

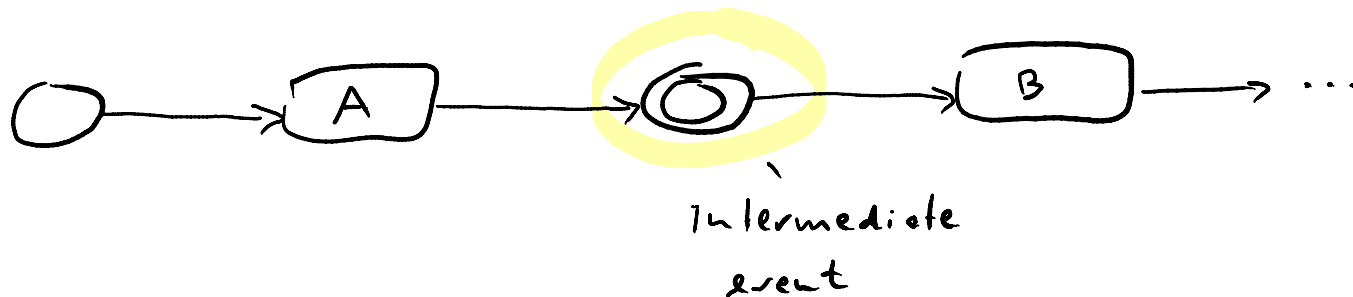
Video Clip 2.6

Intermediate Events

BPMN Meets DMN: Business Process and Decision Modeling:
An openHPI Course by Mathias Weske

Intermediate Events

- Intermediate events happen after the start of a business process and before its termination
- Intermediate events can be catching or throwing
 - Catching: the process waits for the occurrence of the event
 - Throwing: the process emits the event and continues with the outgoing sequence flow
- An intermediate event can be connected to other nodes of a process model by incoming and outgoing sequence flow arcs



Application Scenario (cont.)

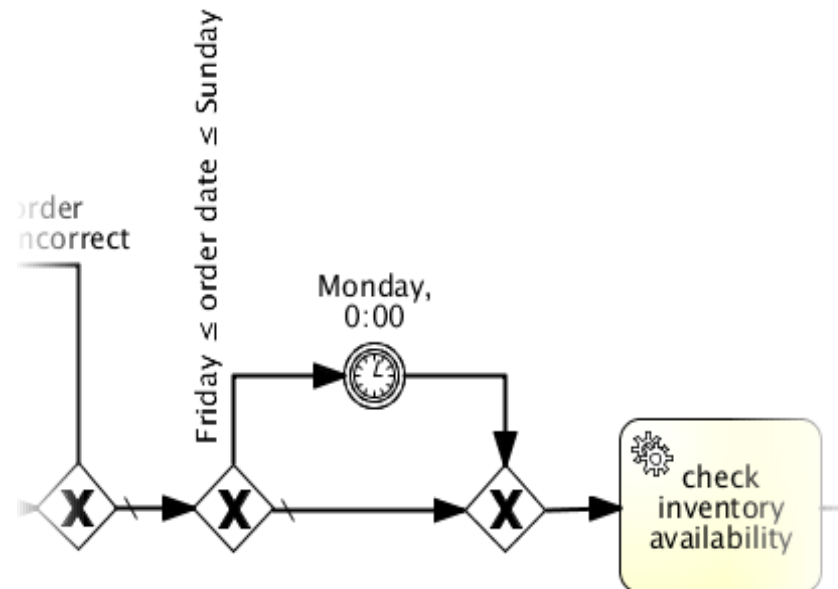
...

If the order is placed on a Friday or on a weekend, the inventory check shall be delayed until Monday, as the inventory is updated every Sunday evening.

...

