

Business Process Management

Paolo Bottoni

DIPARTIMENTO
DI INFORMATICA



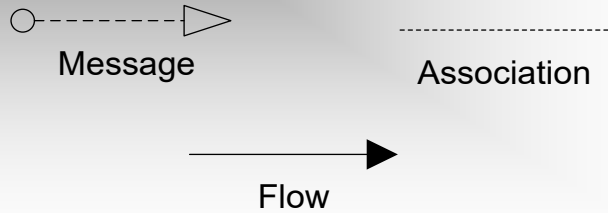
SAPIENZA
UNIVERSITÀ DI ROMA

Lecture 5: AdvancedBPM

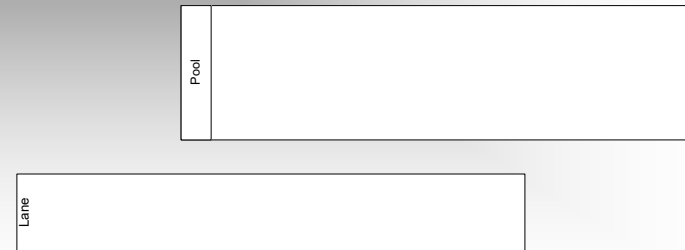
Adapted from the slides for the book : Dumas, La Rosa, Mendling & Reijers:
Fundamentals of Business Process Management, Springer 2013
<http://courses.cs.ut.ee/2013/bpm/uploads/Main/ITlecture3.ppt>

BPMN Main Elements - Recap

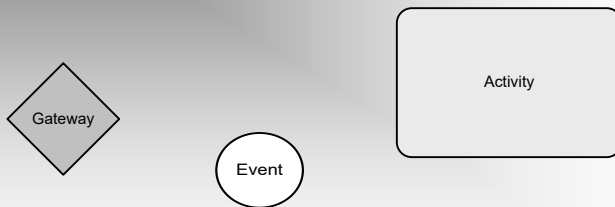
Connections



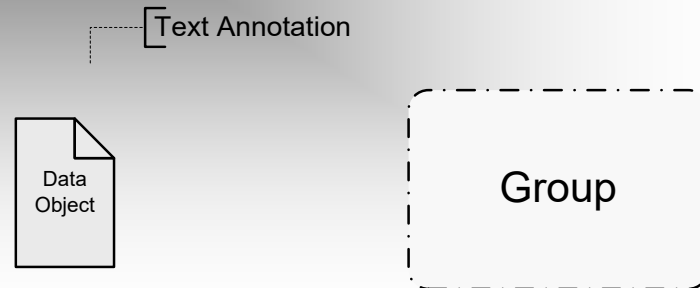
Swimlanes



Flow Objects



Artifacts



BPMN Gateways

Exclusive (XOR)

- Exclusive decision take one branch
- Exclusive merge Proceed when one branch has completed

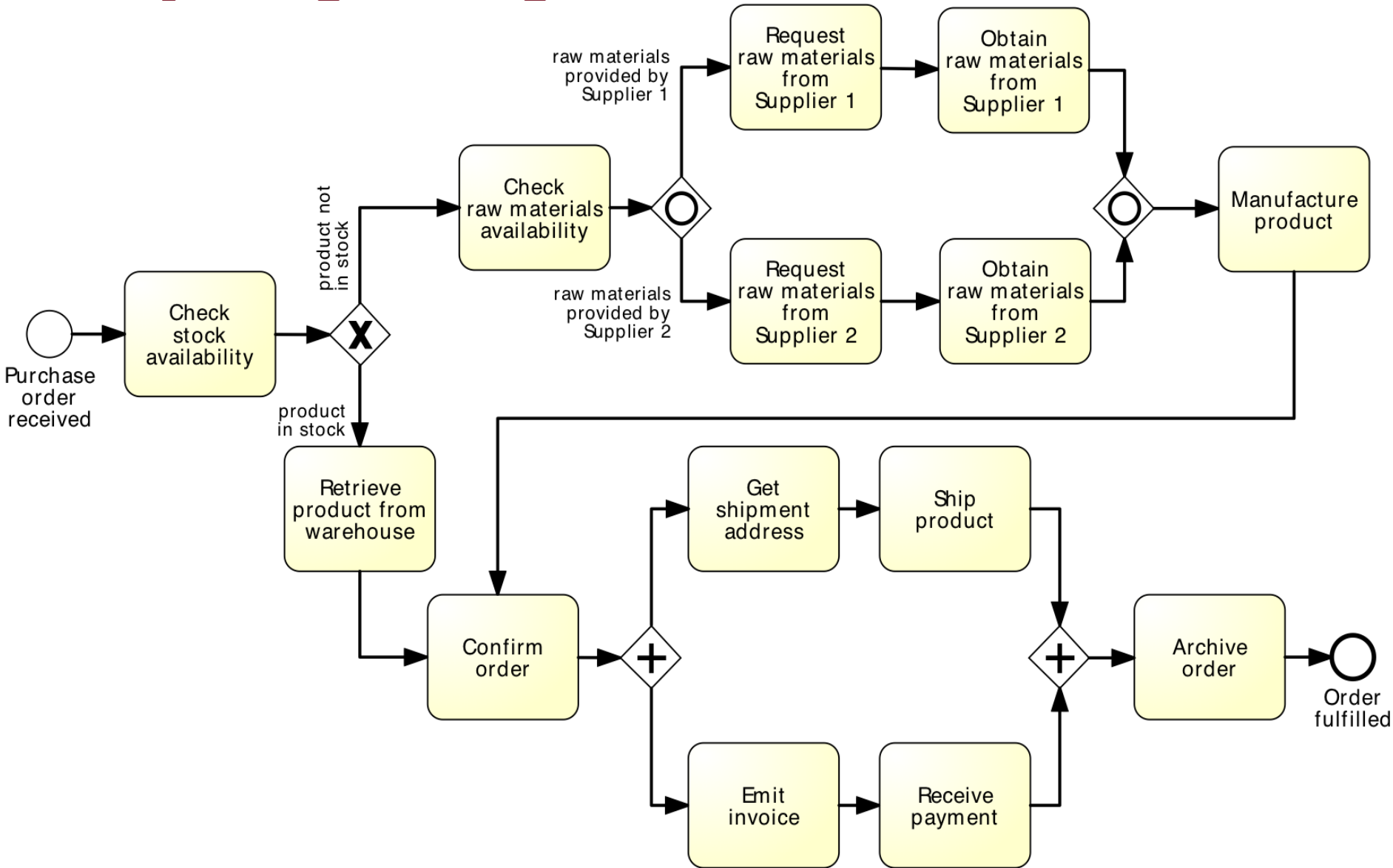
Parallel (AND)

