

BPMN Meets DMN: Business Process and Decision Modeling

*Week 4:
Advanced Business Process Modeling*

An openHPI Course

Prof. Dr. Mathias Weske

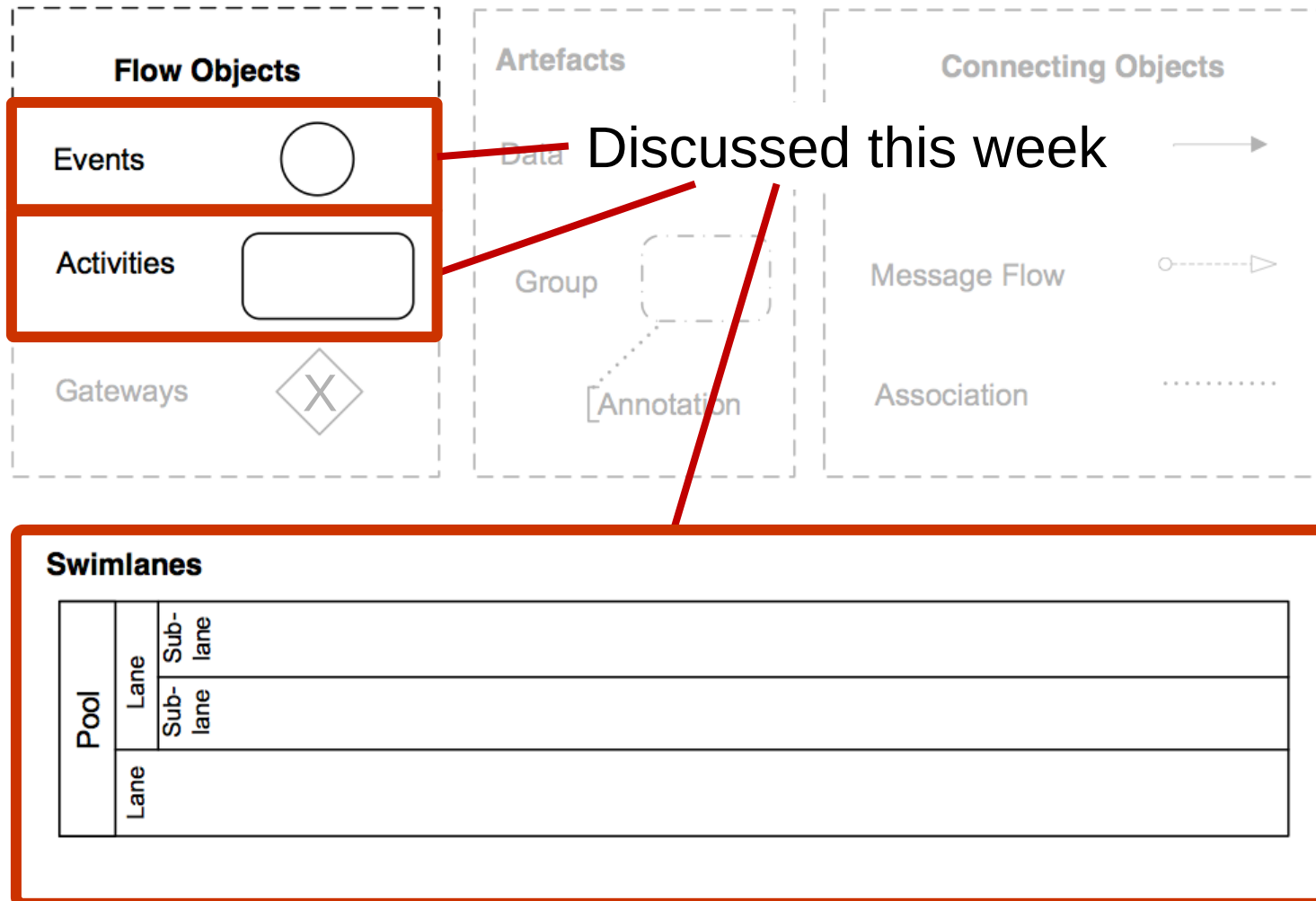
Hasso Plattner Institute at the University of Potsdam
Business Process Technology

Week 4

Advanced Business Process Modeling

- 4.1 Sub-Processes
- 4.2 Sub-Processes and Boundary Events
- 4.3 Activity Modifiers
- 4.4 Racing Events
- 4.5 Modeling Organization
- 4.6 Resource Allocation Patterns

Notation Categories and their Elements



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 4.76. BPMN: categories of elements