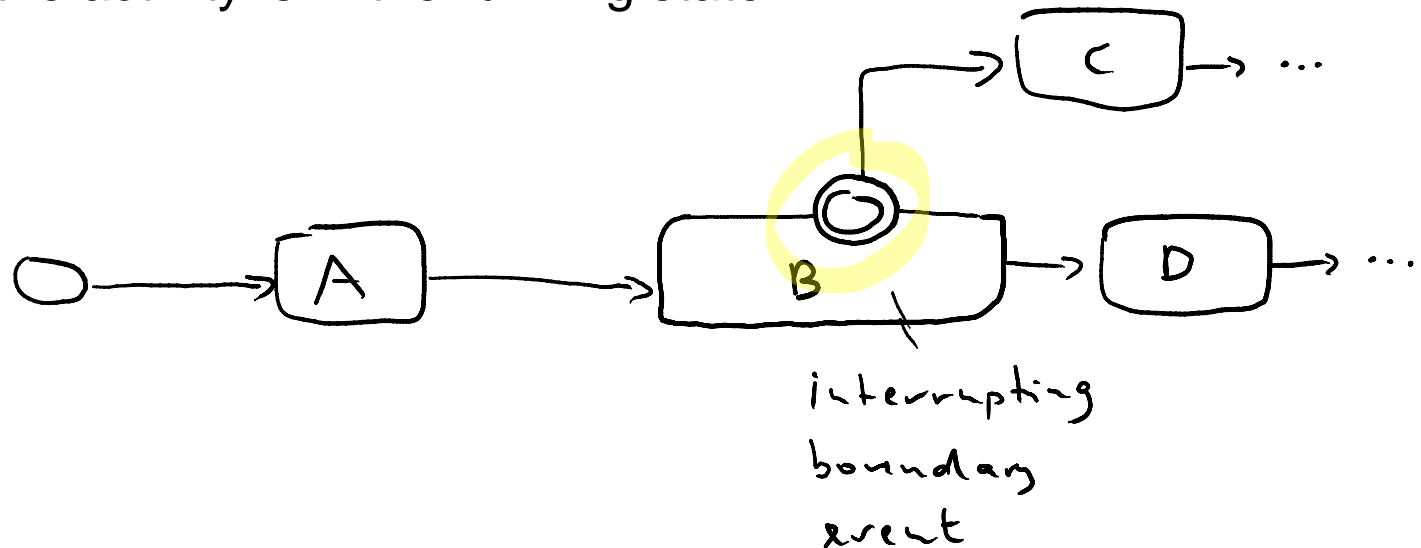
Intermediate Timer Event

- Intermediate timer events can only be catching
- They act as a delay mechanism
 - By a specific date / time, e.g., “December 16, 2013”
 - By a recurring date / time, e.g., “every Saturday at 10:00 am”
 - By a duration, e.g., “one week”



Boundary Event

- A boundary event is on the boundary of an activity
 - Boundary events are always catching events
 - The occurrence of a boundary event is only relevant, if it occurs while the activity is in the running state



- Interrupting boundary events: Interrupt the activity
- Non-interrupting boundary events: Do not interrupt it



Application Scenario (cont.)

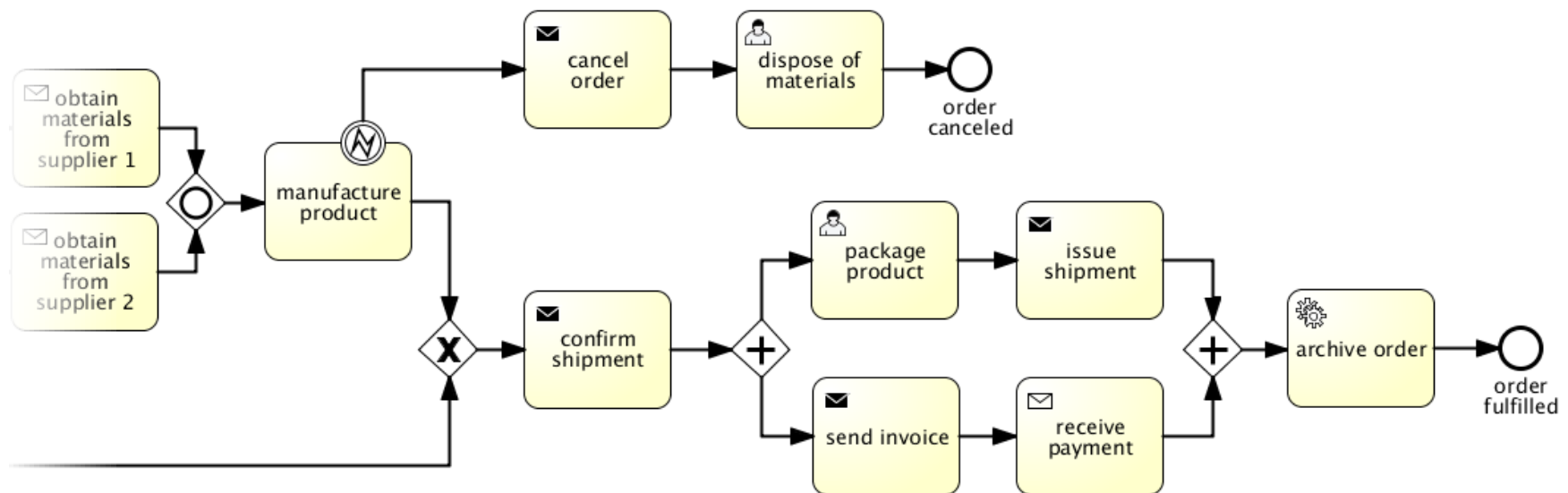
...

