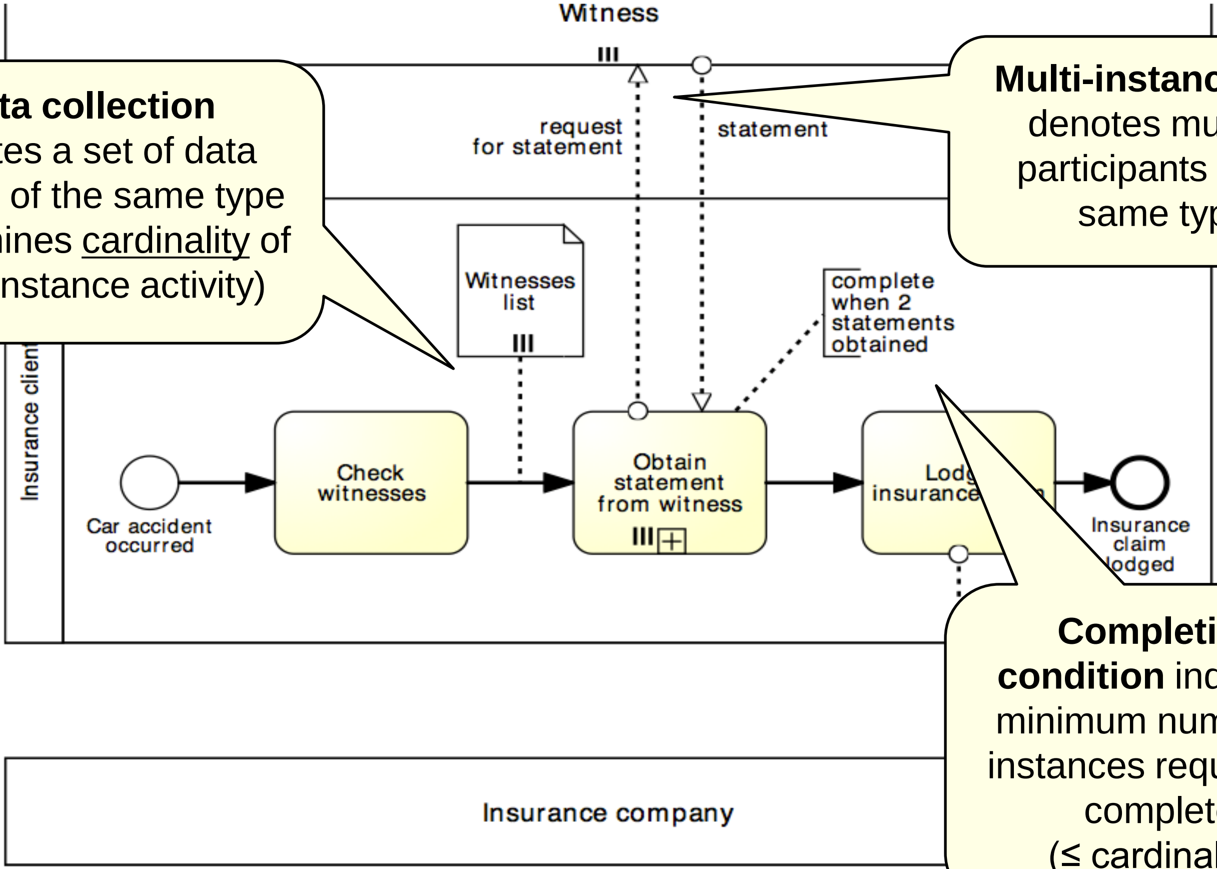
Solution: with multi-instance activity



Further example: multi-instance activity

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



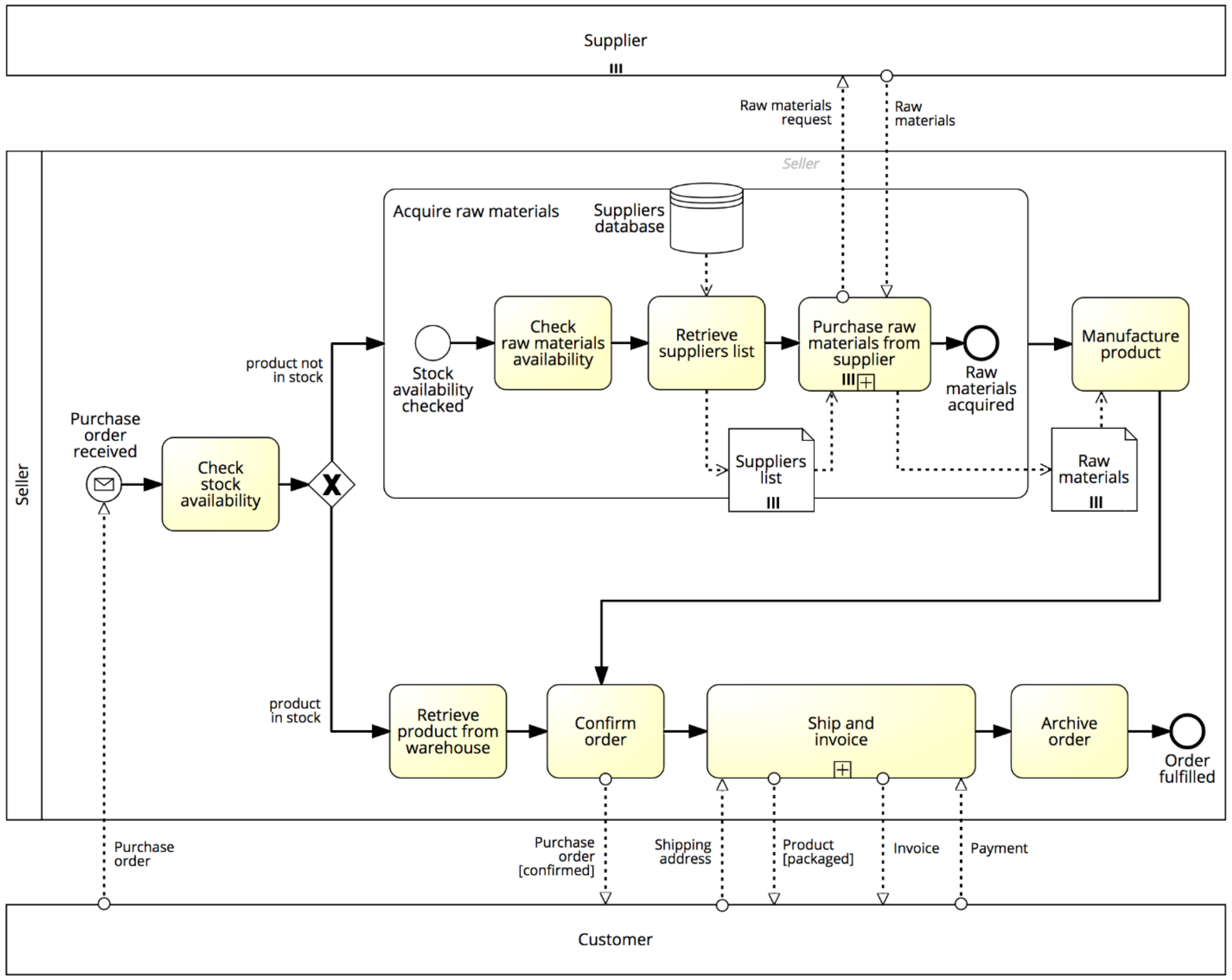
Data collection denotes a set of data objects of the same type (determines cardinality of multi-instance activity)

Multi-instance pool denotes multiple participants of the same type

Completion condition indicates minimum number of instances required to complete (≤ cardinality)

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

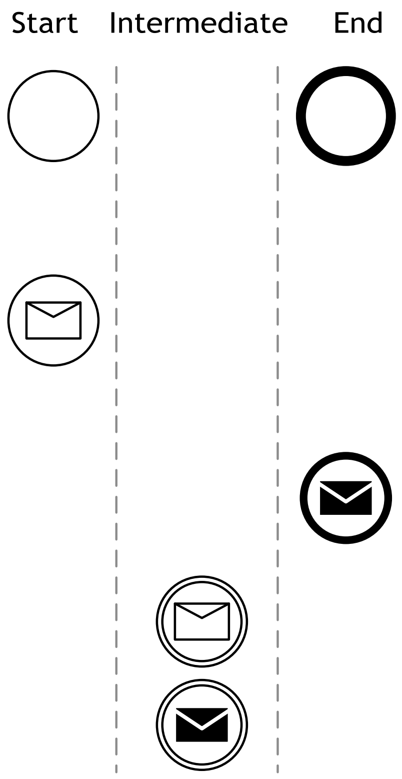
Events handling

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

They affect the process flow:

- Start
- Intermediate
- End

BPMN event types



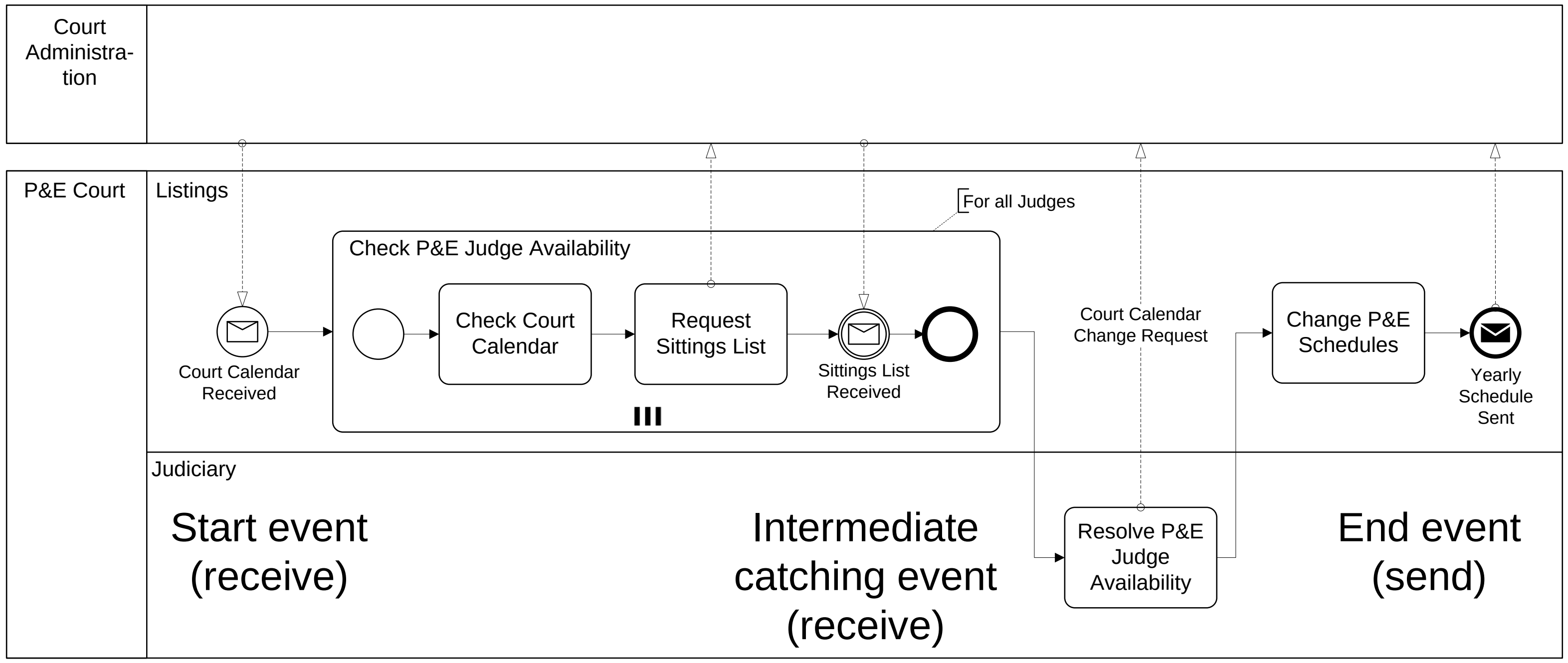
Untyped Event – Indicates that an instance of the process is created (start) or completed (end), without specifying the cause for creation/completion

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

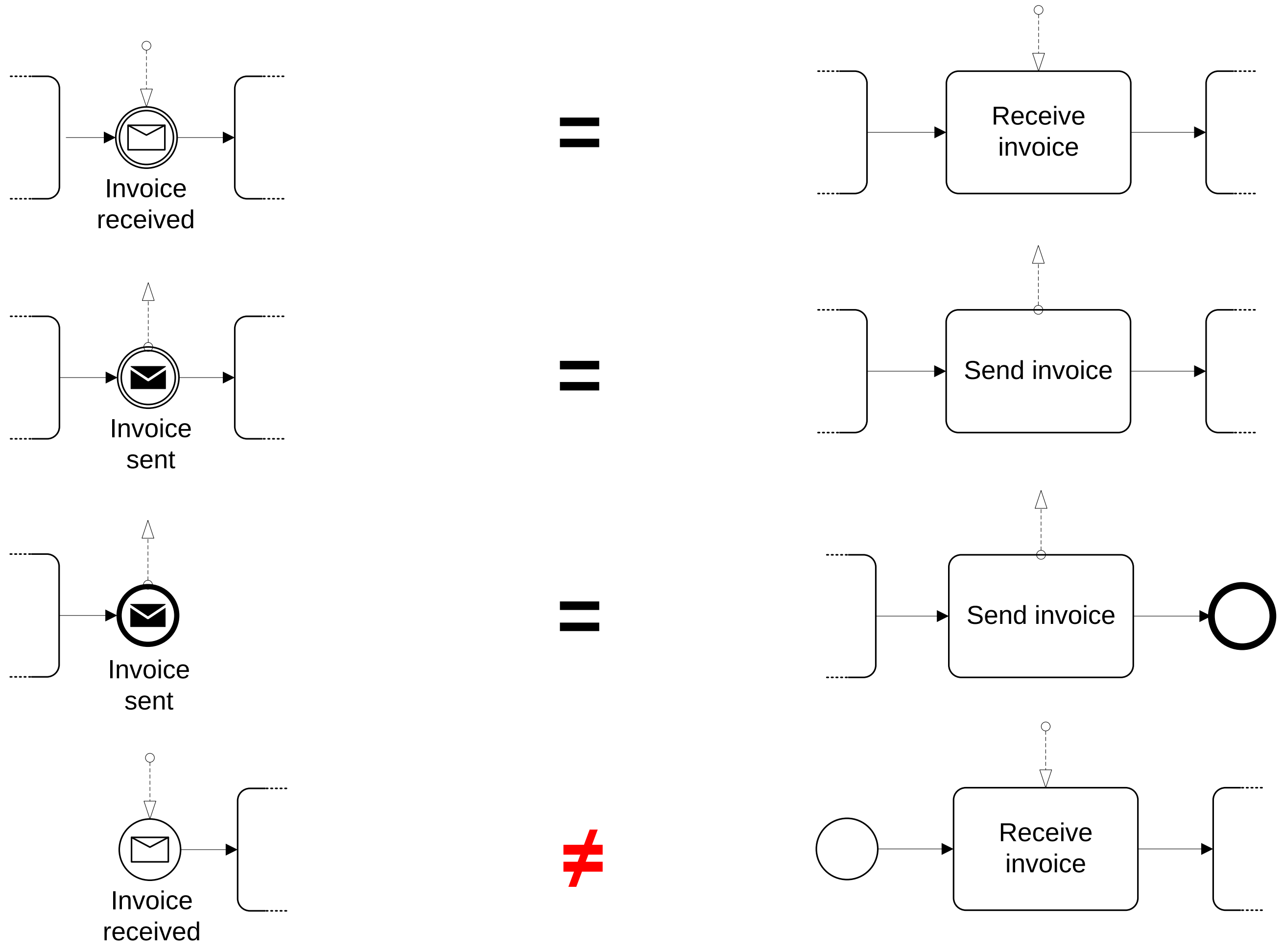
End Message Event – Indicates that an instance of the process is completed when a message is **sent**