- Parallel split take all branches
- Parallel join proceed when all incoming branches have completed

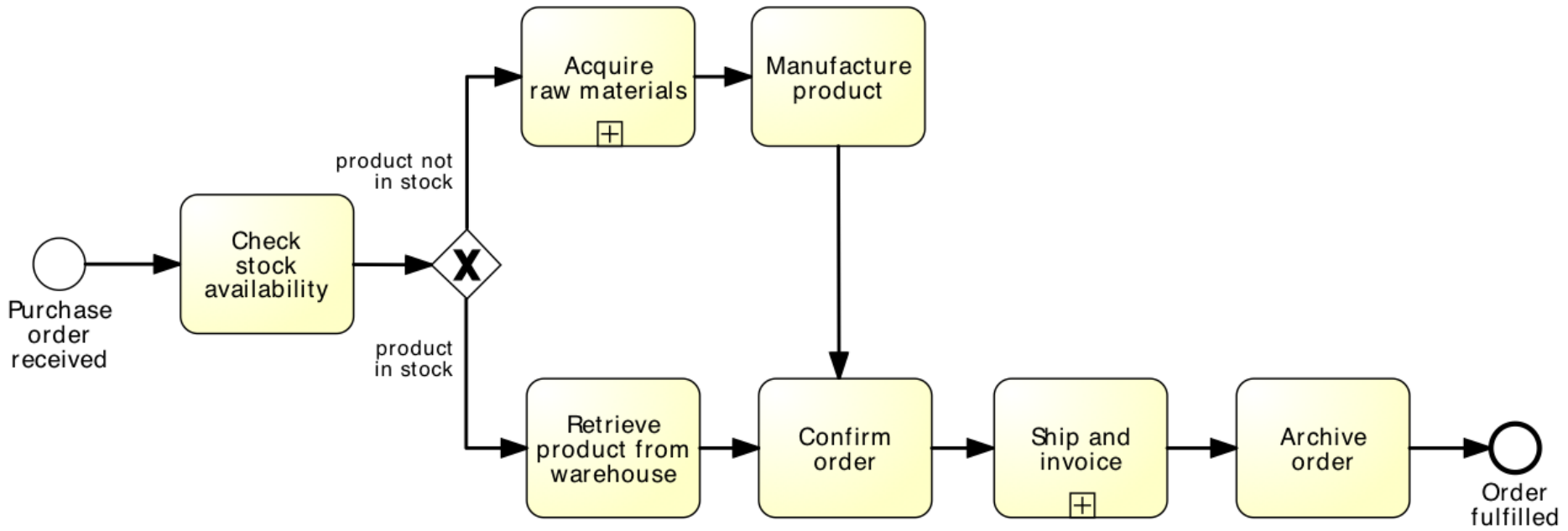
Inclusive (OR)

- Inclusive decision take one or several branches depending on conditions
- Inclusive merge proceed when all active incoming branches have completed

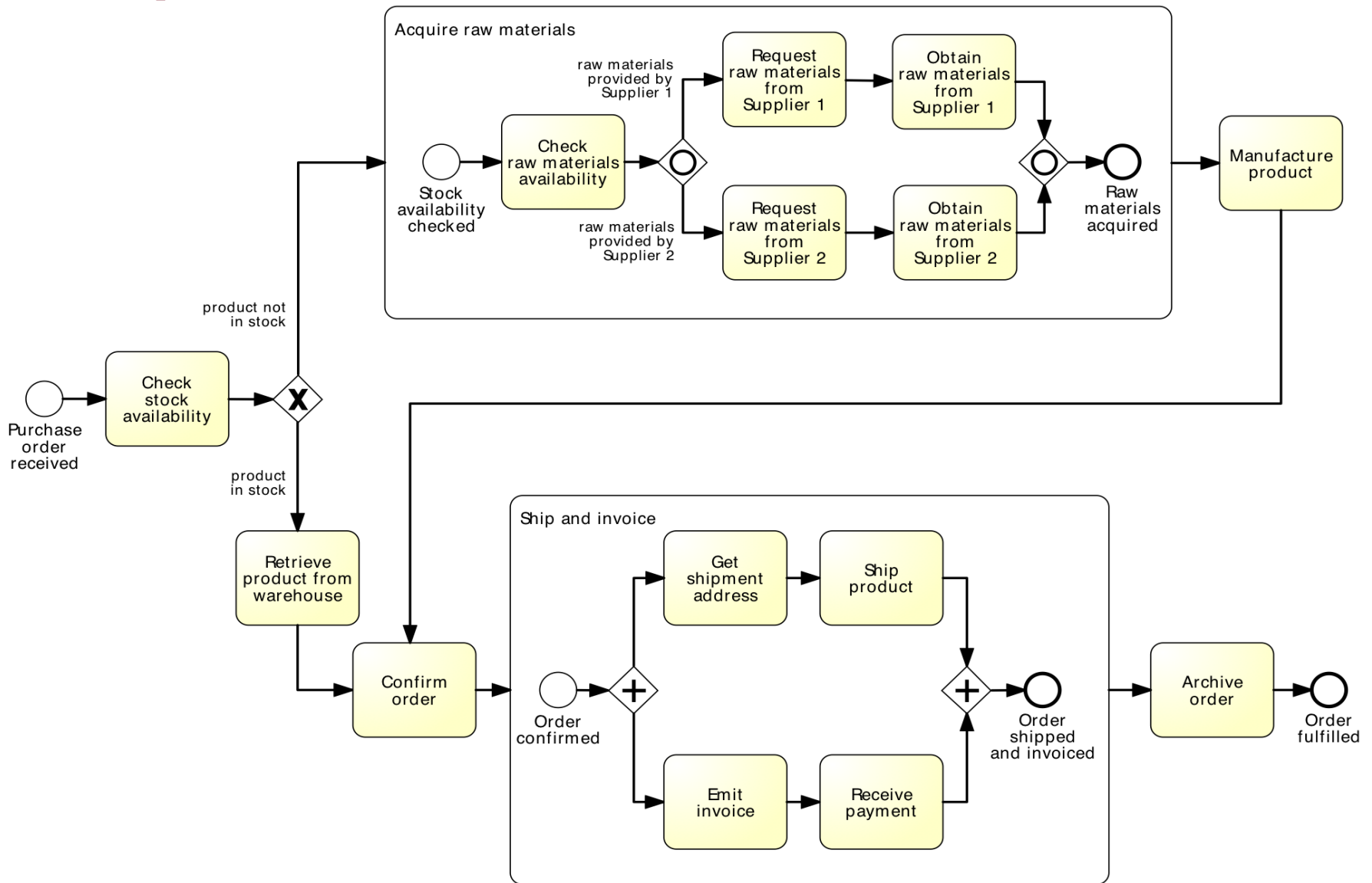
Anything wrong with this model?