If the product is not in stock and an error occurred during manufacturing the product, the order shall be canceled and the obtained materials be disposed of. In that case, the process terminates unsuccessfully.

...

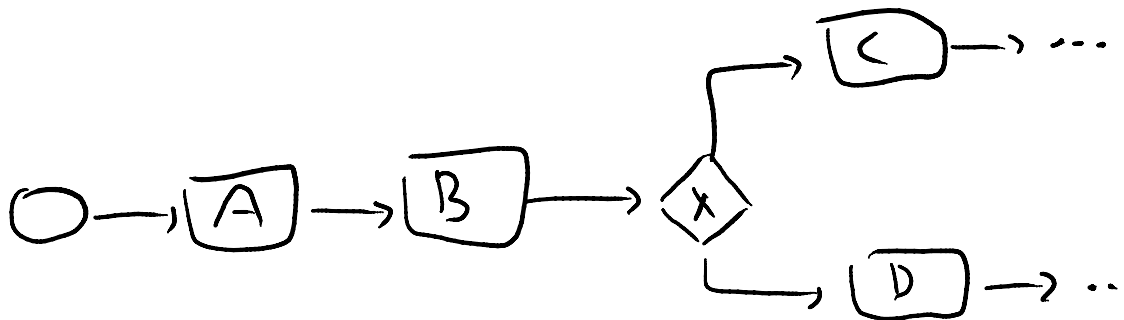
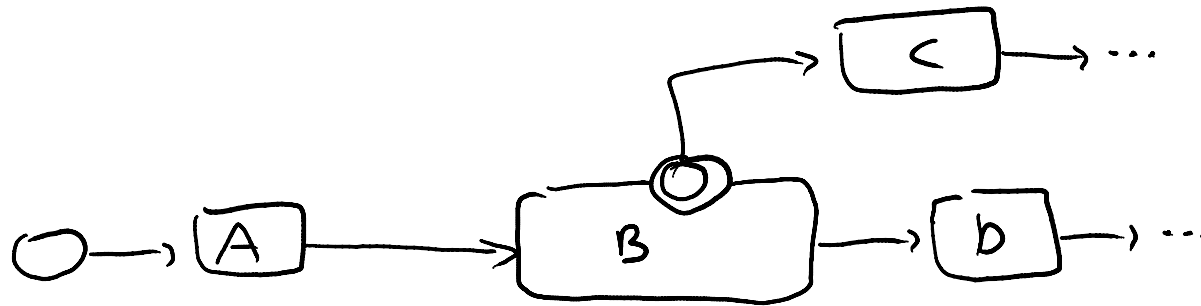
Interrupting Boundary Error Event

- When the boundary event occurs, *manufacture product* is interrupted and the flow continues with cancelling the order
 - Typically, activities to handle the error situation are started after a boundary error event occurred