Intermediate Message Event – Indicates that an event is expected to occur during the process. The event is triggered when a message is **received** or **sent**

Example: message events

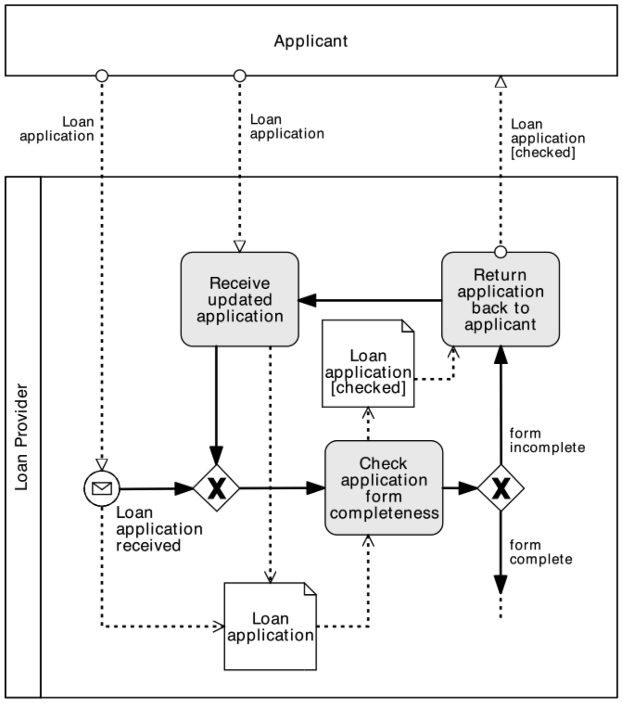


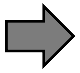
Comparison with sending/receiving tasks

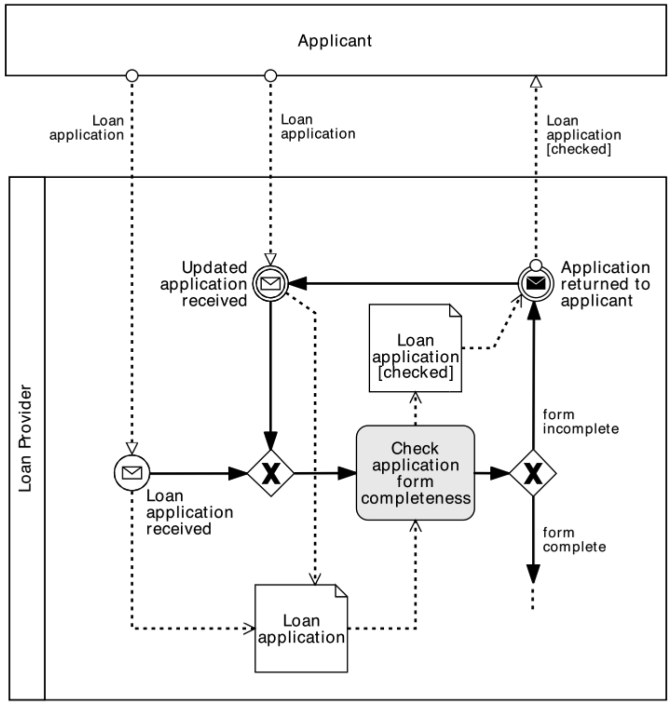


So, when to use what?

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

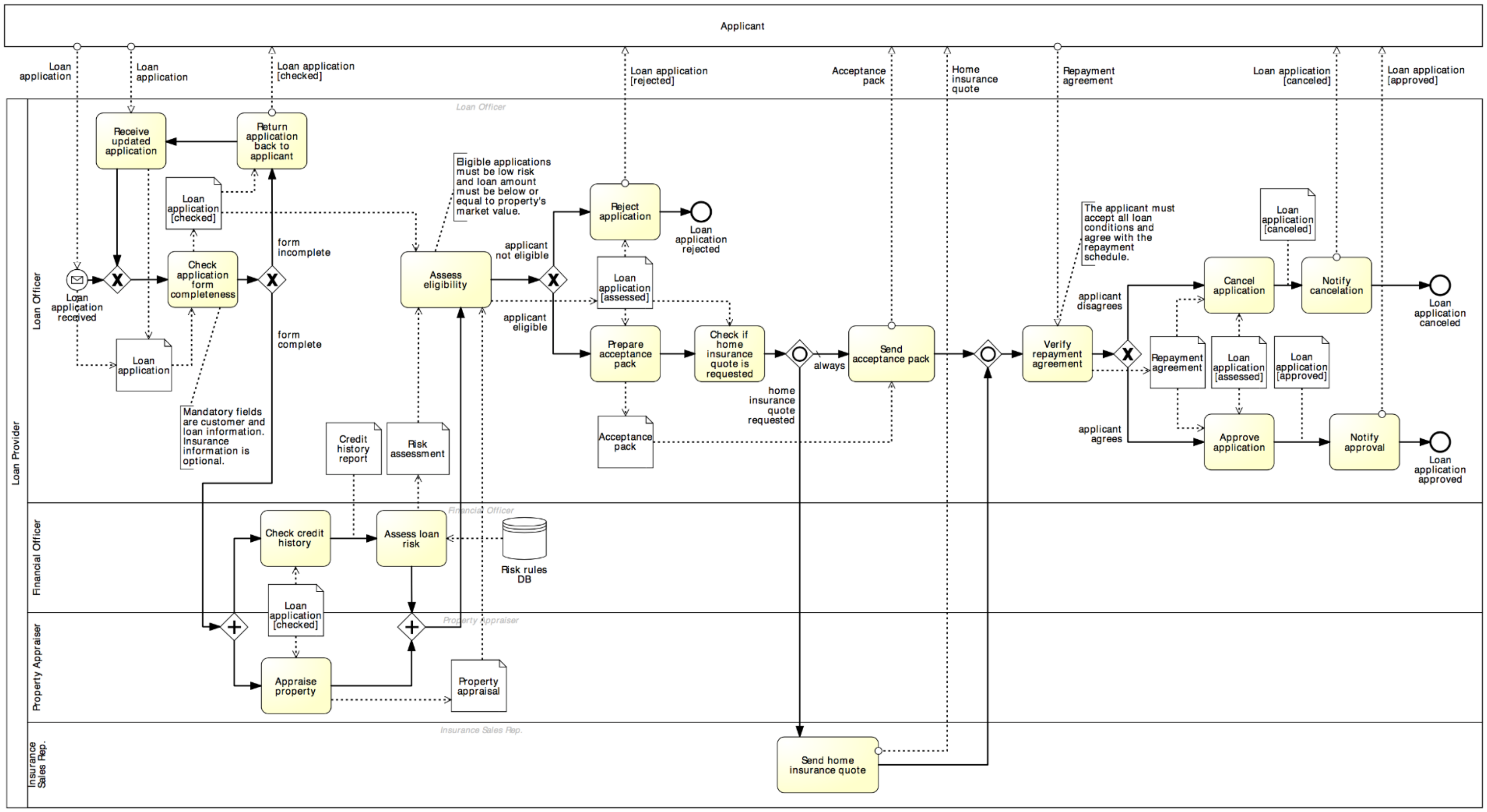

Applicant
Loan application
Loan application
Loan application [checked]
Loan Provider
Receive updated application
Return application back to applicant
Loan application [checked]
Check application form completeness
form incomplete
form complete
Loan application received
Loan application




Applicant
Loan application
Loan application
Loan application [checked]
Loan Provider
Updated application received
Application returned to applicant
Loan application [checked]
Check application form completeness
form incomplete
form complete
Loan application received
Loan application

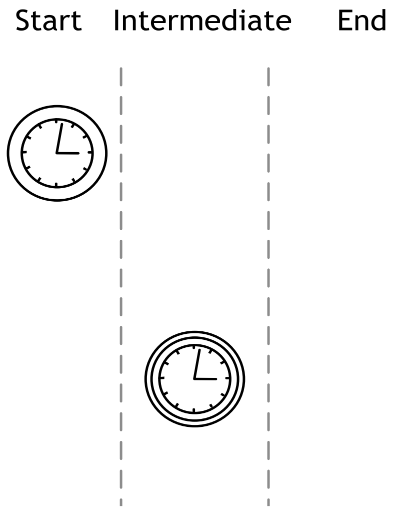
Typed or Untyped Event?

Exercise 4.4



Applicant
Loan application
Loan application
Loan application [checked]
Loan application [rejected]
Acceptance pack
Home insurance quote
Repayment agreement
Loan application [canceled]
Loan application [approved]
Loan Provider
Loan Officer
Receive updated application
Return application back to applicant
Loan application [checked]
Loan application received
Check application form completeness
form incomplete
form complete
Loan application
Mandatory fields are customer and loan information. Insurance information is optional.
Eligible applications must be low risk and loan amount must be below or equal to property's market value.
Assess eligibility
applicant not eligible
applicant eligible
Reject application
Loan application rejected
Loan application [assessed]
Prepare acceptance pack
Check if home insurance quote is requested
Acceptance pack
always
home insurance quote requested
Send acceptance pack
The applicant must accept all loan conditions and agree with the repayment schedule.
Verify repayment agreement
applicant disagrees
applicant agrees
Repayment agreement
Loan application [assessed]
Loan application [approved]
Loan application [canceled]
Cancel application
Notify cancelation
Loan application canceled
Approve application
Notify approval
Loan application approved
Credit history report
Risk assessment
Financial Officer
Check credit history
Assess loan risk
Risk rules DB
Loan application [checked]
Property Appraiser
Appraise property
Property appraisal
Insurance Sales Rep.
Send home insurance quote

Temporal events

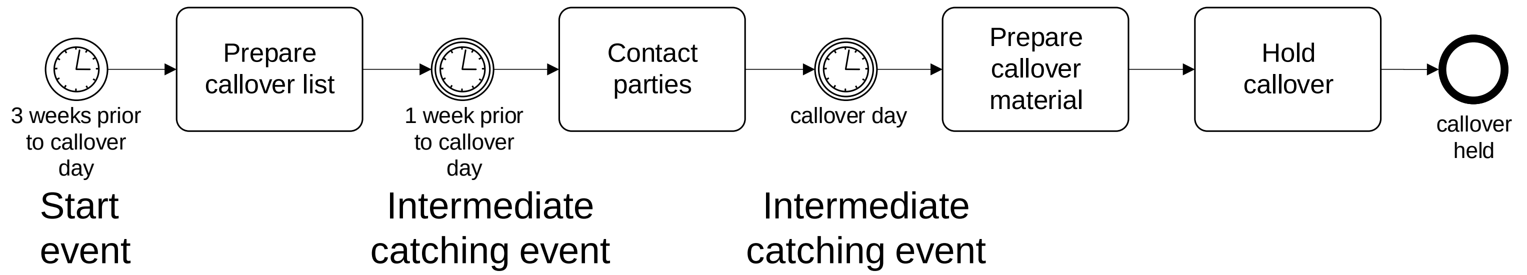


Start Timer Event – Indicates that an instance of the process is created at certain date(s)/time(s), e.g. start process at 6pm every Friday

Intermediate Timer Event – Triggered at certain date(s)/ time(s), or after a time interval has elapsed since the moment the event is “enabled” (delay)

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.

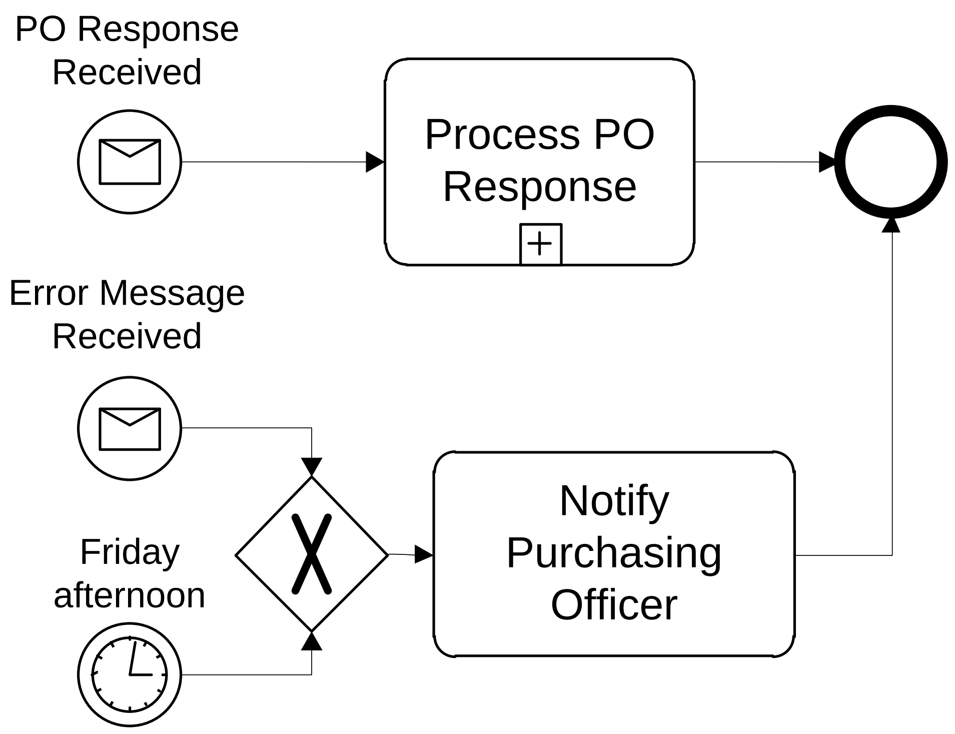

3 weeks prior to callover day
Prepare callover list
1 week prior to callover day
Contact parties
callover day
Prepare callover material
Hold callover
callover held
Start event
Intermediate catching event
Intermediate catching event