Is this better?



Expanded...

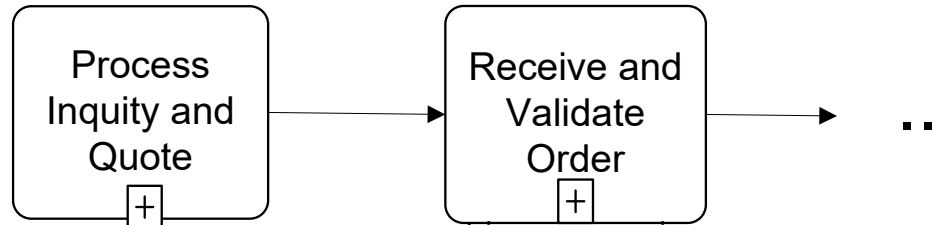


Sub-processes

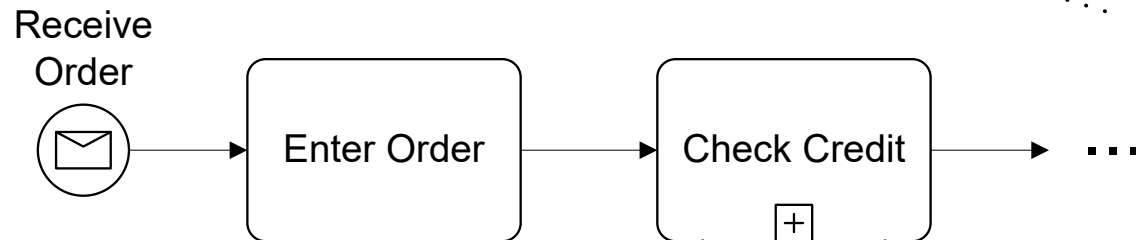
- An activity in a process can “invoke” a separate (sub-)process
- Use this feature to:
 1. Break down large models into smaller ones, making them easier to understand and maintain
 - process hierarchies
 2. Share common fragments across multiple processes
 - shared subprocesses
 3. Identify parts of a process that should be:
 - repeated
 - executed multiple times in parallel
 - cancelled