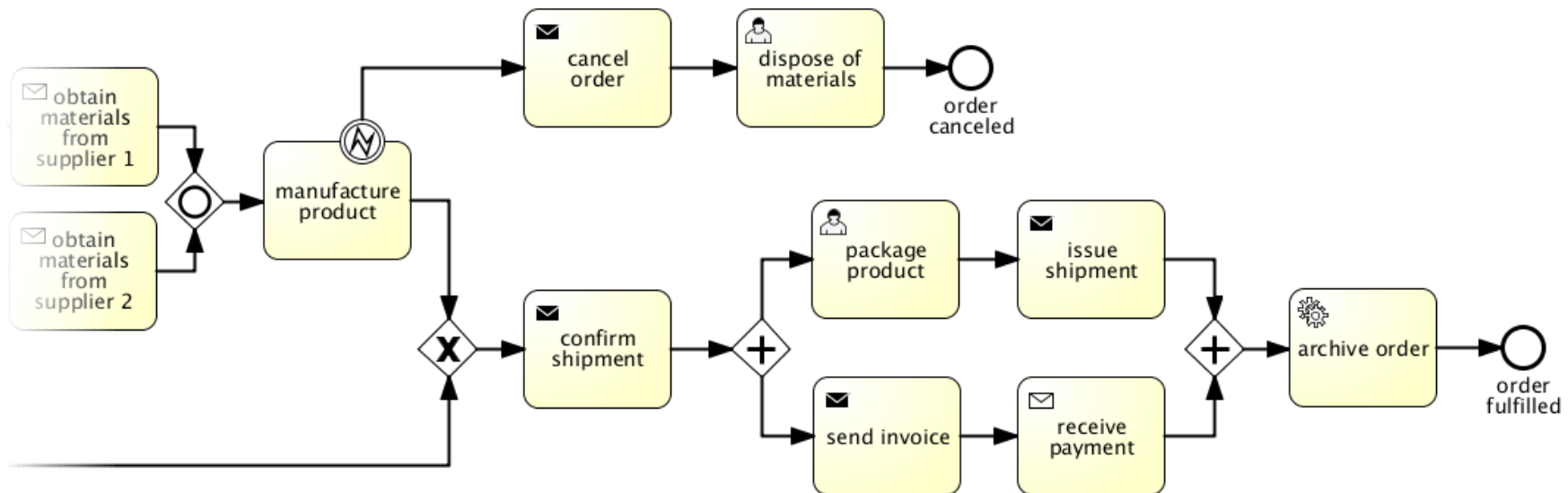
Interrupting Boundary Event and XOR Split

- What is the difference to an XOR split decision?



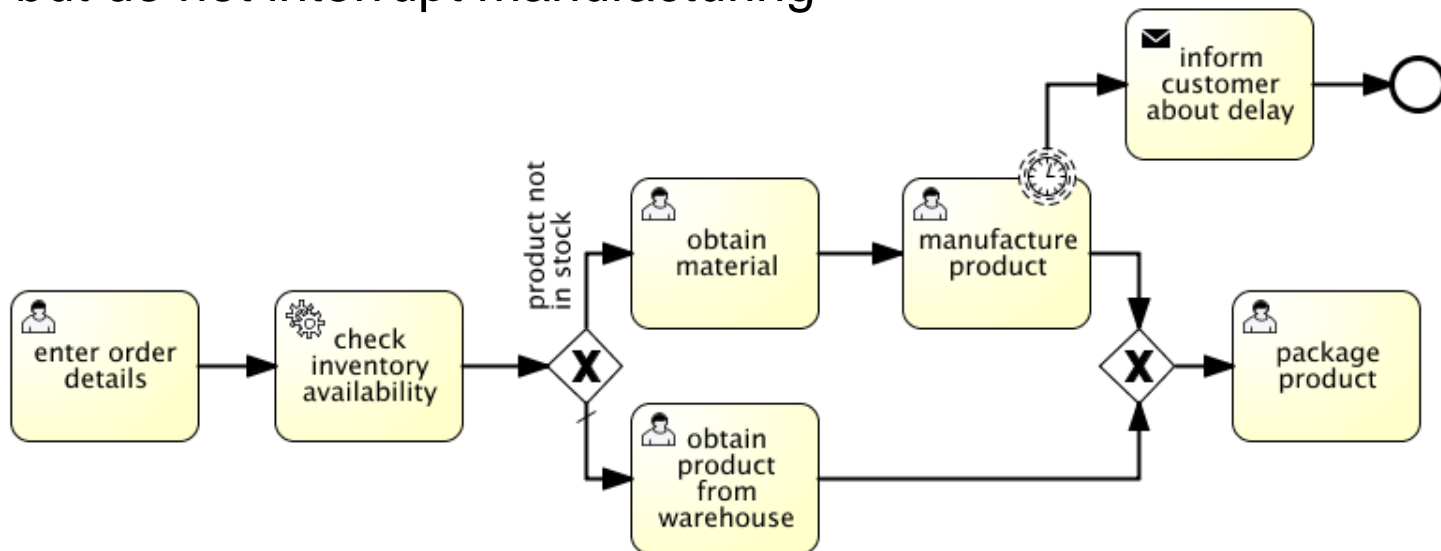
Interrupting Boundary Event and XOR Split