Coming back to our scenario…

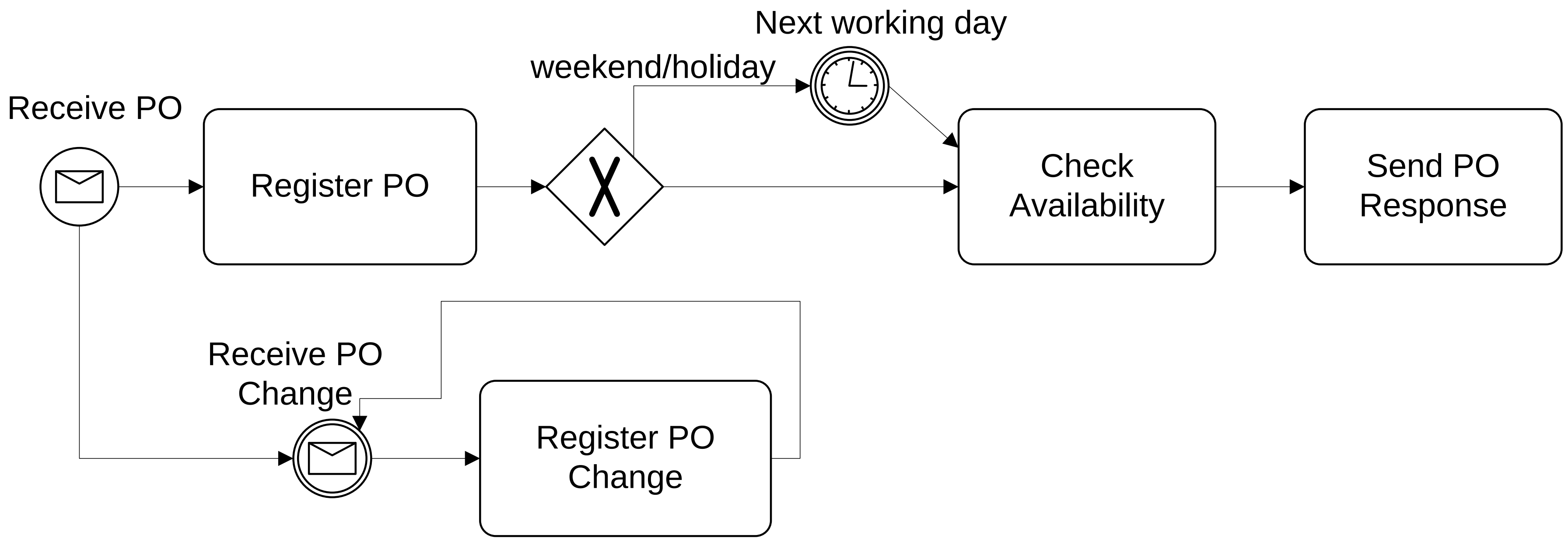
A Purchase Order (PO) handling process starts when a PO is received. The PO is first registered. If the current date is not a working day, the process waits until the following working day before proceeding. Otherwise, an availability check is performed and a PO response is sent back to the customer.

Multiple start events

The first start event that occurs will trigger an instance of the process



Modelling with events - Example



Exercise 4.4: Internet Service Provider

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.

Racing events: Event-based decision

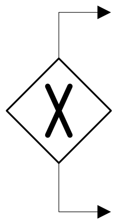
With the XOR-split gateway, a branch is chosen based on conditions that evaluate over available data

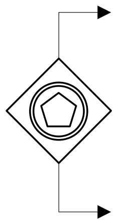
→ The choice can be made immediately after the token arrives from the incoming flow

Sometimes, the choice must be delayed until an event happens

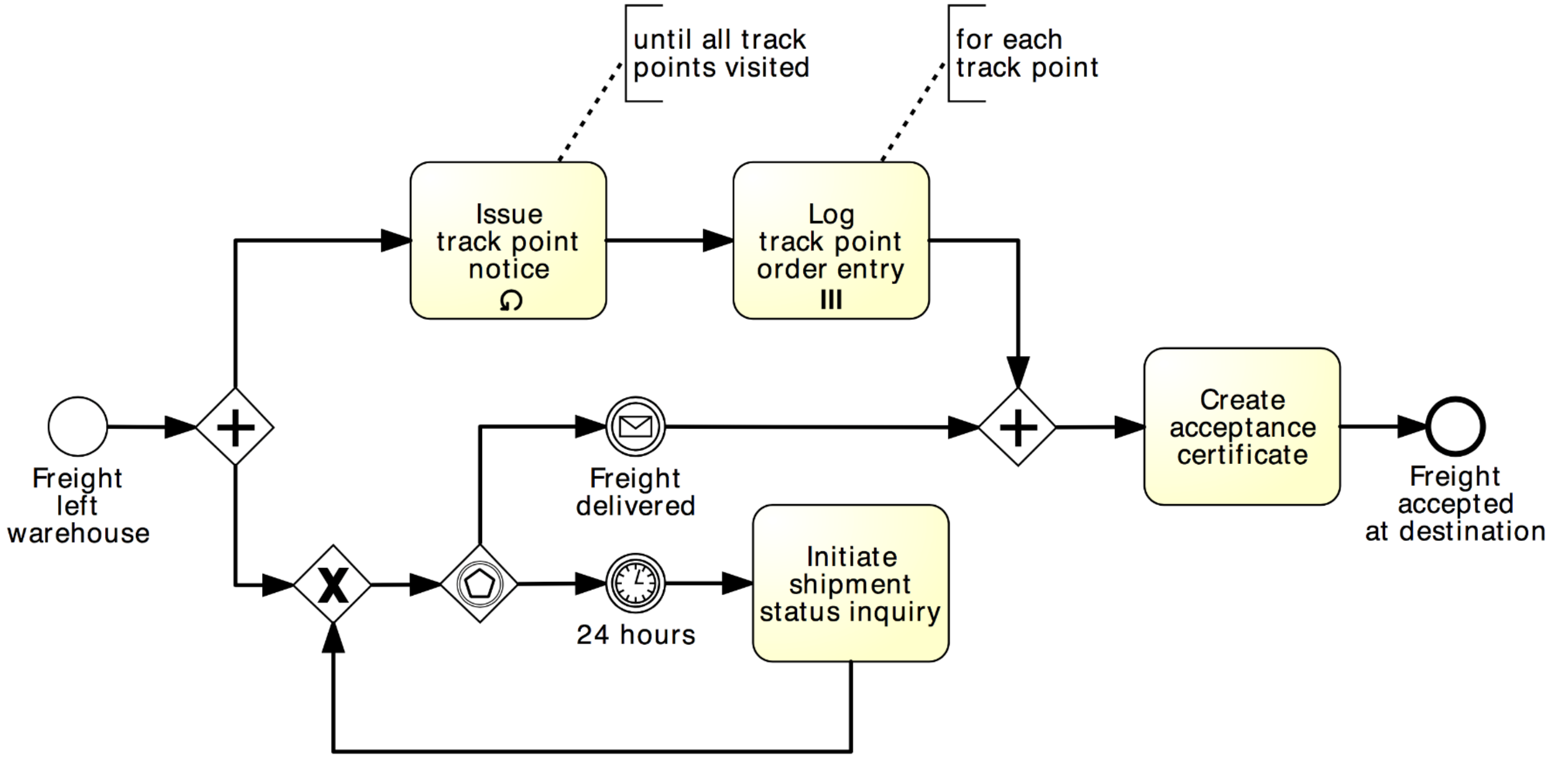
→ The choice is based on a “race” among events

Two types of XOR split:





Example: Event-based decision


until all track points visited
for each track point
Issue track point notice
Log track point order entry
Create acceptance certificate
Freight left warehouse
Freight delivered
Initiate shipment status inquiry
Freight accepted at destination
24 hours

Exercise

In the context of a claim handling process, it is sometimes necessary to send a questionnaire to claimant to gather additional information. Claimant is expected to return the questionnaire within five days. If no response is received after five days, a reminder is sent to claimant. If after another five days there is still no response, another reminder is sent and so on until the completed questionnaire is received.

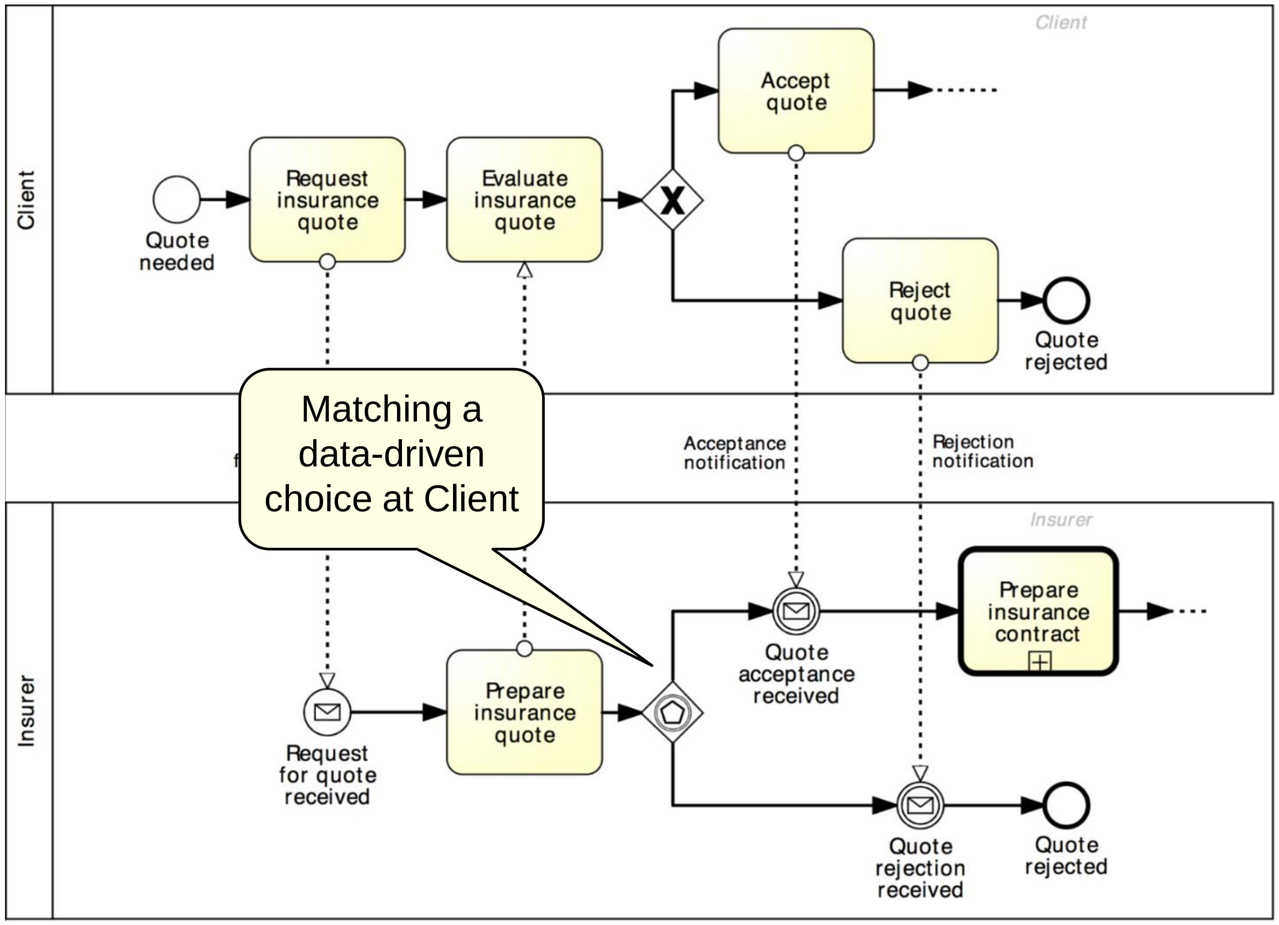
Exercise 4.6: Stock replenishment

Model the following process

A restaurant chain submits a purchase order (PO) to replenish its warehouses every Thursday. The restaurant chain's procurement system expects to receive either a "PO Response" or an error message. However, it may happen that no response is received at all due to system errors or due to delays in handling the PO on the supplier's side. If no response is received by Friday afternoon or if an error message is received, a purchasing officer at the restaurant chain's headquarters should be notified. Otherwise, the PO Response is processed normally.

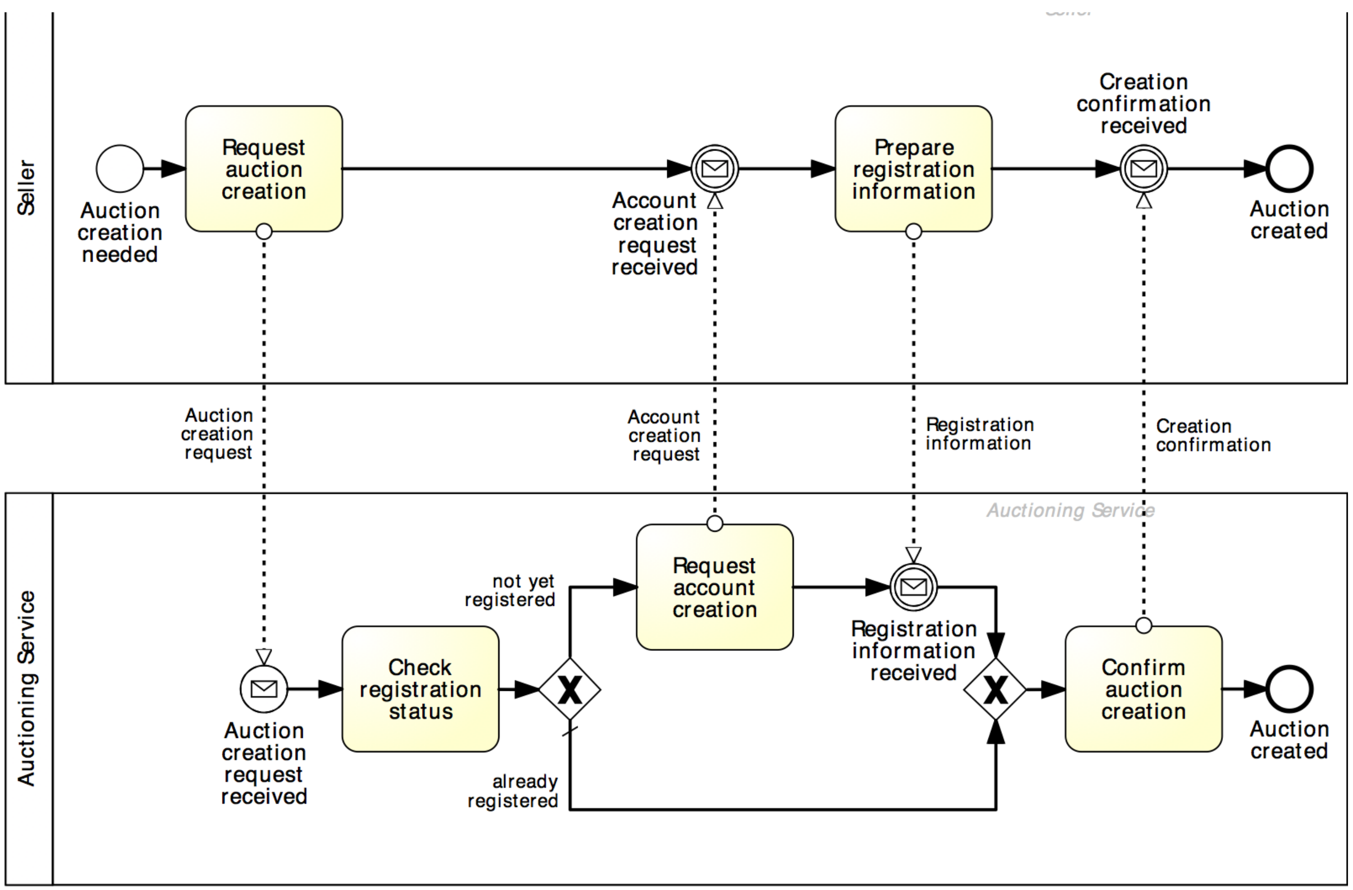
Matching choices in different business parties