Process hierarchies

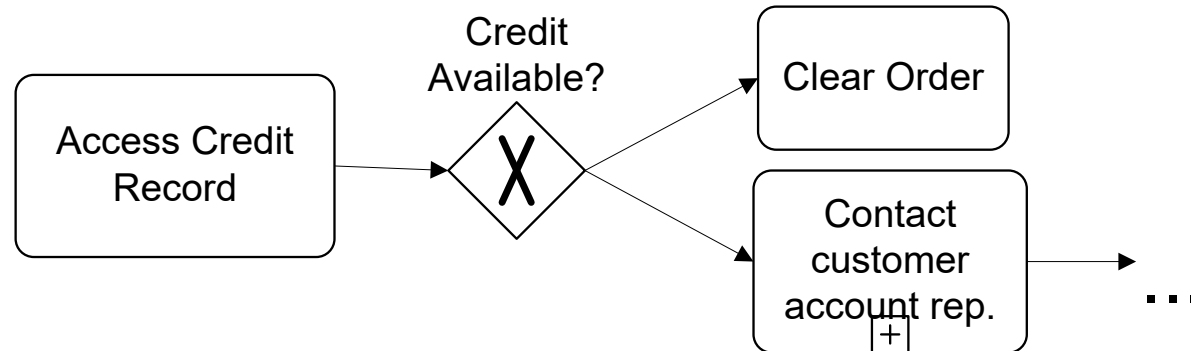
Level 3



Level 4



Level 5



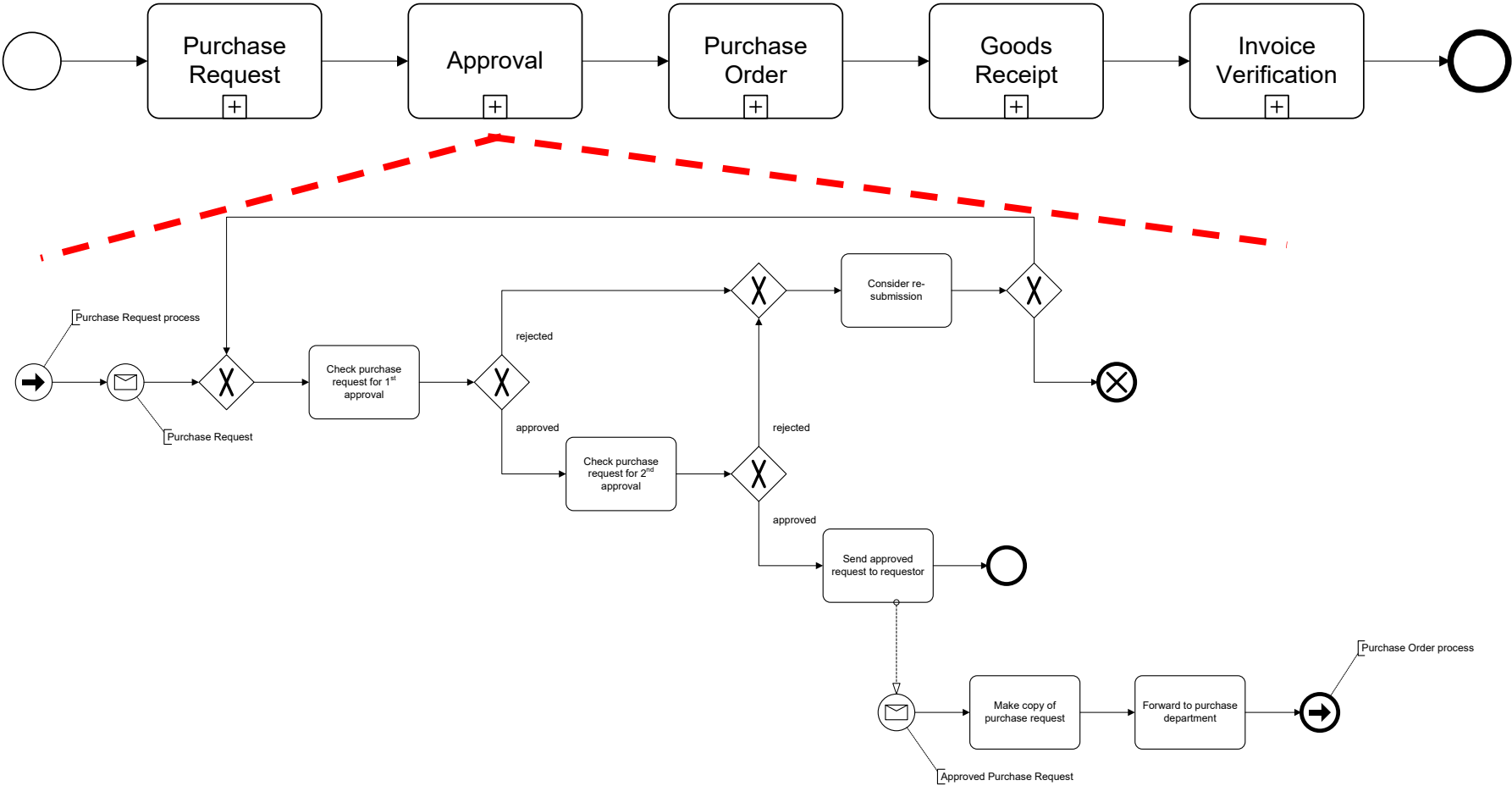
*Fragment of
the SCOR
model*

Modeling Guideline

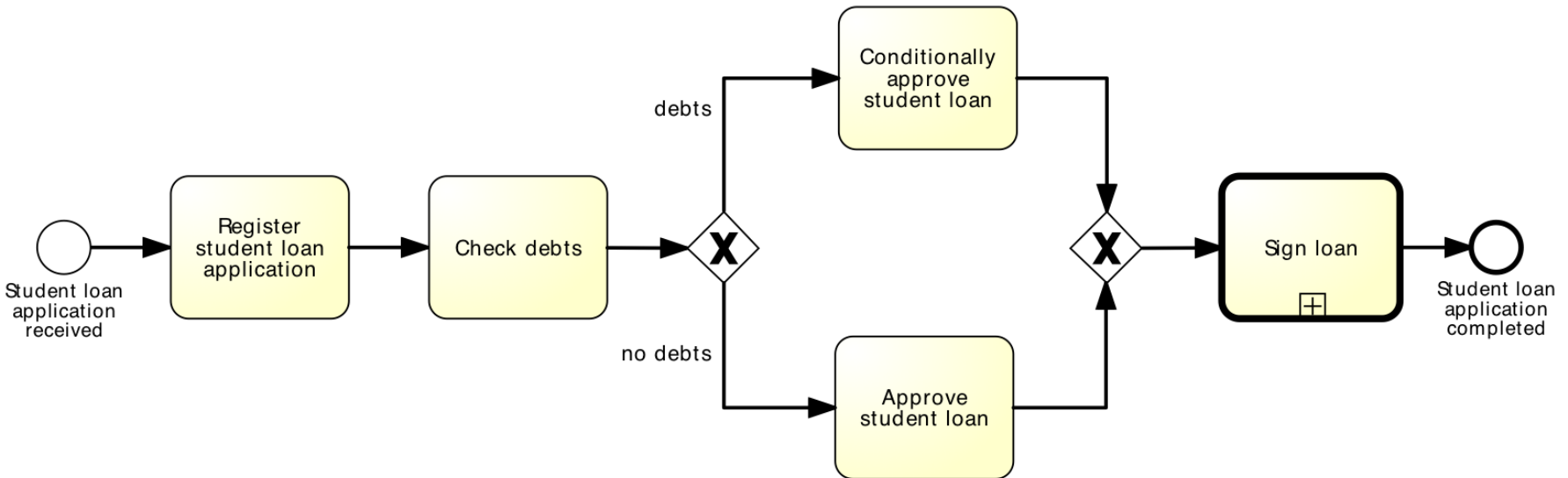
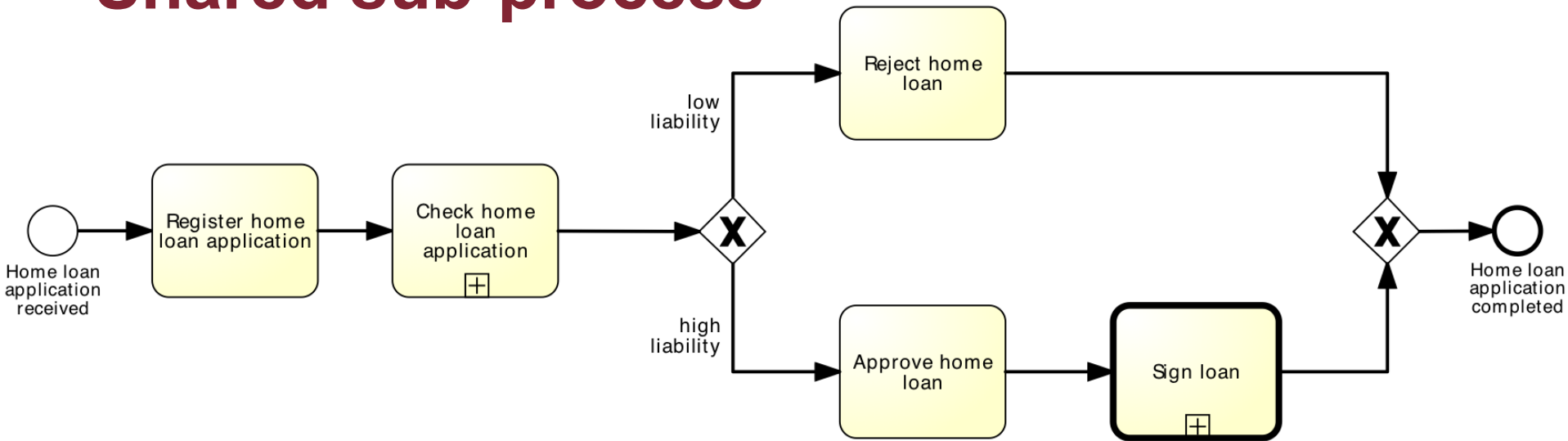
Start with a value chain

- 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”

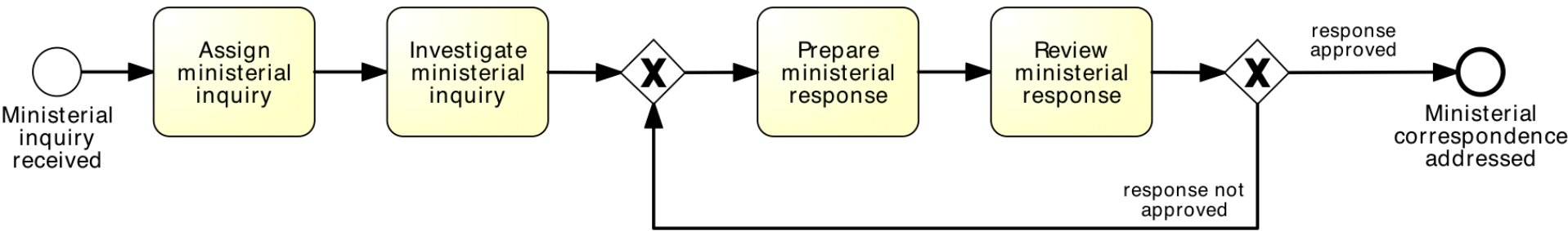
Showing the value chain with sub-processes