- There are subtle differences
 - Decision by event occurrence / by conditions
 - There is a difference in the execution semantics, since in the XOR case, the activity completes, while it does not in the case of the boundary event
 - From the modeling perspective, it is clearer that an error has occurred rather than a normal decision during a process



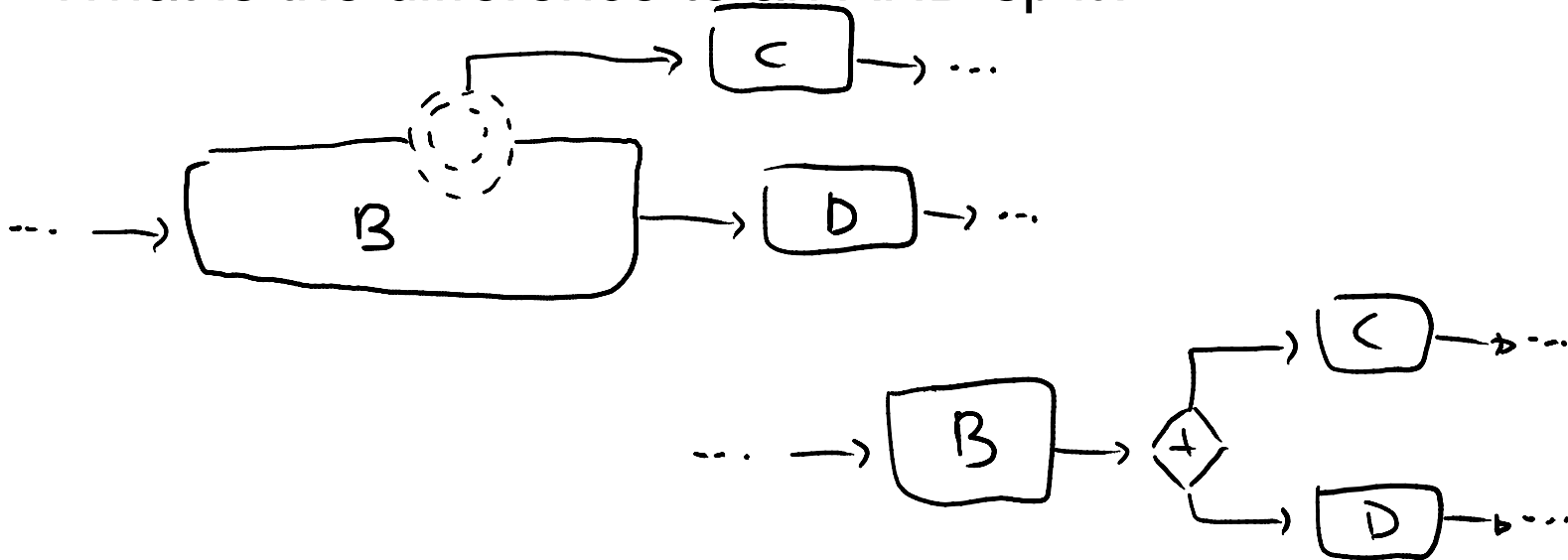
Non-Interrupting Boundary Event

- Boundary events might also be non-interrupting
 - In this case, the activity continues normally, while spawning a reaction to the event
 - Notation: circle with a dashed double border
- Example
 - Inform the customer if manufacturing takes longer than expected, but do not interrupt manufacturing



Non-Interrupting Event and AND Split

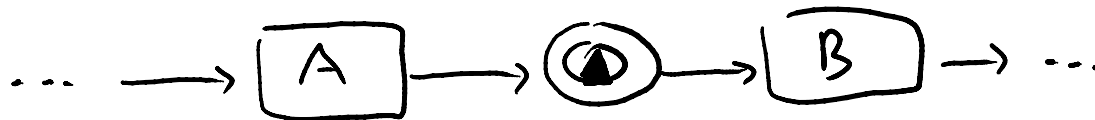
- What is the difference to an AND split?