Lead-to-Quote



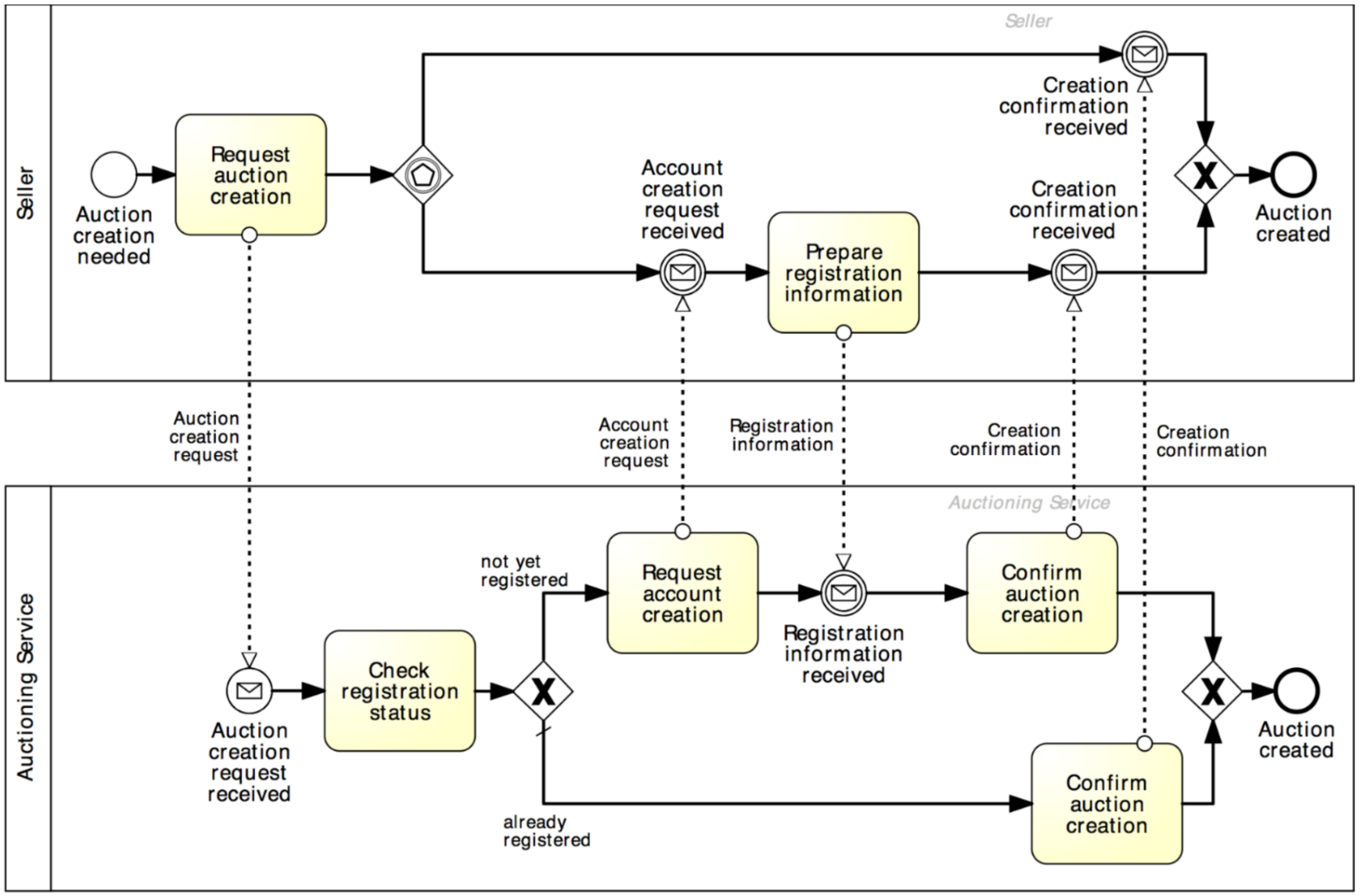
What’s wrong with this collaboration diagram?

Auction creation



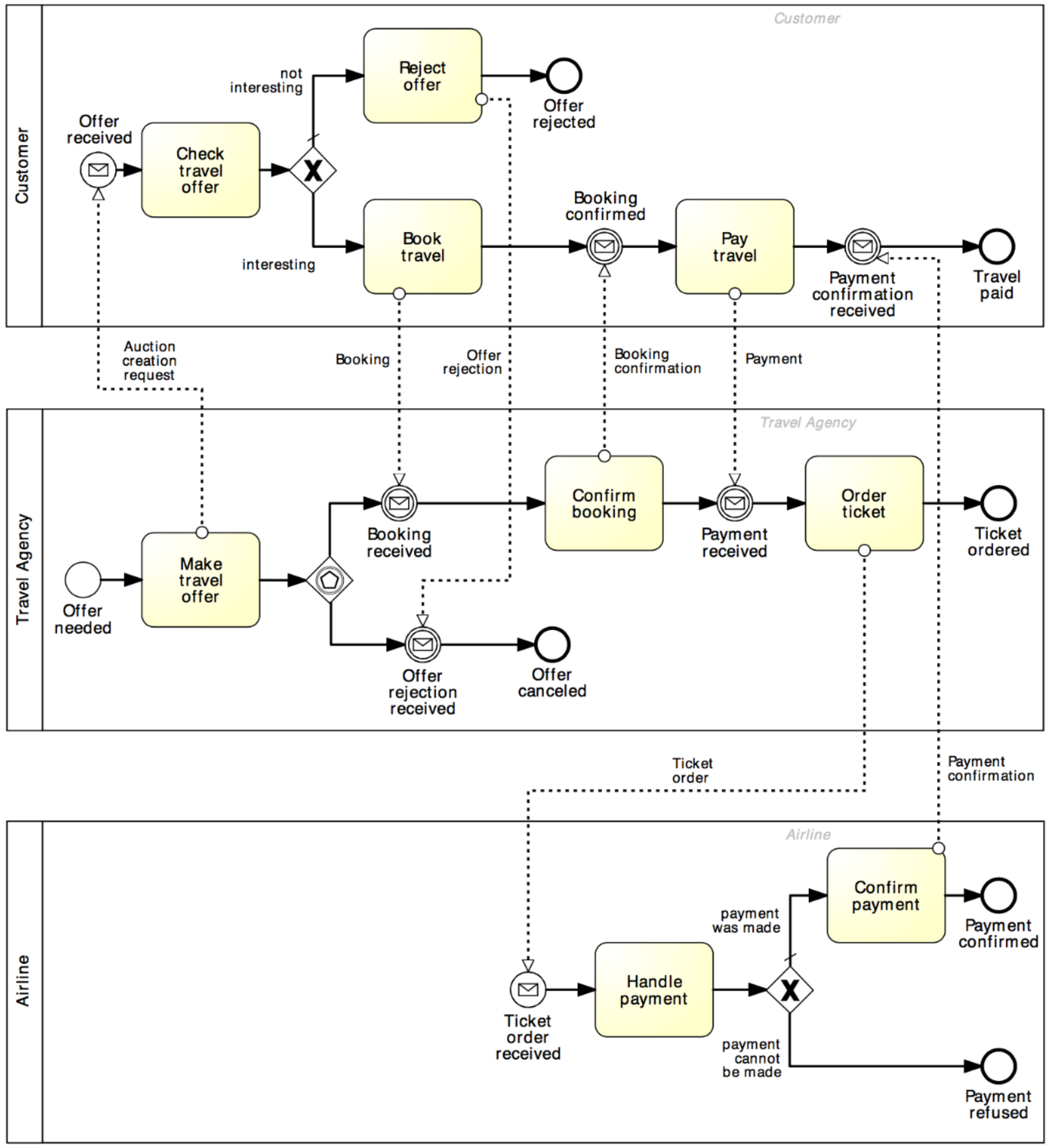
Solution

Auction creation

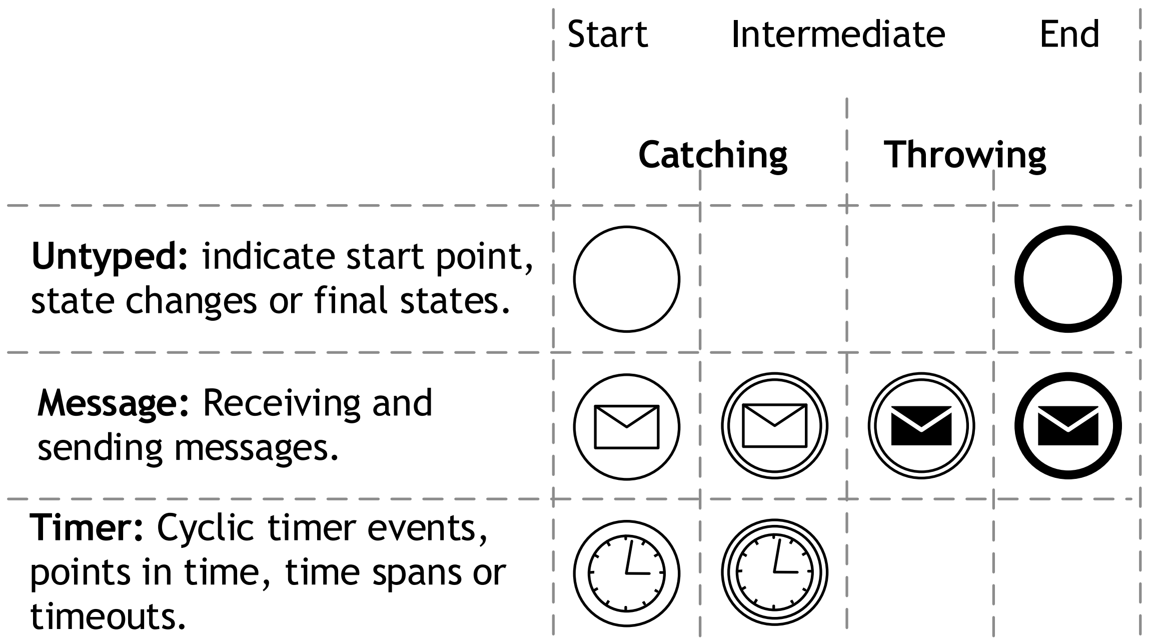


Exercise 4.7

Fix the following collaboration diagram.



Recap: message and timer events



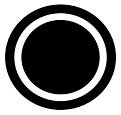
	Start	Intermediate		End
	Catching		Throwing	
Untyped: indicate start point, state changes or final states.	○			○
Message: Receiving and sending messages.	✉	✉	✉	✉
Timer: Cyclic timer events, points in time, time spans or timeouts.	🕒	🕒		

Process abortion

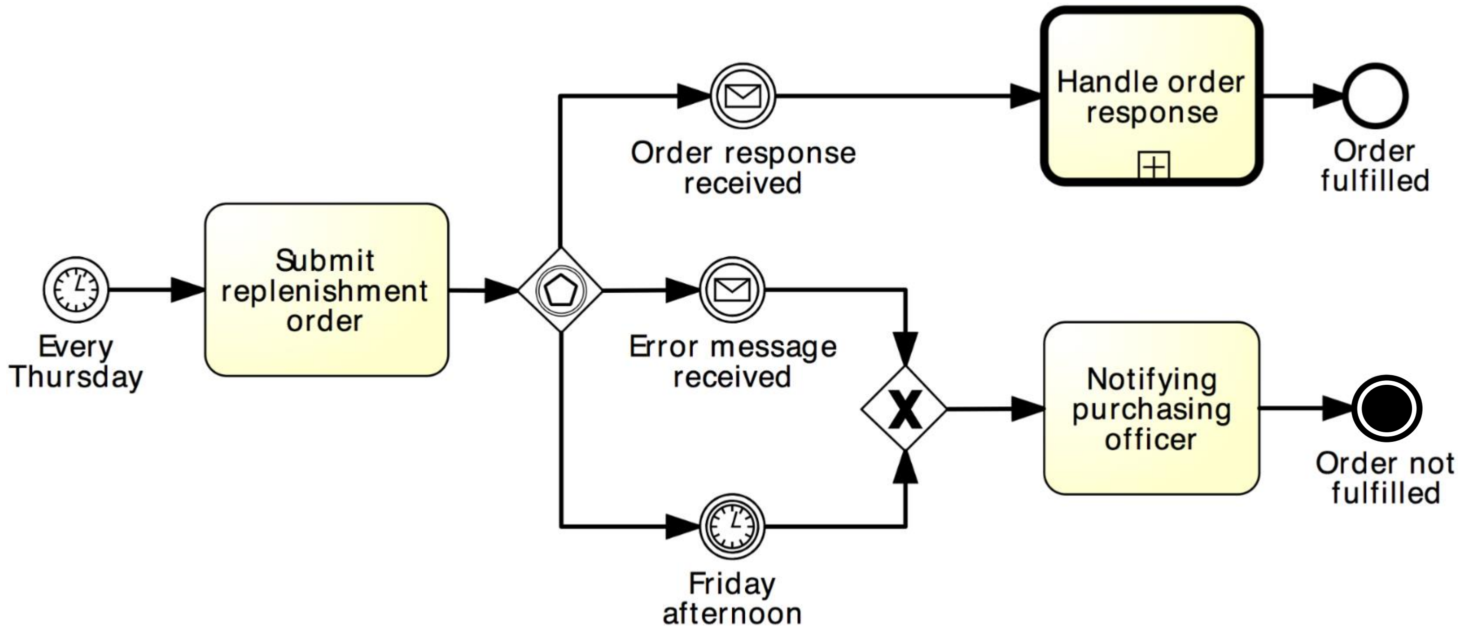
Exceptions are events that deviate a process from its “normal” course