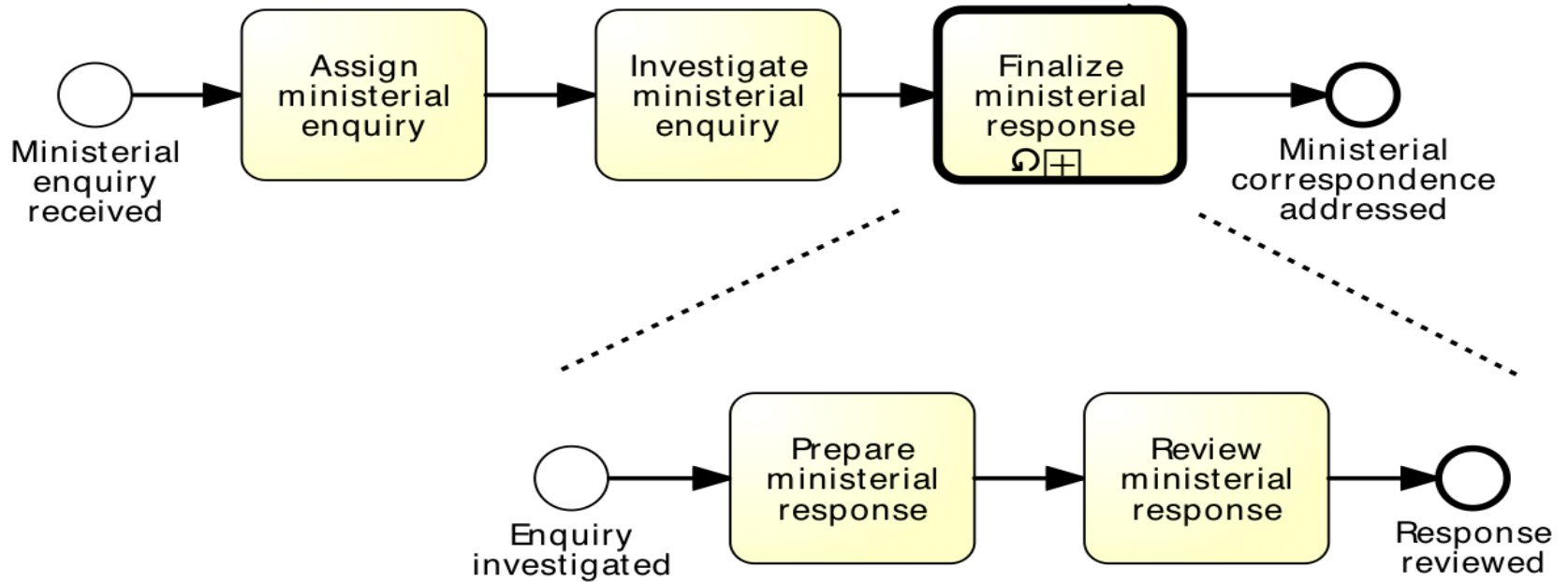
Shared sub-process



Sub-processes and loop marker



equivalent to:



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**.

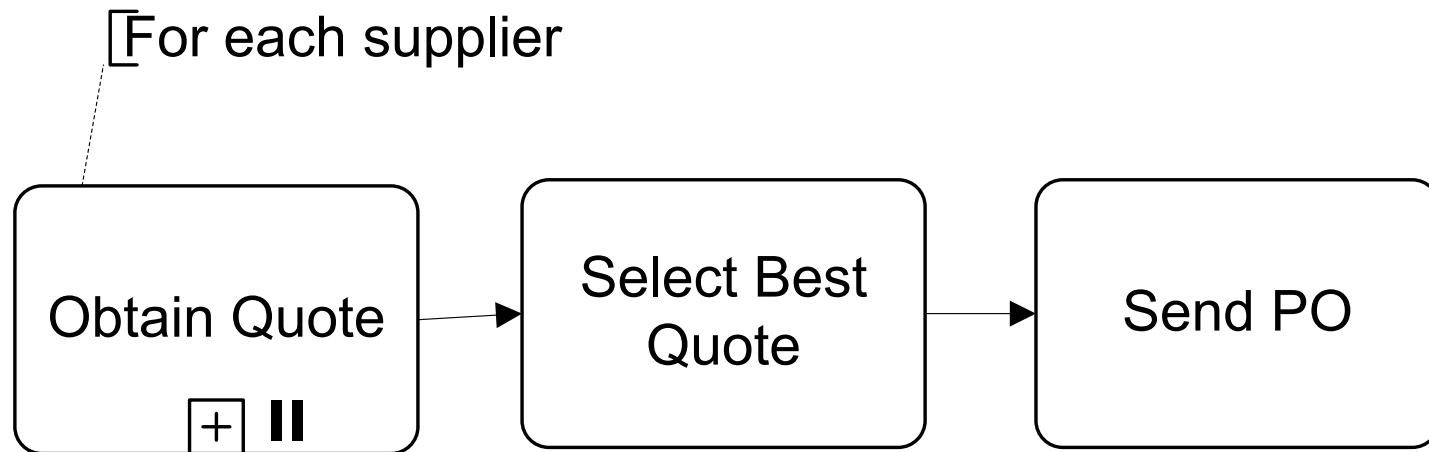
- Objects
- Tasks
- Paths
- Iterations
- Actors



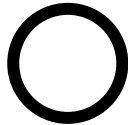
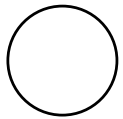
Multiple instance marker

- “Multiple instance” marker ~ “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

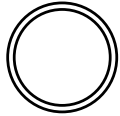
Multiple instance activity - example



Event types



Start/End Event – Indicates that an instance of the process is created/terminated when an event occurs without specifying the cause of this event



Intermediate event – Indicates that an event is expected to occur during the process, without indicating the cause of the event



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



End Message Event – Indicates that the process is terminated when a message is received



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

Event types (cont.)



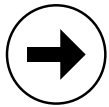
Start Timer Event – Indicates that an instance of the process is created at certain date(s)/time(s), e.g. start process at 6pm every Friday



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



End Link Event – Indicates that the process flow continues elsewhere (e.g. in a separate diagram)



Start/Intermediate Link Event – Indicates that the process flow is being picked up from a “previous” diagram”.