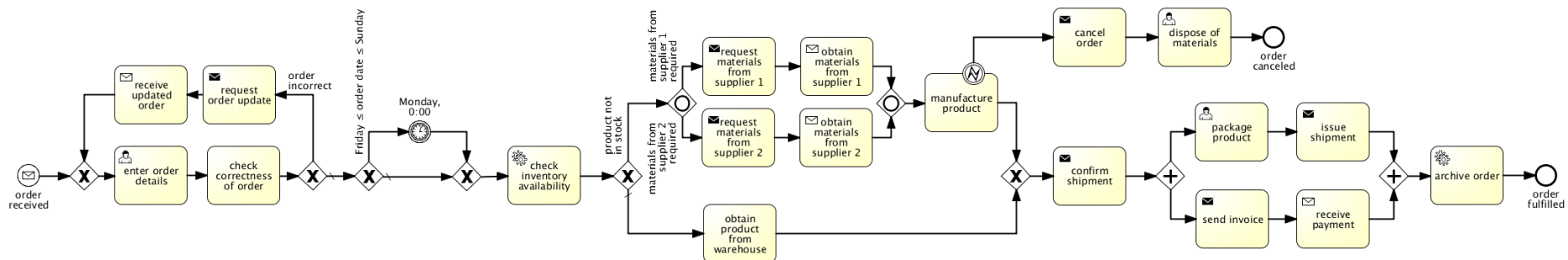
Video Clip 4.1

Sub-Processes

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

Recap of Week 2

- Already covered
 - Atomic activities (tasks) and sequence flow
 - Gateways (XOR, AND, OR)
 - Uncontrolled flow
 - Events (start, intermediate, end, throwing, catching, boundary)
- Application scenario
 - Order fulfillment process