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

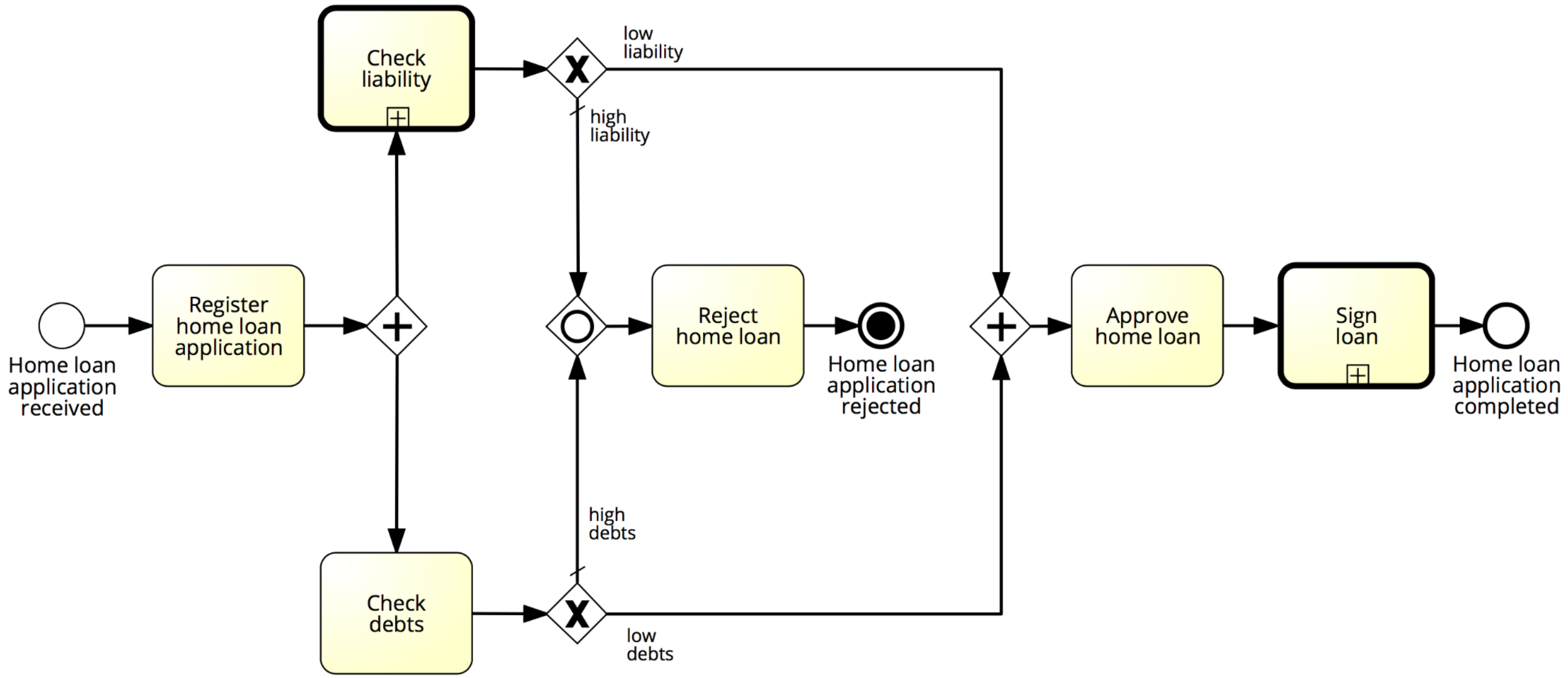
This can be done via the Terminate end event: it forces the whole process to *abort* (“wipes off” all tokens left behind, if any)



Example 1: terminate event



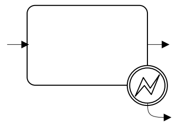
Example 2: terminate event


Check liability
low liability
high liability
Home loan application received
Register home loan application
Reject home loan
Home loan application rejected
Approve home loan
Sign loan
Home loan application completed
high debts
Check debts
low debts

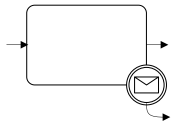
Exercise 4.8

Revise the examples presented so far by using the terminate event appropriately.

Exception handling



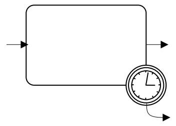
– **Internal**: something goes wrong inside an activity, whose execution must thus be interrupted. Handled with the Error event



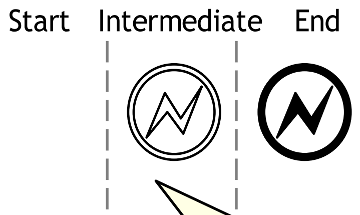
– **External**: something goes wrong outside the process, and execution of the current activity must be interrupted. Handled with Message event

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

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



Internal exception: error event



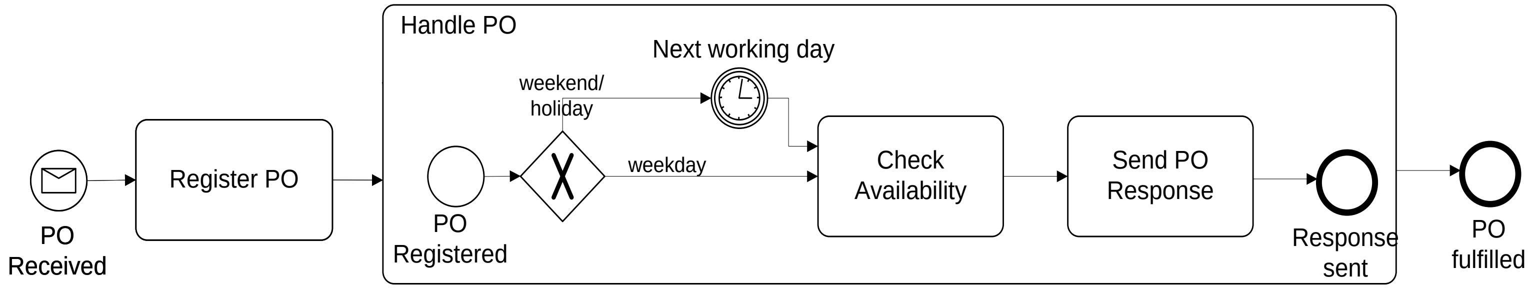
Error Event – Indicates an error: the "end" version generates an error event while the "catching intermediate" version consumes it when <u>attached</u> to the boundary of an activity

Must be attached to the activity's boundary

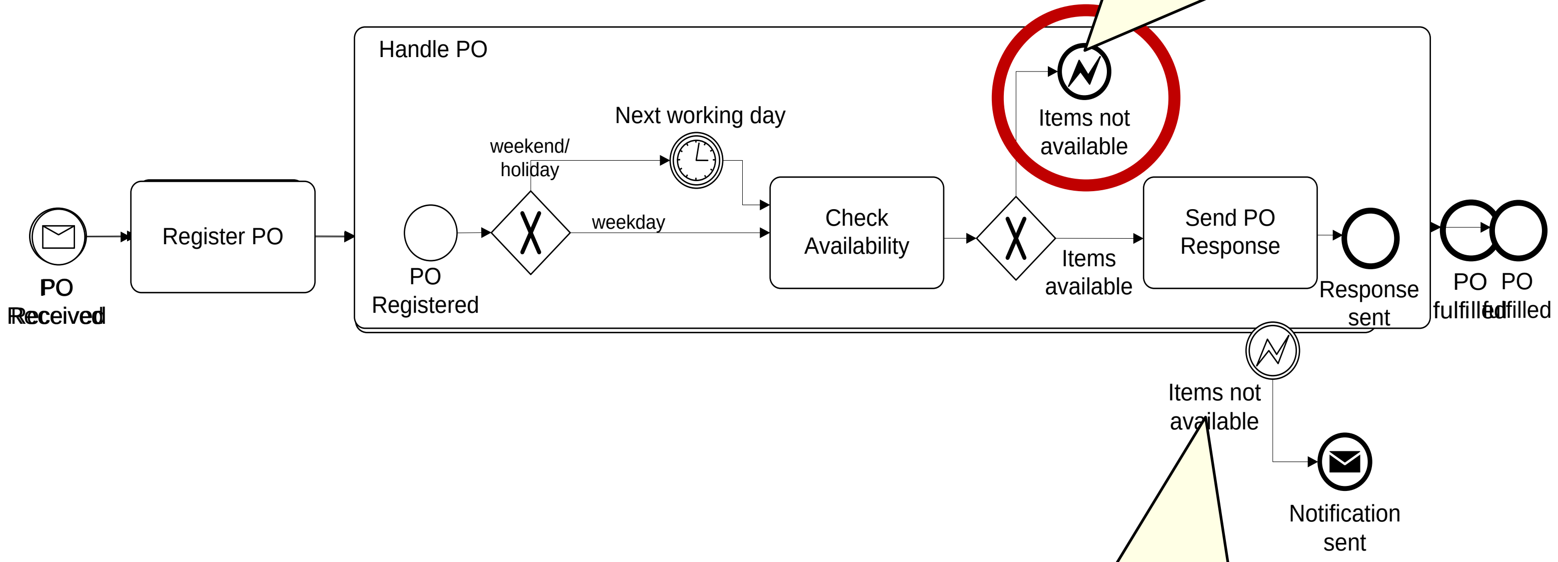
```
A problem has been detected and windows has been shut down to prevent damage
to your computer.

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to be sure you have adequate disk space. If a driver is
identified in the Stop message, disable the driver or check
with the manufacturer for driver updates. Try changing video
adapters.

Check with your hardware vendor for any BIOS updates. Disable
BIOS memory options such as caching or shadowing. If you need
to use Safe Mode to remove or disable components, restart your
computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.

Technical information:

*** STOP: 0x0000007E (0xC0000005,0x46069F2E,0xF79B0AC4,0xF79B07C0)


Beginning dump of physical memory
Physical memory dump complete.
Contact your system administrator or technical support group for further
assistance.
```

Example: internal exception

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, client needs to be notified that the PO cannot be further processed.



Solution: internal exception



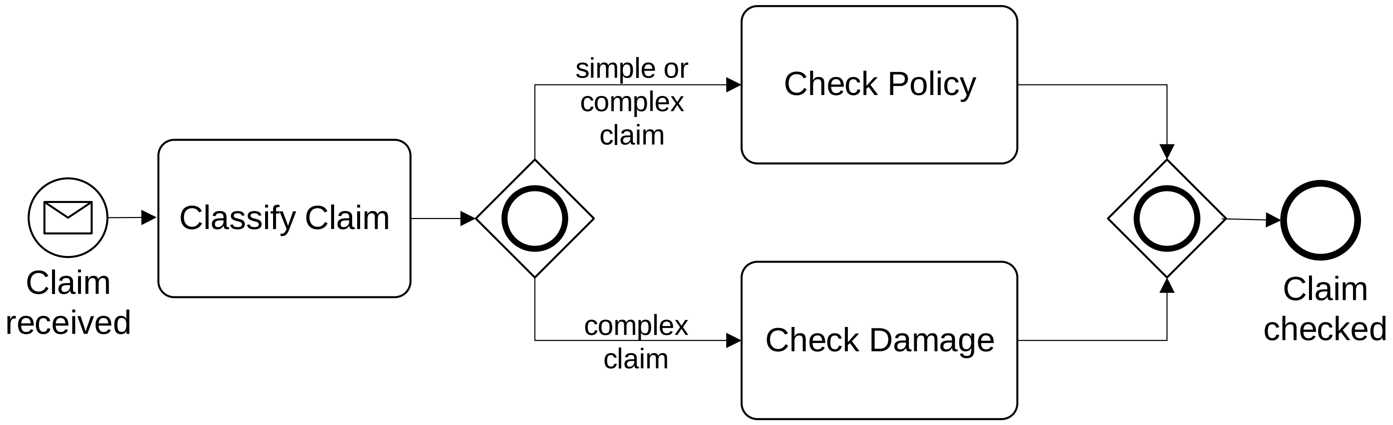
Throwing and catching error events must have the **same** label

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

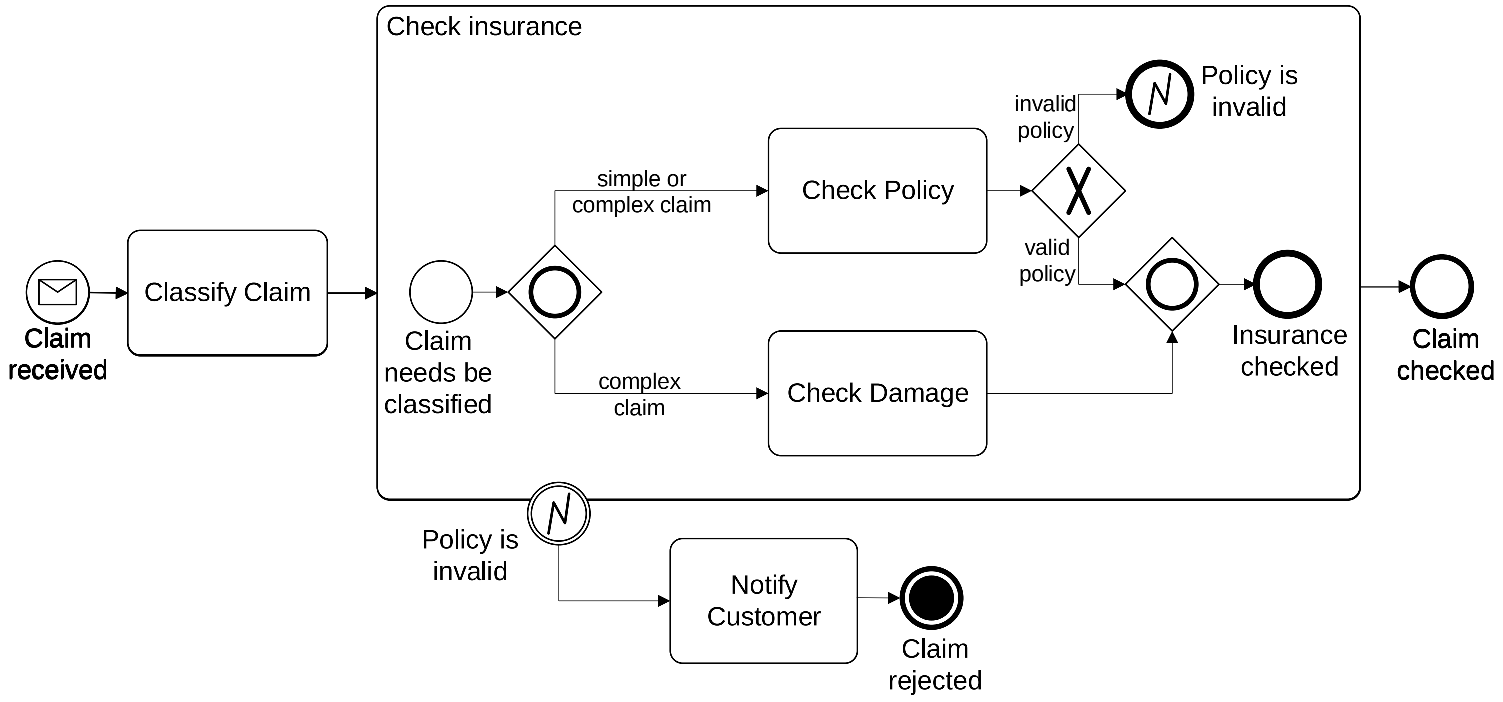
One more example: internal exception

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.