Intermediate/End Exception Event – Indicates an error: the “end” version generates the exception event while the “intermediate” version consumes it

And more...

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

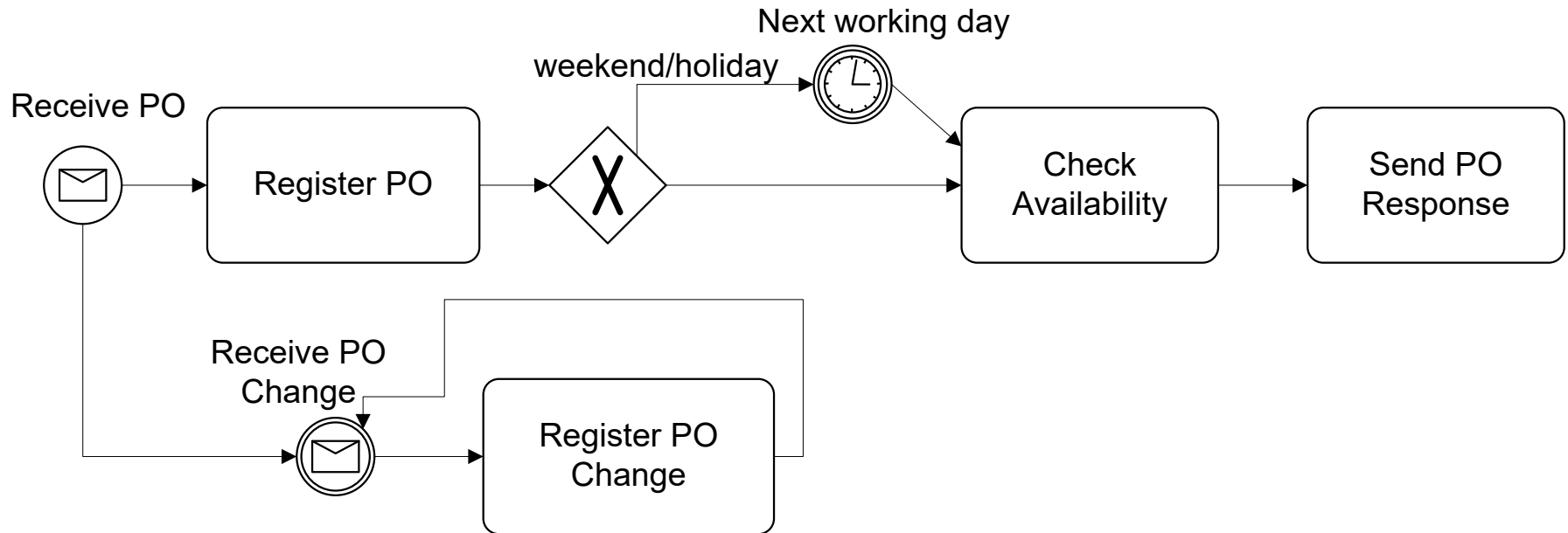
Modelling with events - Example

A 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.

Anytime during the process, the customer may send a “PO change request”. When such a request is received, it is just registered, without further action.

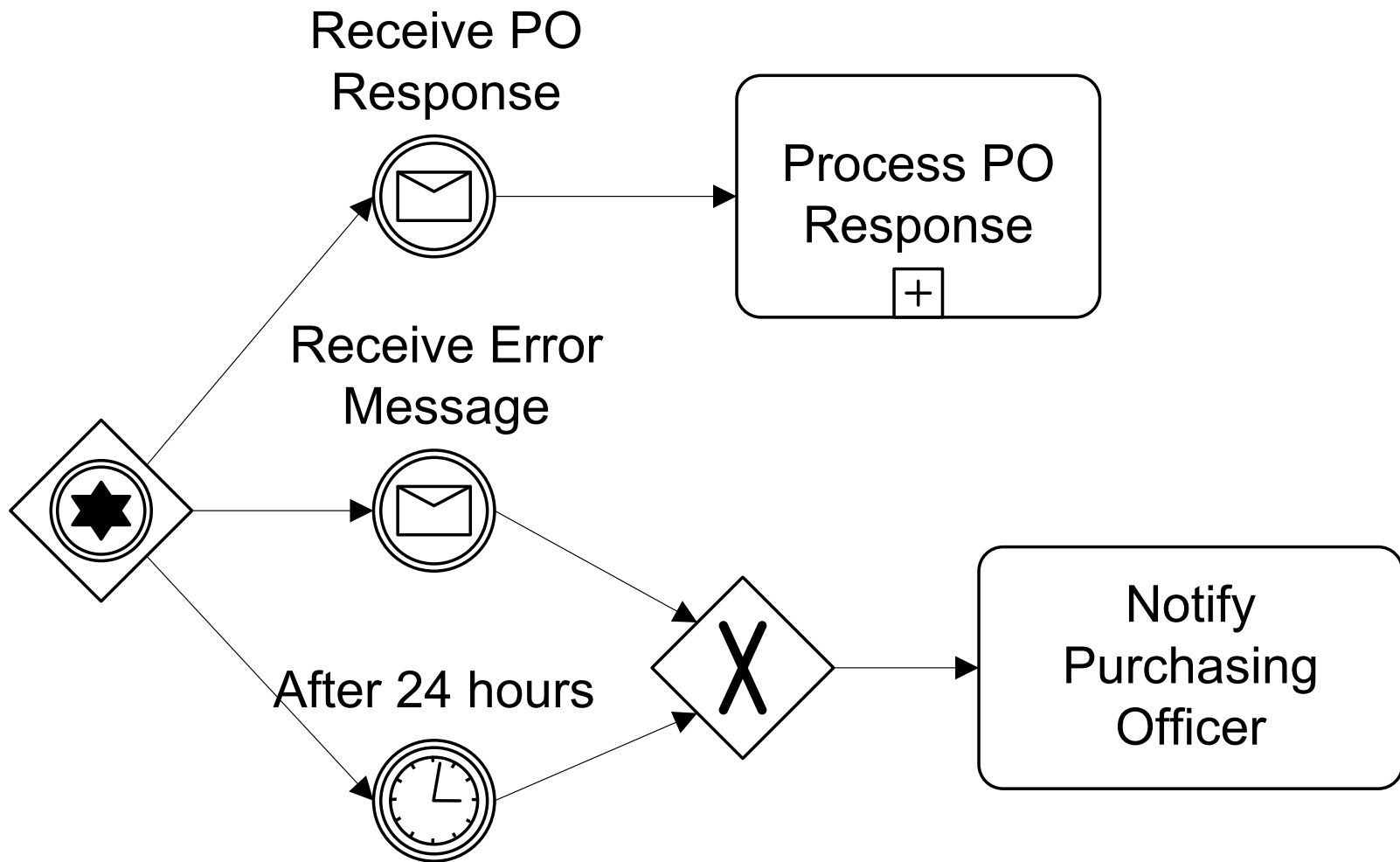
Modelling with events - Example



Data-based vs. event-based decision

- In an XOR-split gateway, one branch is chosen based on expressions evaluated over available data
 - Choice is made immediately when the gateway is reached
- Sometimes, the choice must be delayed until something happens
 - Choice is based on a “race between events”
- BPMN distinguishes between:
 - Exclusive decision gateway (XOR-split)
 - Event-based decision gateway