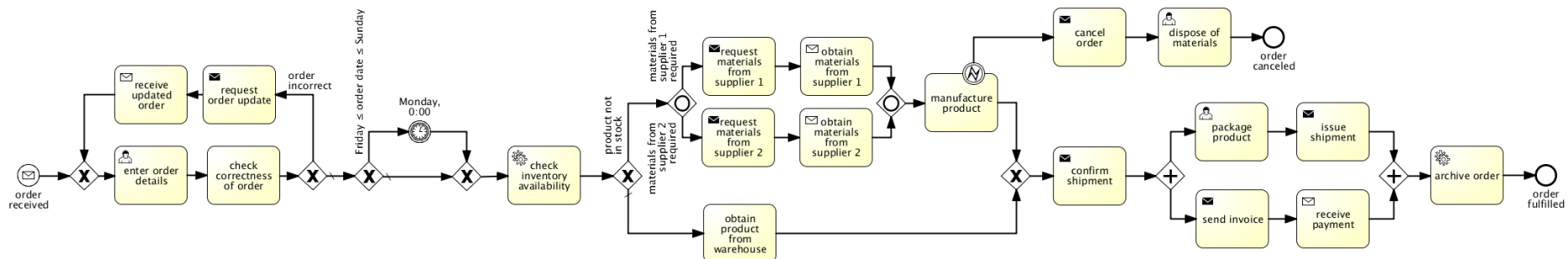


Modeling Guideline

*Size of Process Models:
Each model shall fit on one page.*

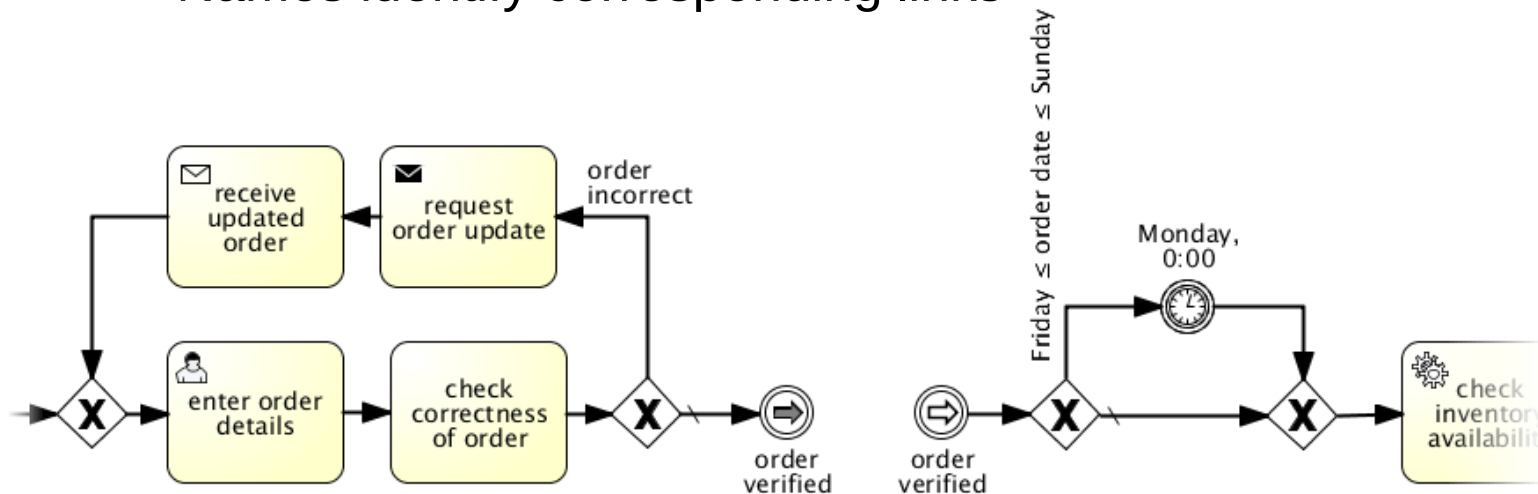
- Motivation

- Process models can provide a common understanding only, if they are understood by all stakeholders
- Process models with hundreds of elements are, however, incomprehensible



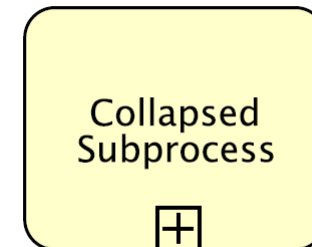
Splitting up by Link Events

- If a process model does not fit a reasonable space, it can be split and distributed over several pages
- Link events “cut” sequence flow
 - Only intermediate link events
 - Sequence flow from the throwing to the catching event
 - Names identify corresponding links



Sub-processes

- Activities
 - Tasks, atomic units of work
 - Sub-processes, activities with an internal structure
- Sub-processes provide hierarchical decomposition, which is very useful for coping with complexity
- Collapsed sub-process
 - The internal structure is hidden
 - A collapsed sub-process references another process model
 - There cannot be multiple references to a given sub-process
 - Exception: Call activities, discussed below