- There are subtle differences as well
 - There is a difference in the execution semantics, since in the AND case, activities are sequential, while this is not the case with the boundary event
 - We need to be careful with joining the flows in an AND gateway, since the event can occur several times

Intermediate Signal Events

- Scope of events
 - The scope of throwing and catching events is a single process instance and its environment (timer, incoming messages, etc.)
- Signal events lift this assumption
 - Signal events can be caught by any other business process

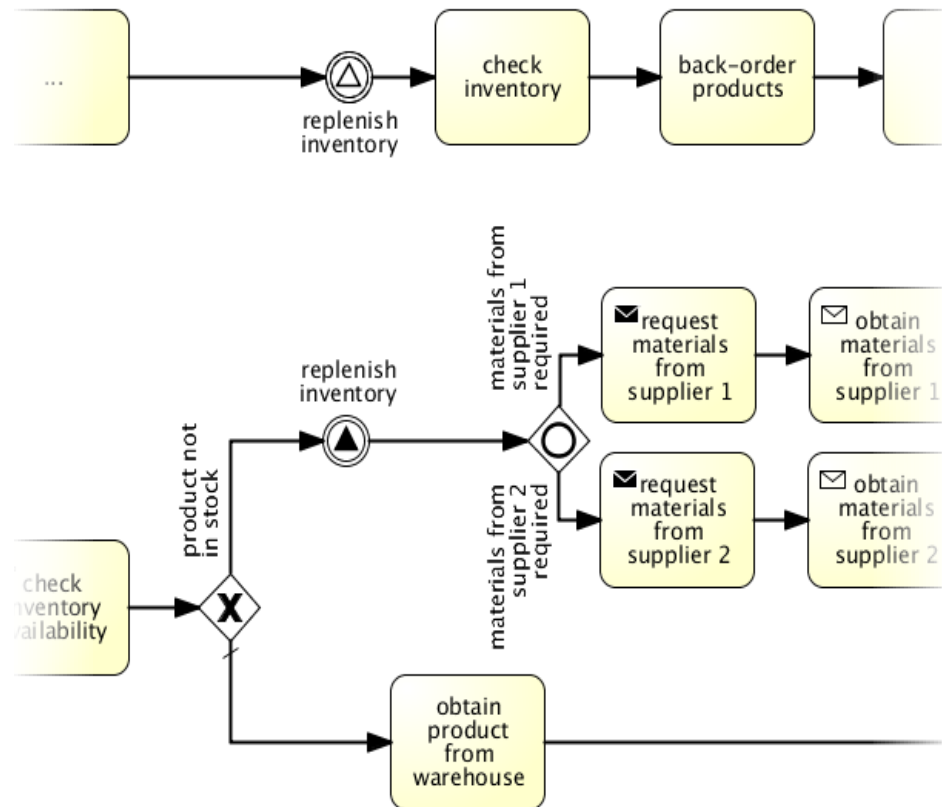


- A BPMN Signal is similar to a signal flare that shot into the sky is visible to everyone who might be interested (from the BPMN 2.0 Specification)

Intermediate Signal Events
































































- Example

- A signal event may be used to inform another process to replenish the inventory, if a product is not available



Overview of Events

S

	Start			Intermediate			End	
	Standard	Event Sub-Process Interrupting	Event Sub-Process Non-Interrupting	Catching	Boundary Interrupting	Boundary Non-Interrupting	Throwing	Standard
None: Untyped events, indicate start point, state changes or final states.								
Message: Receiving and sending messages.								
Timer: Cyclic timer events, points in time, time spans or timeouts.								
Escalation: Escalating to an higher level of responsibility.								
Conditional: Reacting to changed business conditions or integrating business rules.								
Link: Off-page connectors. Two corresponding link events equal a sequence flow.								
Error: Catching or throwing named errors.								
Cancel: Reacting to cancelled transactions or triggering cancellation.								
Compensation: Handling or triggering compensation.								
Signal: Signalling across different processes. A signal thrown can be caught multiple times.								
Multiple: Catching one out of a set of events. Throwing all events defined								
Parallel Multiple: Catching all out of a set of parallel events.								
Terminate: Triggering the immediate termination of a process.								

Source: BPMN Poster
<http://www.bpmb.de/index.php/BPMNPoster>

Video Clip 6

Intermediate Events

- If catching, can act as delay mechanism
- Boundary events restrict the scope to an activity
- Can be interrupting or non-interrupting
- Signal events

Summary of Week 2

- Activities
 - Represent units of work, take time
 - Activity instances follow a state transition diagram
 - Task types determine the nature of a task
- Gateways
 - XOR – for exclusive choices (1/m) and loops
 - AND – for interleaved ordering or concurrency (m/m)
 - OR – for inclusive choices (n/m)
 - Be careful with uncontrolled flow
- Events
 - Start events, end events
 - Intermediate events might be throwing, catching
 - Boundary events might be interrupting, non-interrupting