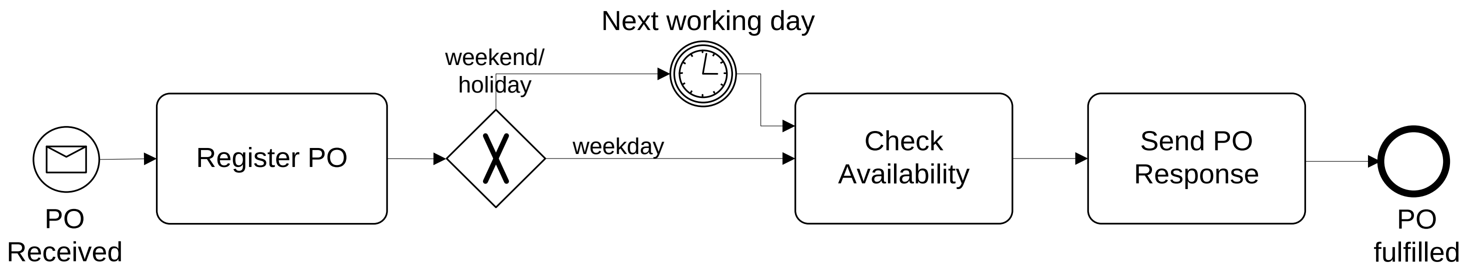
Solution: internal exception



Example: external exception

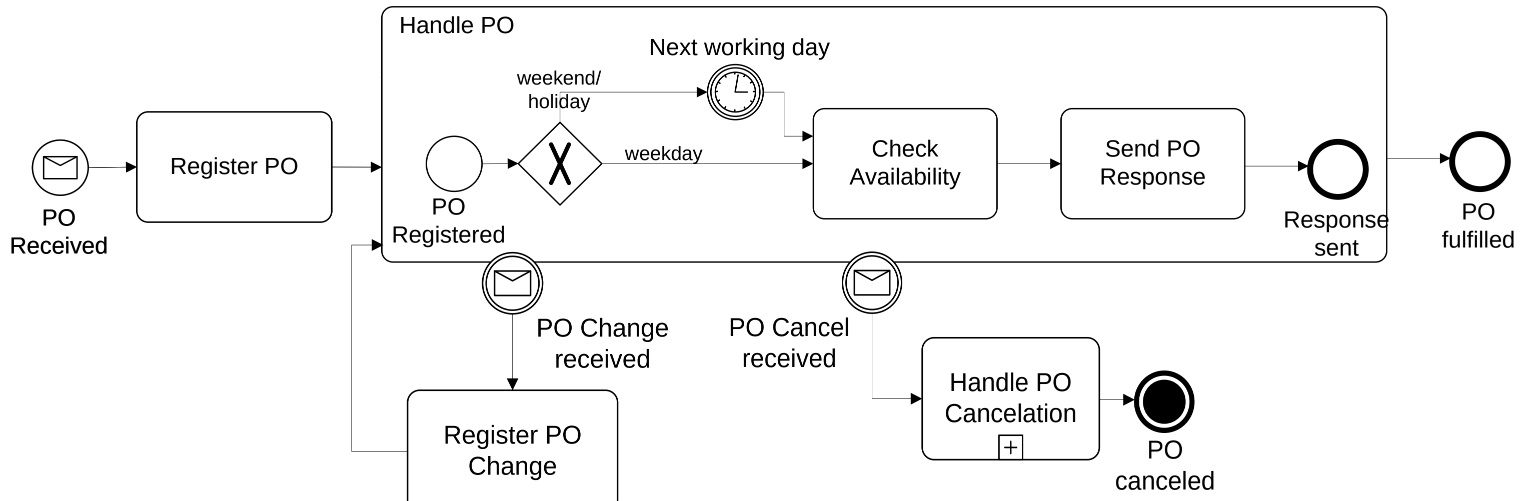
PO handling

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.

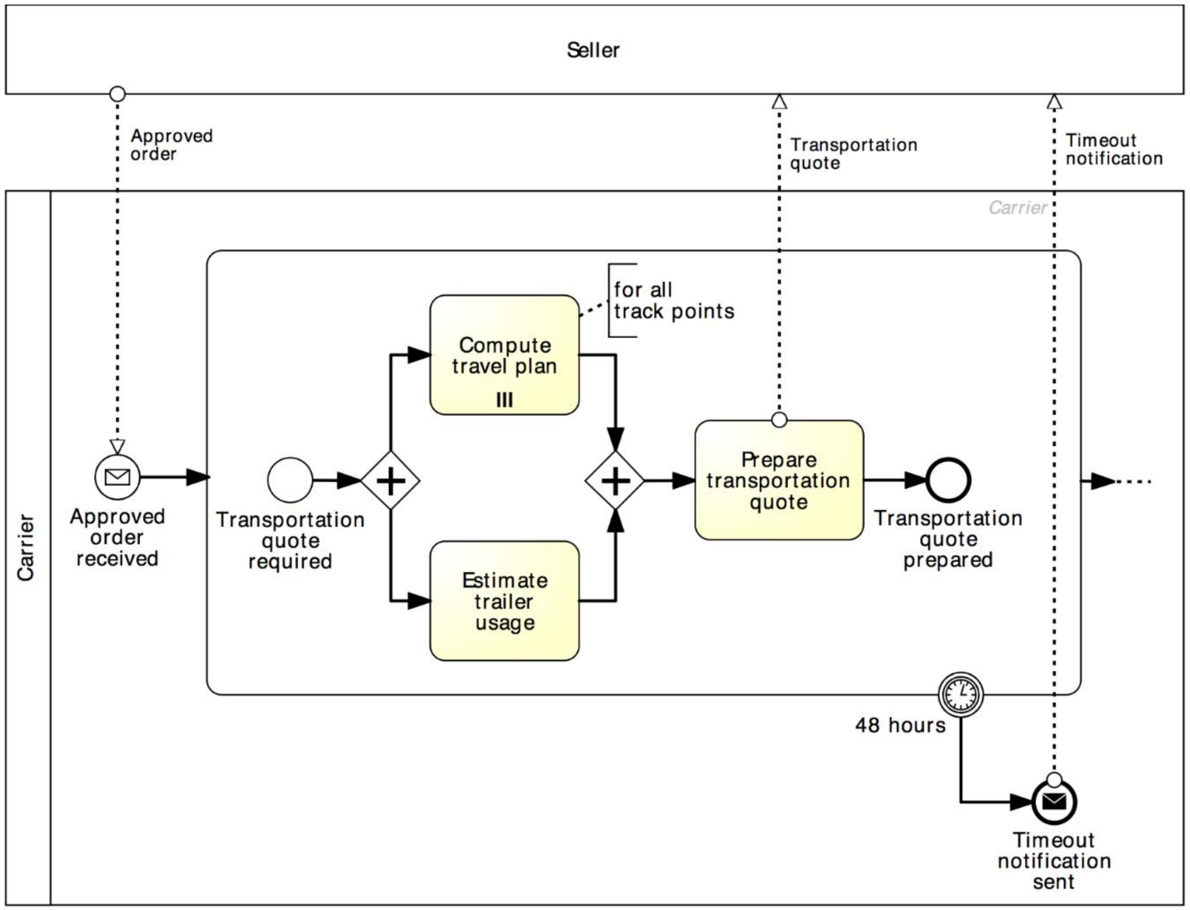
Exercise 4.10: activity timeout

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.

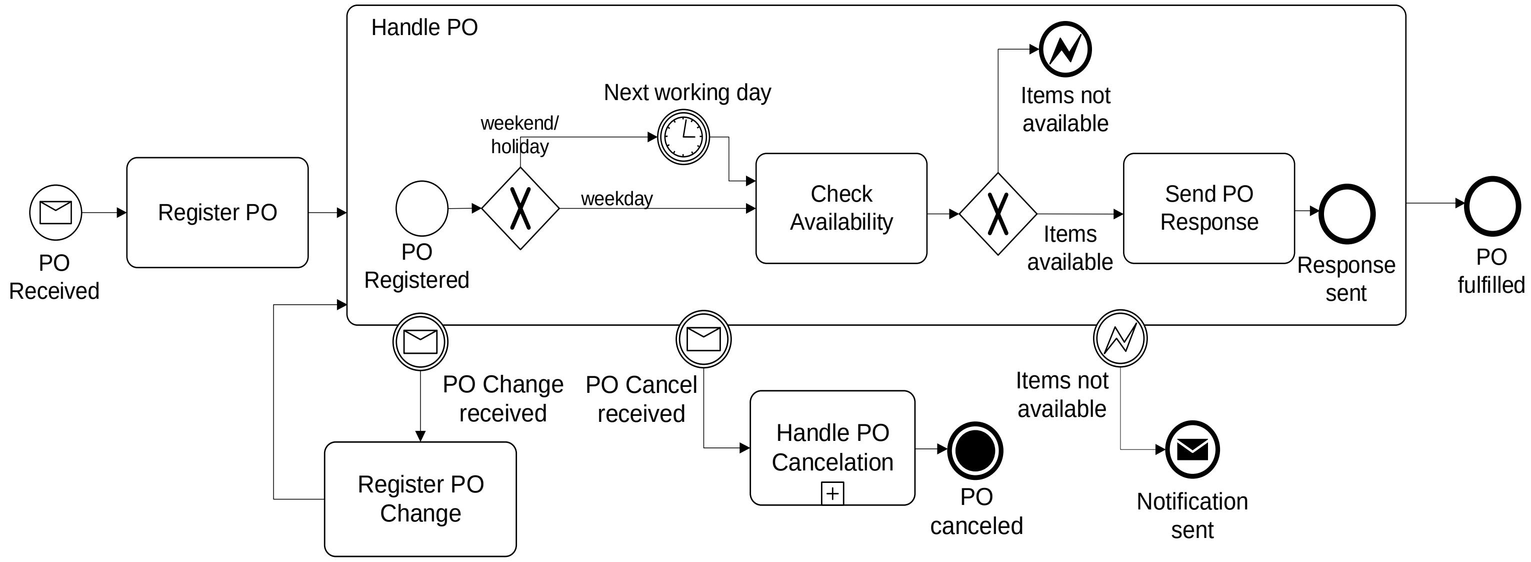
Solution: activity timeout

Order-to-transportation quote



Example: Non-interrupting boundary events

PO handling



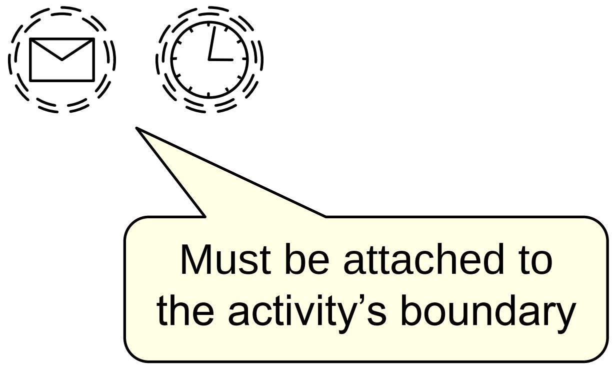
The customer may send a request for address change after the PO registration. When such a request is received, it is just registered, without further action.