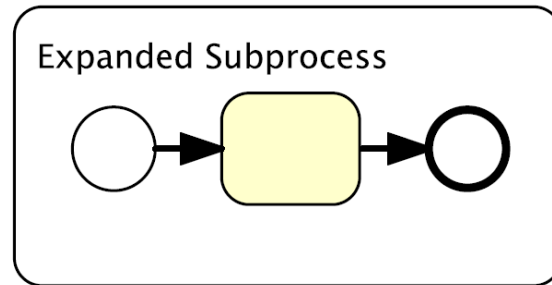


Collapsed Sub-Processes

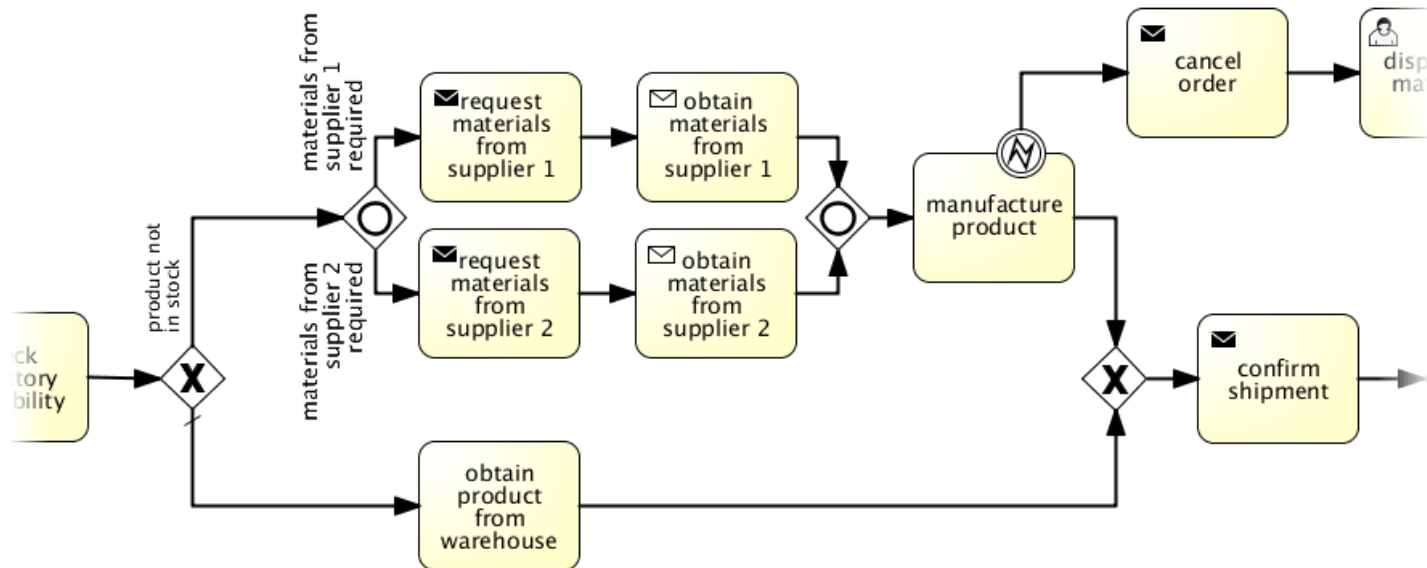
- Provide hierarchical decomposition

Expanded Sub-Processes

- Provide a mechanism for specifying parts of a business process that logically belong together
 - Expanded sub-processes expose their inner structure

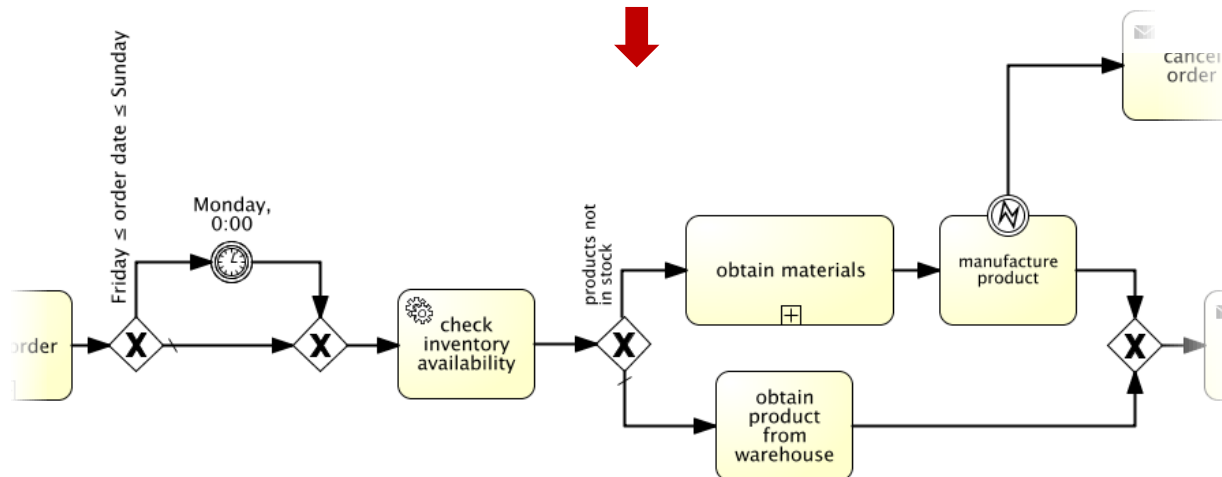
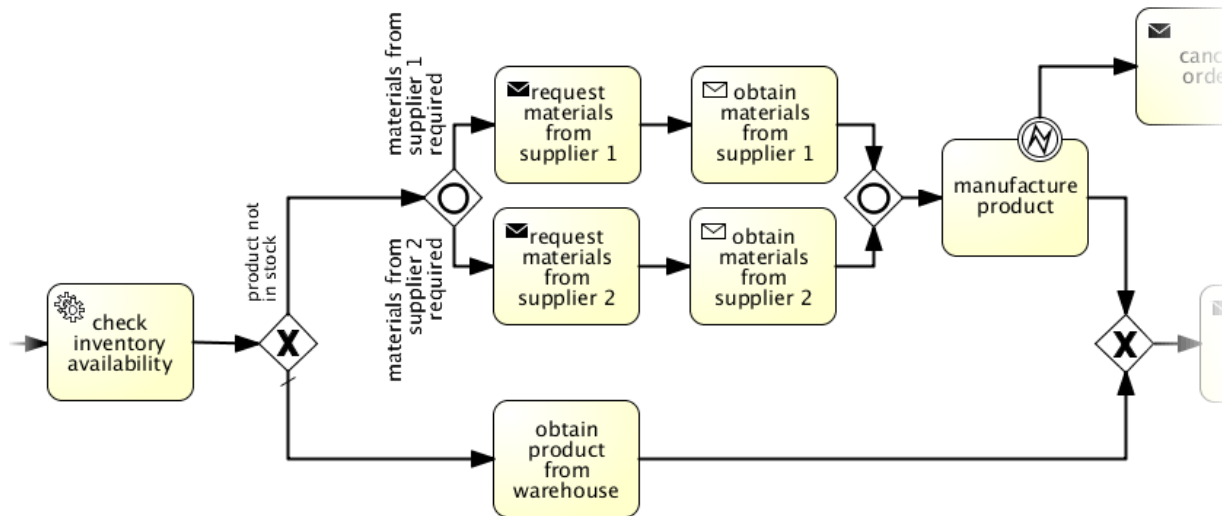