Event-driven Decision – Example



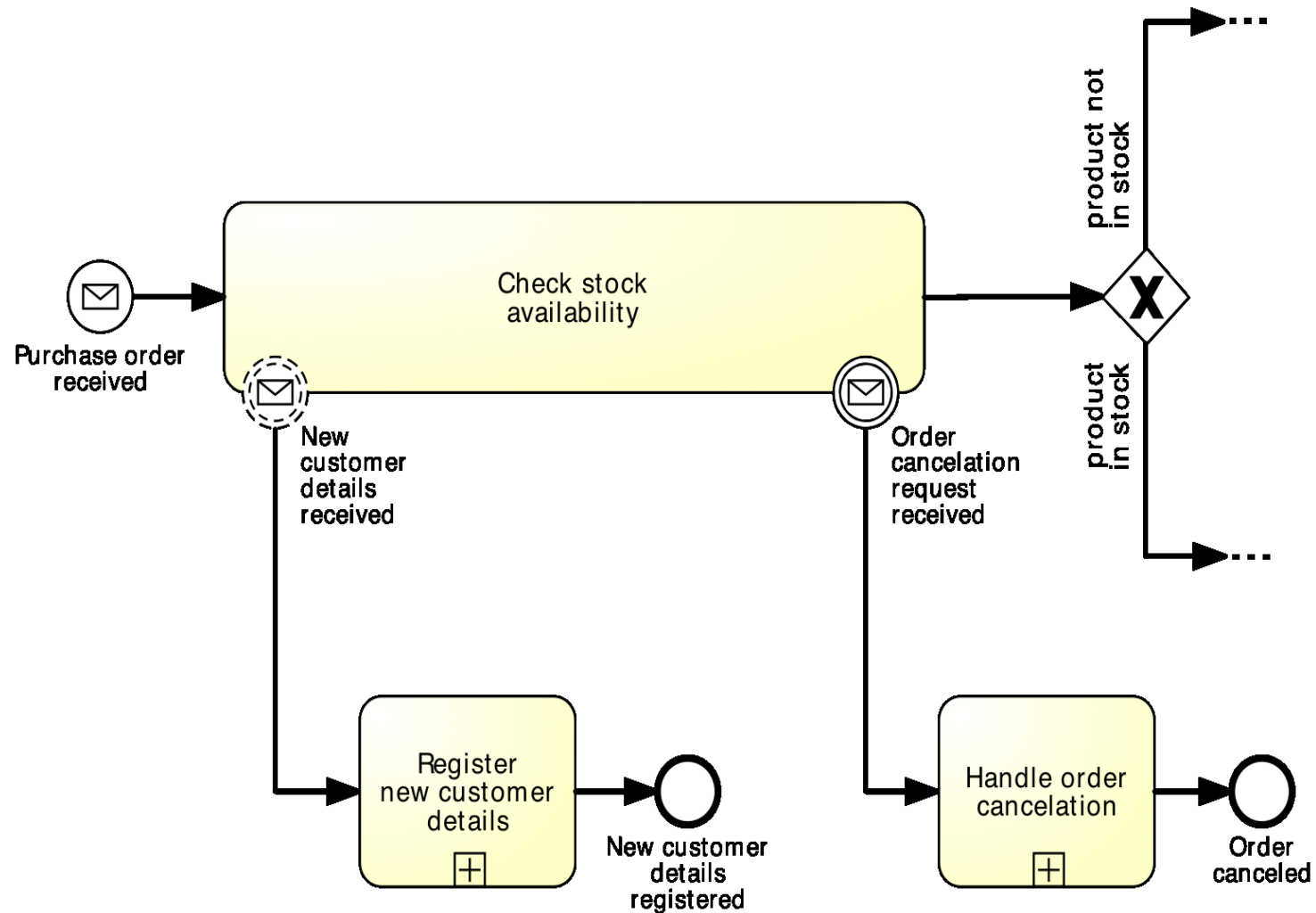
Exercise

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

Boundary events

- Sometimes during a sub-process execution, some event may occur that needs some action...
- Such events are placed at the boundaries of the sub-process (boundary events)
- Two flavors:
 - Interrupting boundary events
 - Non-interrupting boundary events