Block5AdvancedModeling

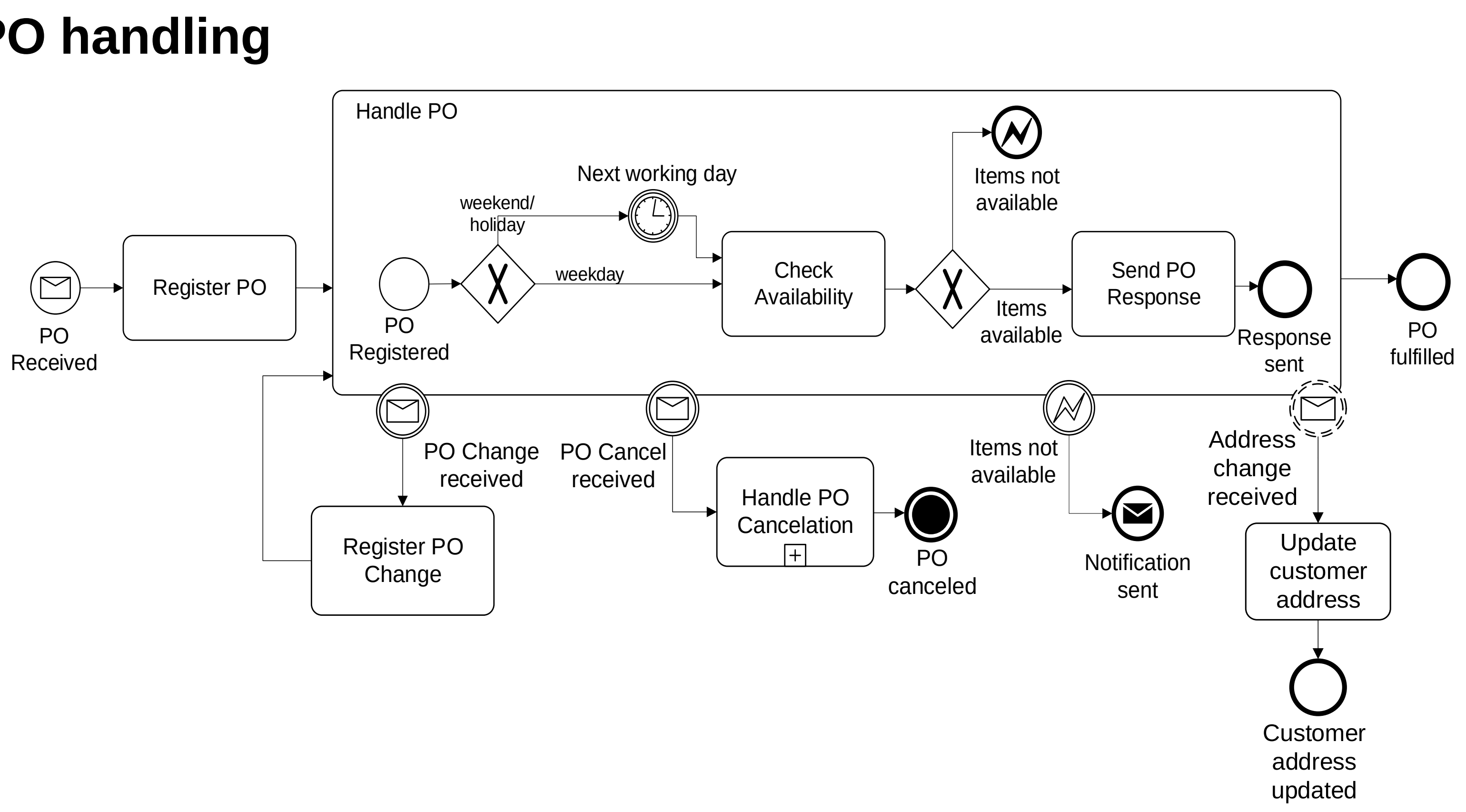
Non-interrupting boundary events

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

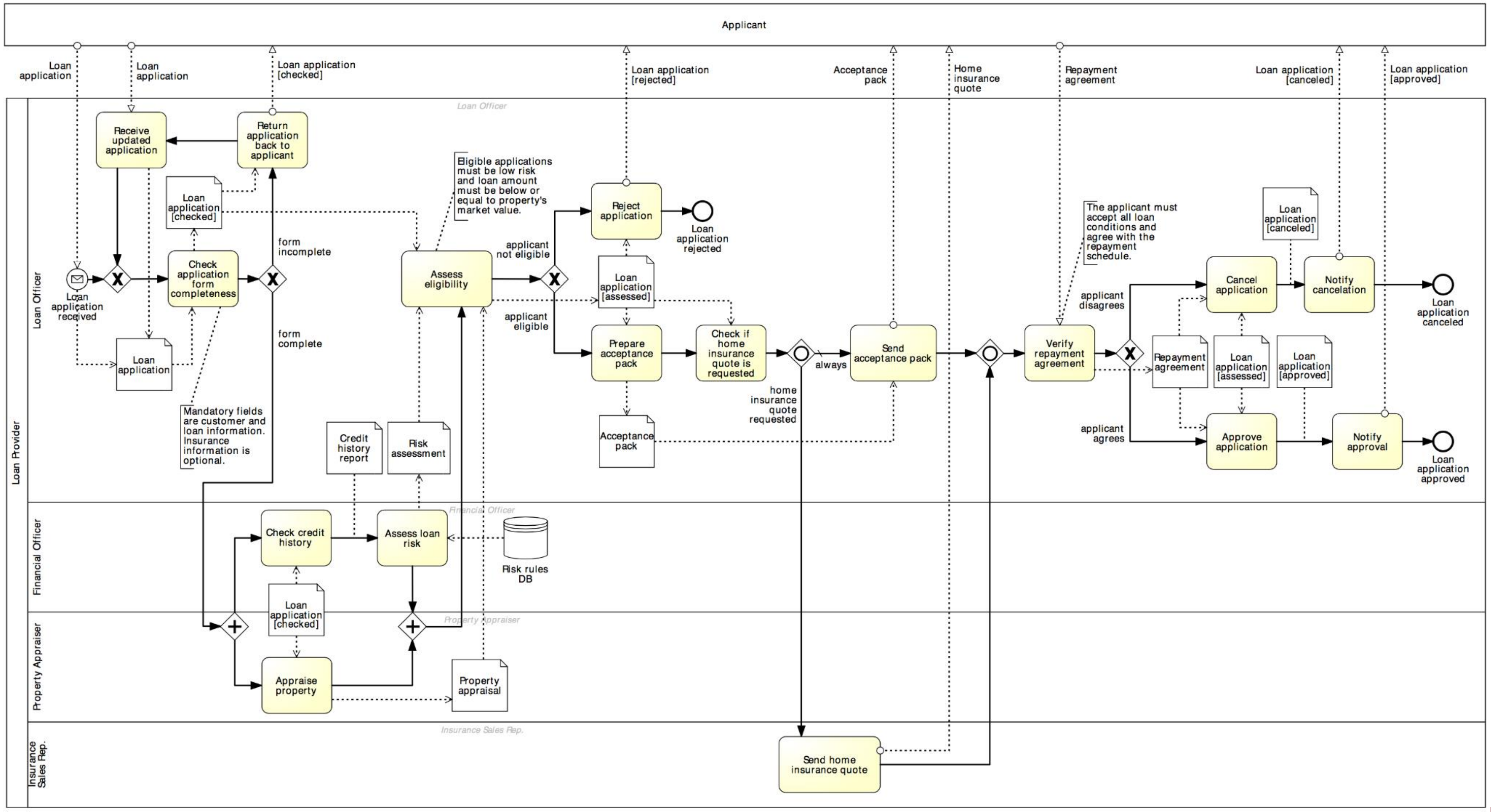


Solution: non-interrupting boundary events



Exercise 4.11

Consider the process for assessing loan applications below.


Applicant
Loan application
Loan application
Loan application [checked]
Loan application [rejected]
Acceptance pack
Home insurance quote
Repayment agreement
Loan application [canceled]
Loan application [approved]
Loan Provider
Loan Officer
Receive updated application
Return application back to applicant
Loan application [checked]
Loan application received
Check application form completeness
form incomplete
form complete
Loan application
Mandatory fields are customer and loan information. Insurance information is optional.
Eligible applications must be low risk and loan amount must be below or equal to property's market value.
Assess eligibility
applicant not eligible
applicant eligible
Reject application
Loan application rejected
Loan application [assessed]
Prepare acceptance pack
Check if home insurance quote is requested
Acceptance pack
always
home insurance quote requested
Send acceptance pack
The applicant must accept all loan conditions and agree with the repayment schedule.
Verify repayment agreement
applicant disagrees
applicant agrees
Cancel application
Notify cancelation
Loan application [canceled]
Loan application canceled
Repayment agreement
Loan application [assessed]
Loan application [approved]
Approve application
Notify approval
Loan application approved
Credit history report
Risk assessment
Financial Officer
Check credit history
Assess loan risk
Risk rules DB
Loan application [checked]
Property Appraiser
Appraise property
Property appraisal
Insurance Sales Rep.
Send home insurance quote

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.

Complex Exceptions

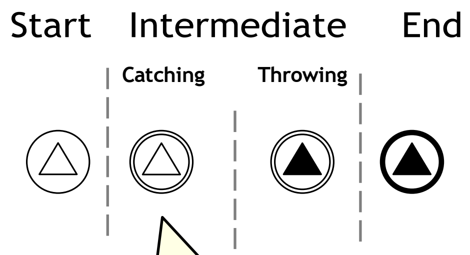
PO handling

[…] Further, if the customer sends a PO cancelation request after the PO registration, the PO processing must be stopped and the change request needs to be handled.

[…] Further, if the customer sends a PO cancelation request after the PO registration, the request is first assessed. Based on the progress of the order handling (e.g. if items have already been fetched from external warehouses) a penalty is determined and communicated to the customer. The customer has 48 hours to accept the penalty and go on with the cancelation or to stop the cancelation and continue with the order handling.

Complex Exceptions: Signal event

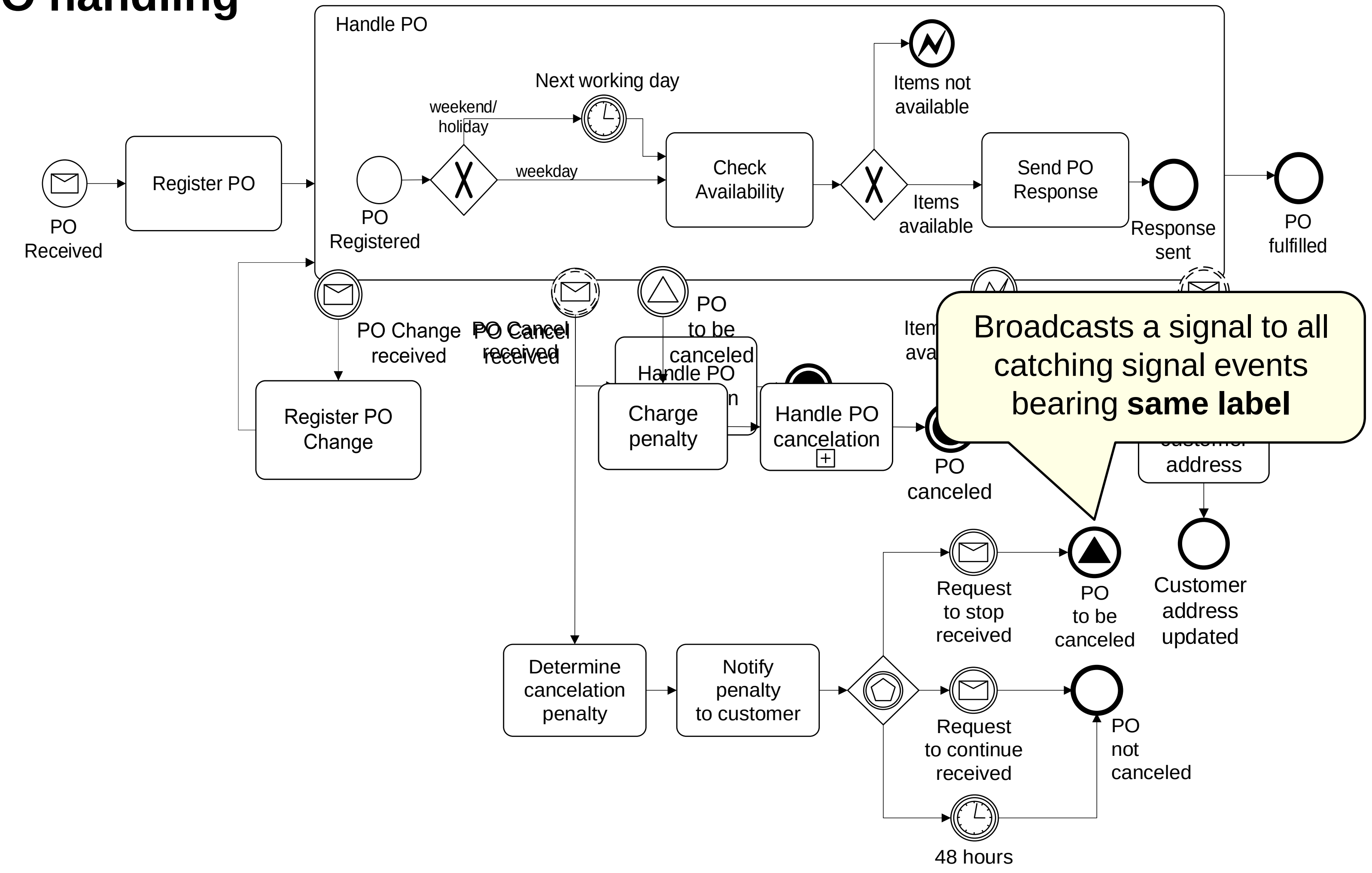
- To send or receive a signal for synchronization purposes
- A signal is broadcasted without any specific target. Thus it can be caught multiple times
- Signals are different than messages which are routed to a specific target



Can be attached to the activity's boundary

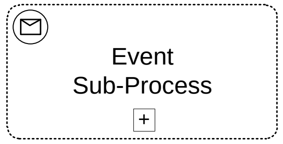
Solution: non-interrupting event + signal event

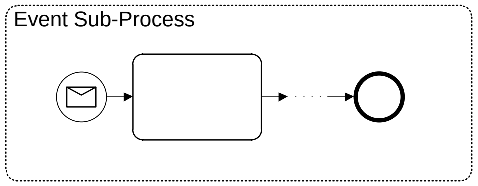
PO handling



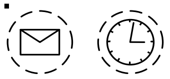
Event sub-process

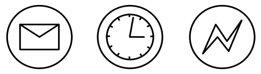
To handle events that may not refer to a particular task/ sub-process within a process





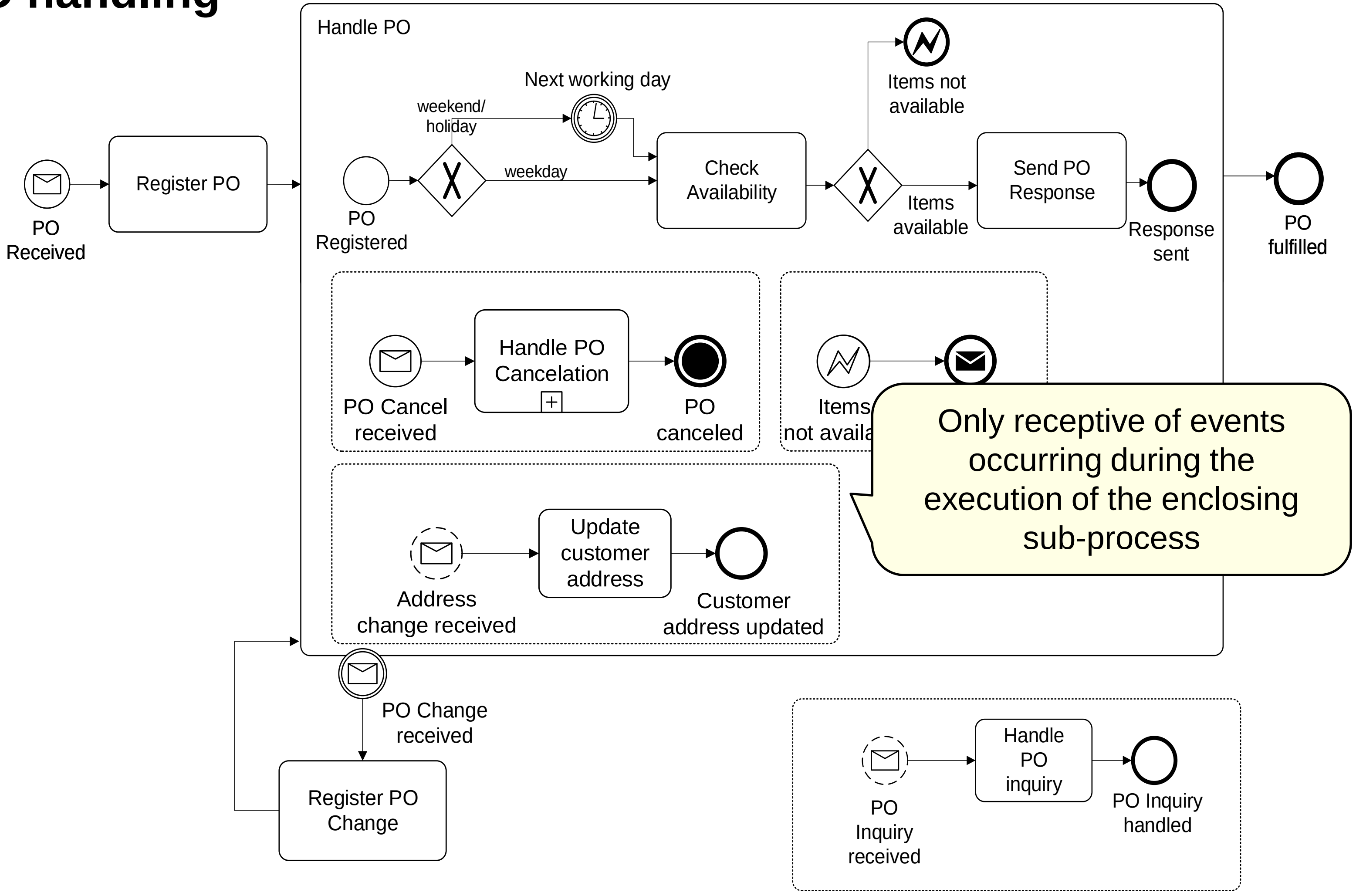
- Placed into a process or subprocess
- Is activated when its start event is triggered
- It may or may not interrupt the parent process or sub-process, depending on the type of its start event:
  - Non-Interrupting:
  - Interrupting:





Example: event sub-process

PO handling



Only receptive of events occurring during the execution of the enclosing sub-process

Event sub-processes or boundary events?

Exercise 4.12

Model the following business process for reimbursing expenses.

After an expense report is received from an employee, the employee is notified of the receipt of the report. Next, a new account must be created if the employee does not already have one. The report is then reviewed for automatic approval. Amounts under EUR 1,000 are automatically approved while amounts equal to or over EUR 1,000 require manual approval. In case of rejection, the employee must receive a rejection notice by email. In case of approval, the reimbursement is deposited directly to the employee's bank account and an approval notice is sent to the employee via email, with the details of the money transfer. At any time during the review, the employee can send a request for amount rectification. In that case the rectification is registered and the report needs to be reviewed again. Moreover, if the report is not handled within 30 days, the process is stopped and the employee receives a cancelation notice email so that he can resubmit the expense report from scratch.

Example: activity compensation

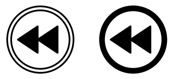
PO handling

After a PO has been registered, checked, and a response has been sent back, a payment sub-process and a fulfilment sub-process are started. During these two sub-processes, a PO cancellation may be received from the customer. In this case, both sub-processes are stopped, and their effects are reverted (e.g. reimbursement and/or goods return may need to occur).

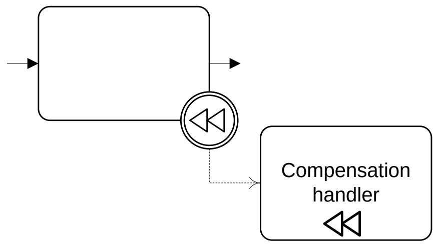
Activity compensation

- Rollback of completed process activities
- May be used as part of an exception handling procedure

- Triggered by throwing Compensate event

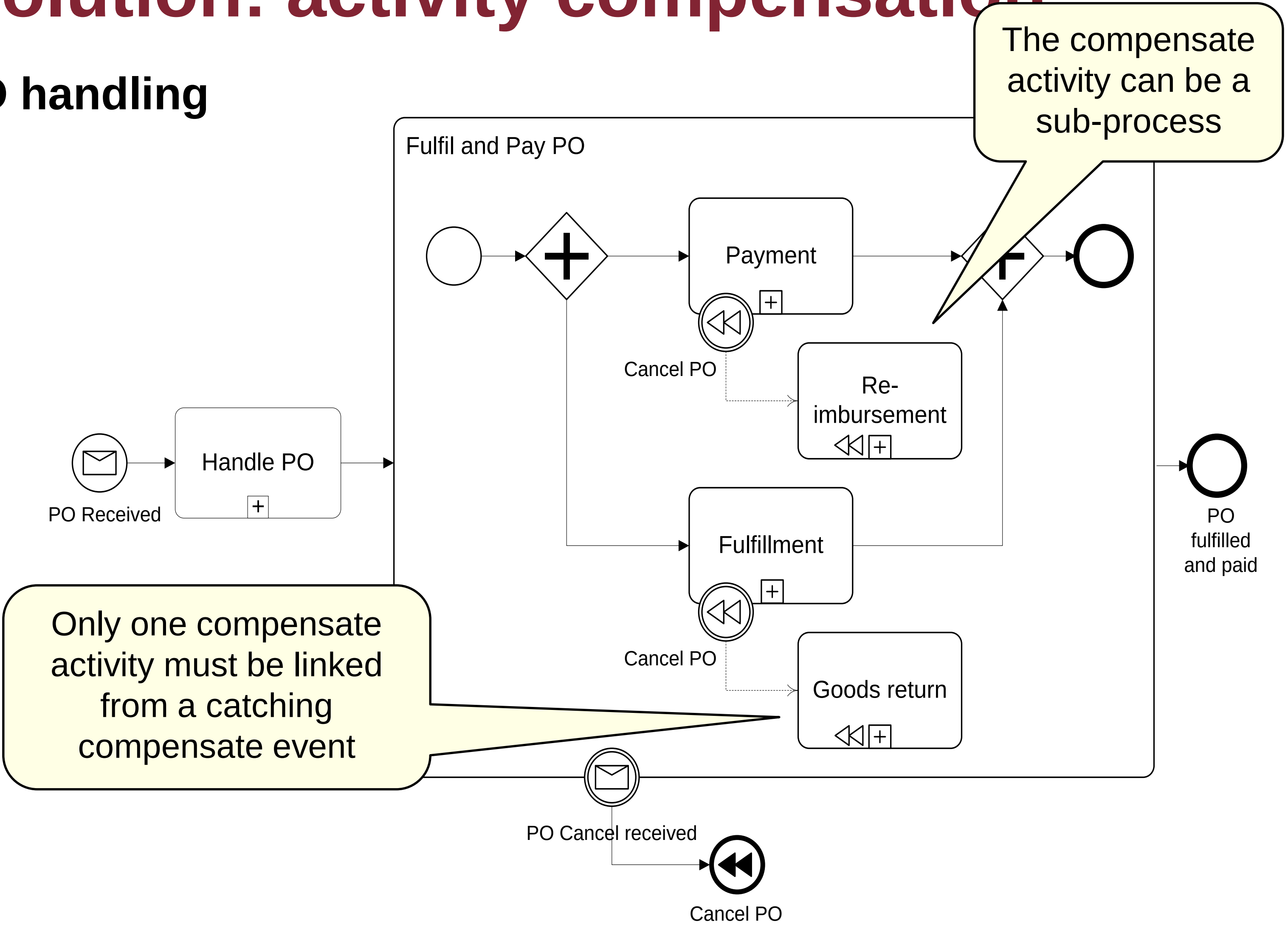


- Compensation Handler (a catching Compensate event + a Compensate activity) performs the rollback



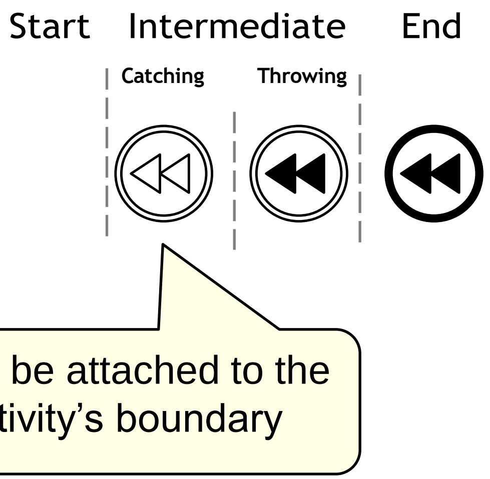
Solution: activity compensation

PO handling



Compensate event

- Indicates that the enclosing activity must be compensated
- The "throwing intermediate" and the "end" version generate the compensation event
- The "intermediate catching" version triggers the compensation when attached to the boundary of an activity

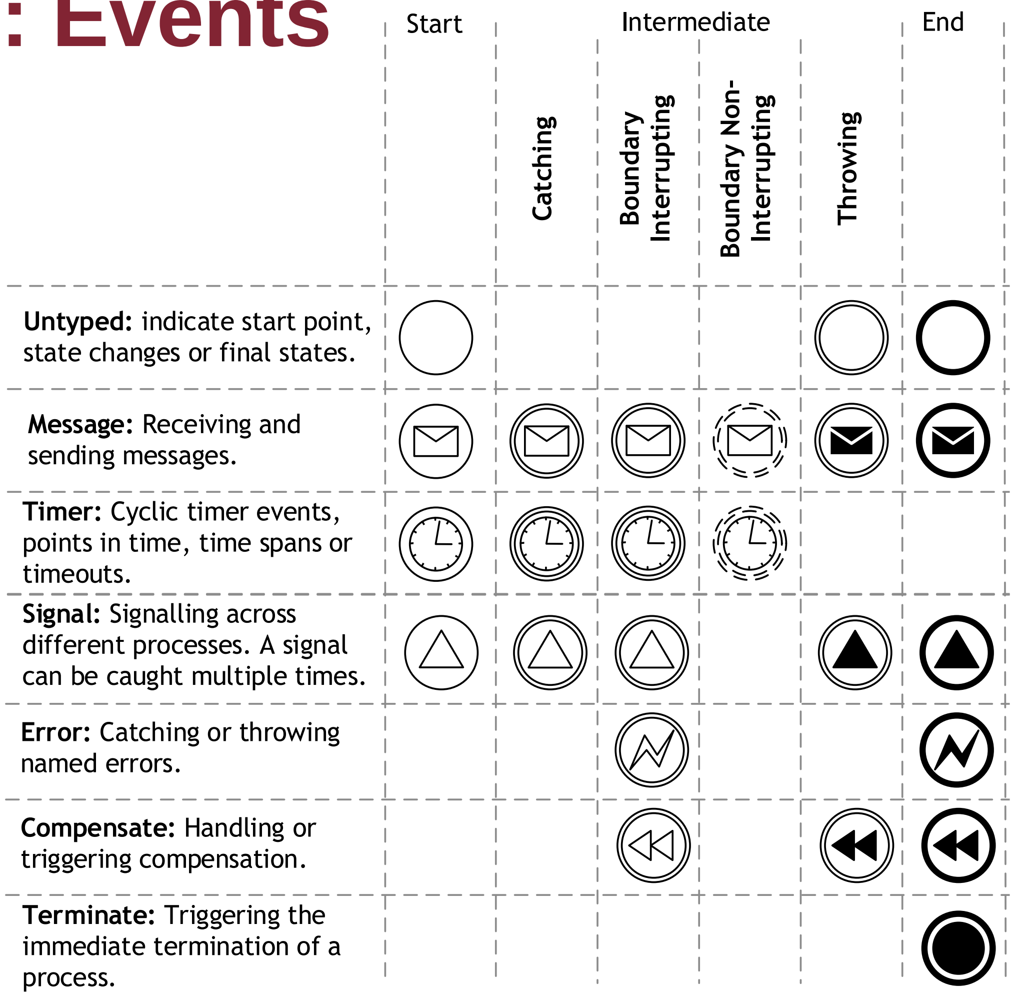


Exercise 4.13

Modify the model created in Exercise 4.13 as follows.

If the report is not handled within 30 days, the process is stopped, the employee receives a cancelation notice email and must resubmit the expense report. However, if the reimbursement for the employee’s expenses had already been made, a money recall needs to be made, to get the money back from the employee, before sending the cancelation notice email.

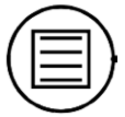
Recap: Events



	Start	Intermediate				End
		Catching	Boundary Interrupting	Boundary Non-Interrupting	Throwing	
Untyped: indicate start point, state changes or final states.	○				◎	○
Message: Receiving and sending messages.	✉	✉	✉	✉	✉	✉
Timer: Cyclic timer events, points in time, time spans or timeouts.	🕒	🕒	🕒	🕒		
Signal: Signalling across different processes. A signal can be caught multiple times.	△	△	△		▲	▲
Error: Catching or throwing named errors.			ϟ			ϟ
Compensate: Handling or triggering compensation.			⏪		⏪	⏪
Terminate: Triggering the immediate termination of a process.						●

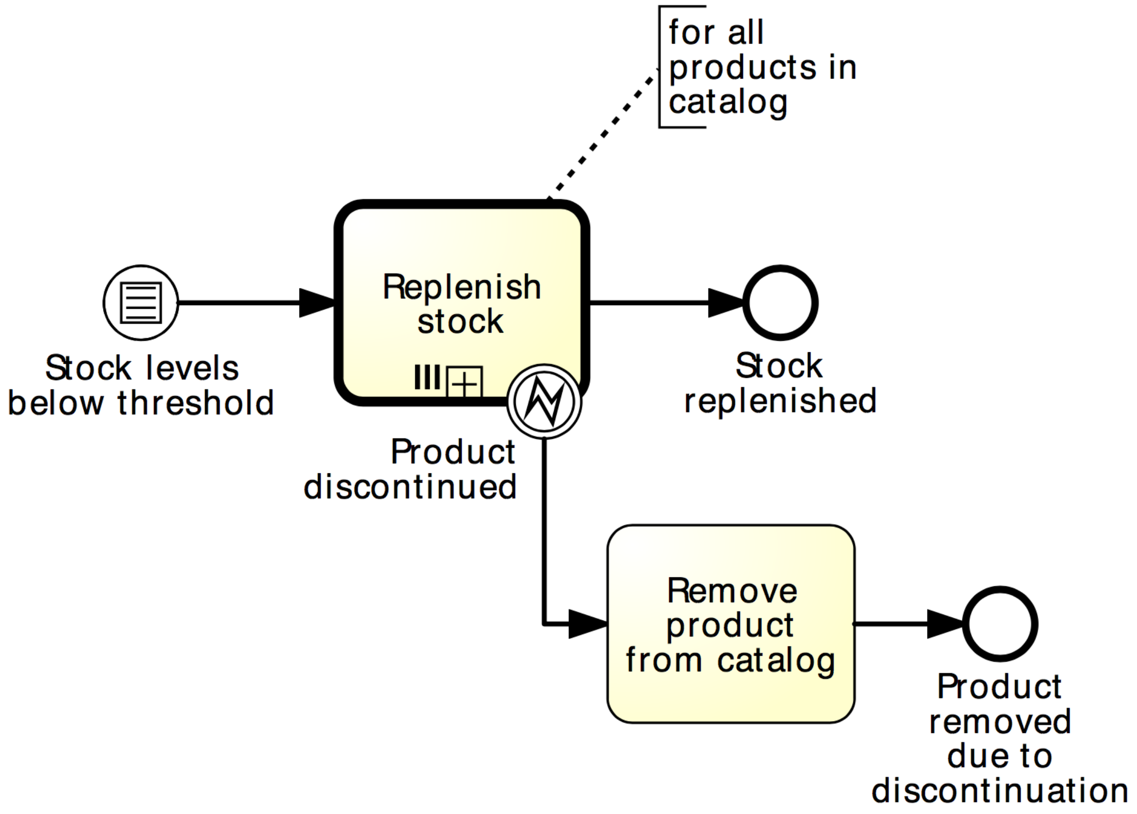
Processes and Business Rules

- Business rules implement organizationl 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


for all products in catalog
Replenish stock
Stock levels below threshold
Stock replenished
Product discontinued
Remove product from catalog
Product removed due to discontinuation

Exercise 4.14

Model the following business process snippet.

In a stock exchange, stock price variations are continuously monitored during the day. A day starts when the opening bell rings and concludes when the closing bell rings. Between the two bells, every time the stock price changes by more than 10%, the entity of the change is first determined. Next, if the change is high, a “high stock price” alert is sent, otherwise a “low stock price” alert is sent.

Recap

1. Structured loops can be modelled with loop activities, but arbitrary cycles cannot
2. Multi-instance activities model activities that need to be executed multiple times without a-priori knowledge of their number
3. Multi-instantiation extends to business objects and resources
4. Intermediate events can either be catching or throwing
5. Message events capture message exchange at the start, during and at the end of a process
6. Timer events capture temporal events (absolute or periodic)

Recap (cont‘ed)

7. Exceptions can be technology or business based
   – Also internal, external or activity timeouts
8. Exceptions best handled via process abortion using terminate events
9. Boundary error events capture internal extensions
10. Boundary message events capture external extensions
11. Boundary timer events capture activity timeouts
12. Signal events broadcast multiple messages
    – Can be used to capture complex exceptions
13. Compensation events required to revert effects of completed activities
14. Conditional events are one way of capturing business rules

And more...

- Condition events
- Escalation events
- Signal events, …
- Check the BPMN poster:
  - http://www.bpmb.de/images/BPMN2_0_Poster_EN.pdf

Summary

- In this lecture we have learned about:
  - BPMN sub-processes
  - Sub-process markers: loop and multiple-instance
  - Events: timer, message and error events
  - Event-based choice gateway
  - Boundary events (interrupting and non-interrupting)

And once I’ve got a model, what’s next?

Some process analysis techniques:

- Added-Value Analysis
- Root-Cause Analysis
- Flow Analysis
- Queuing Analysis
- Process Simulation