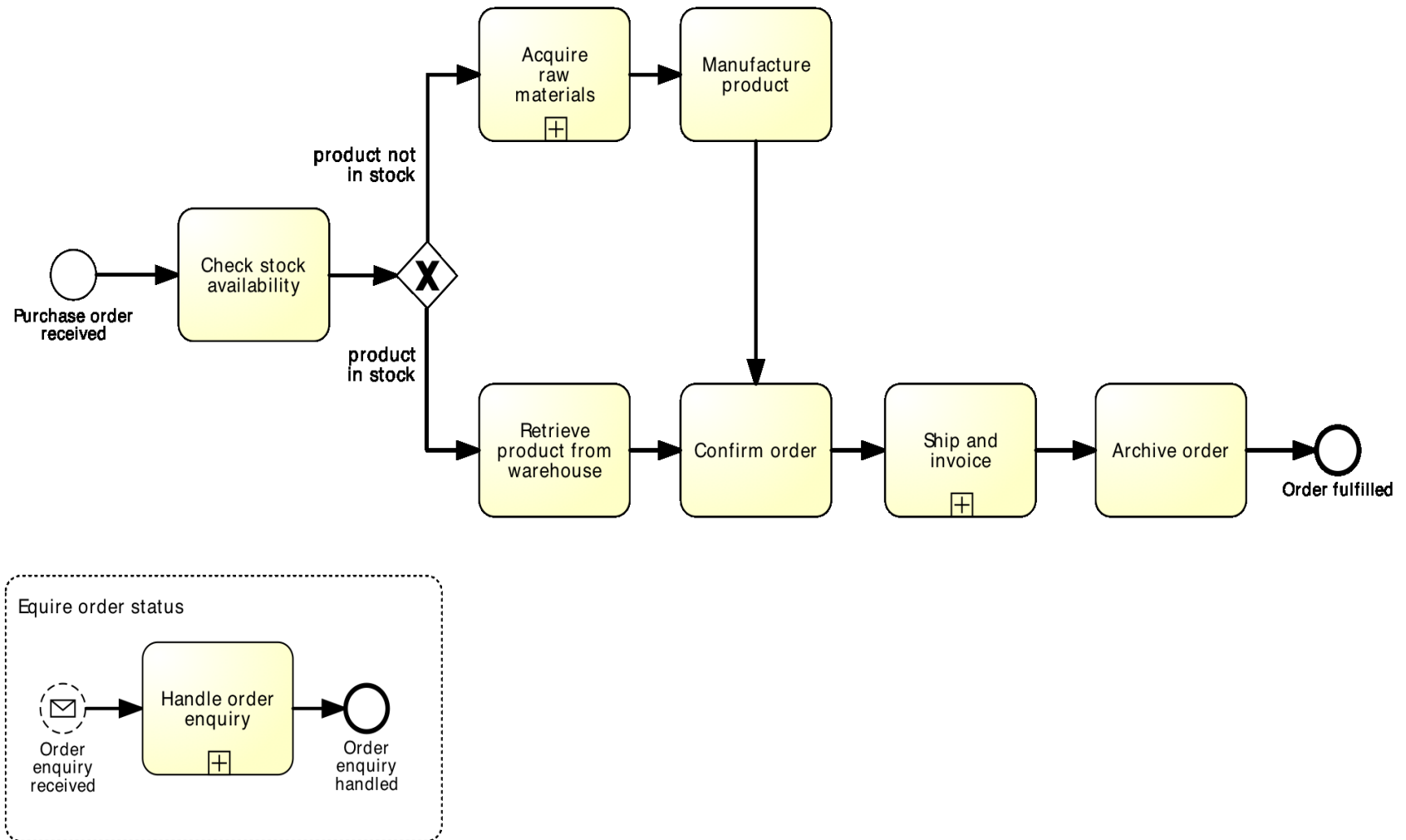
Boundary Events – Example



Event sub-processes

- Event sub-process are processes attached to a parent process, that are triggered when an event happens
- Alternative to putting a boundary non-interrupting event around the parent process

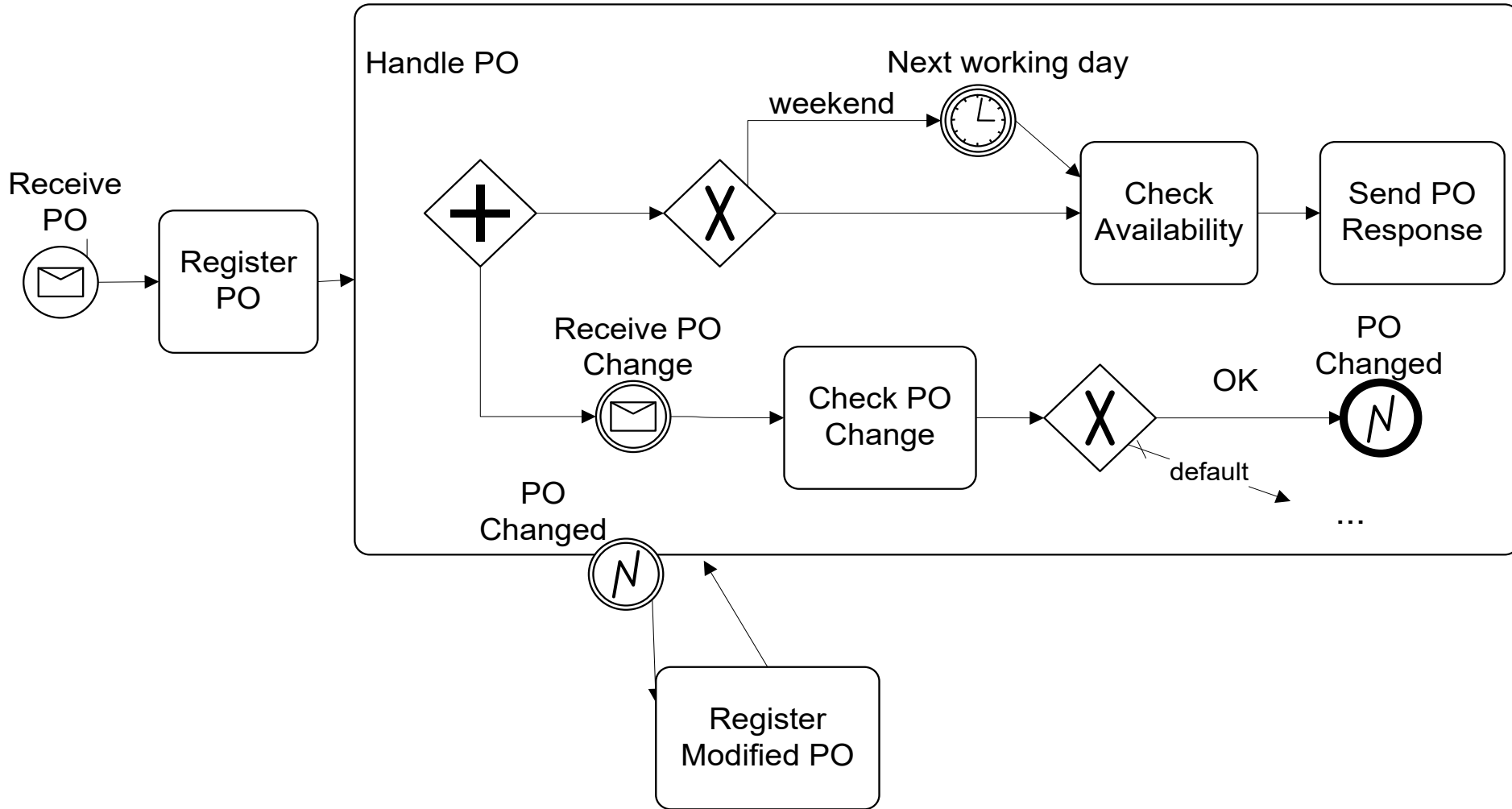
Event sub-processes – Example



Exception handling (error events)

- Exceptions are events that deviate a process from its “normal” course
- Handling exceptions often involves stopping a subprocess and performing a special activity
- Achieved using two event nodes:
 - An “end error event” that stops the enclosing subprocess execution
 - An “intermediate error event” attached to the enclosing subprocess – this is where the process execution will continue after the error

Error events – Example



Exercise

When a claim is received, it is registered. After registration, the claim is classified leading to two possible outcomes: simple or complex. If the claim is simple, the policy is checked. For complex claims, both the policy and the damage are checked independently.

A possible outcome of the policy check 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.

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