Application Scenario Sub-Process

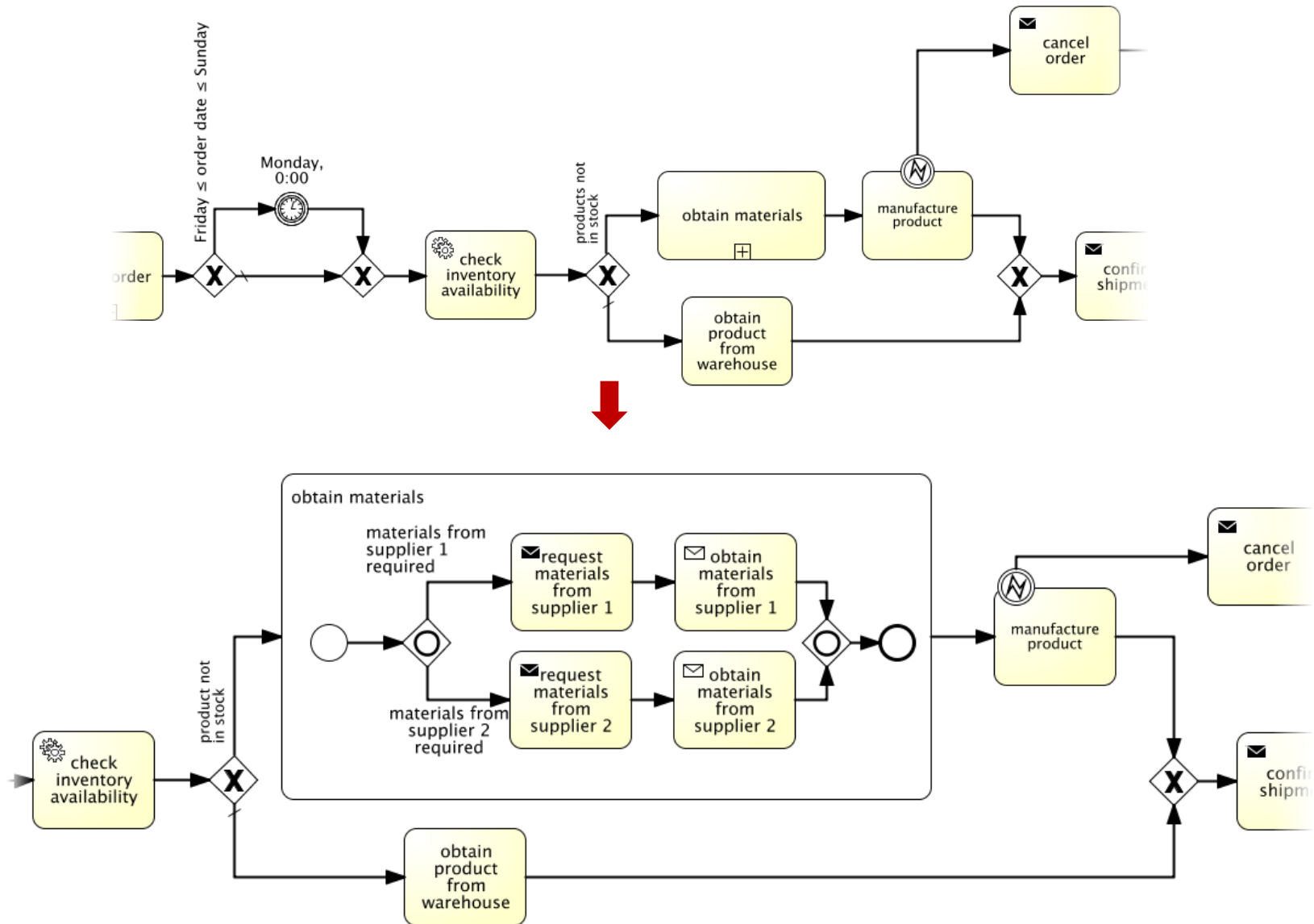


- Regarding the modeling goal, the suppliers and the way the material is obtained might not be relevant
 - However, in the process model, the activities are represented and lead to model complexity, which is not required
- We can use sub-processes to hide this complexity

Collapsed Sub-Process

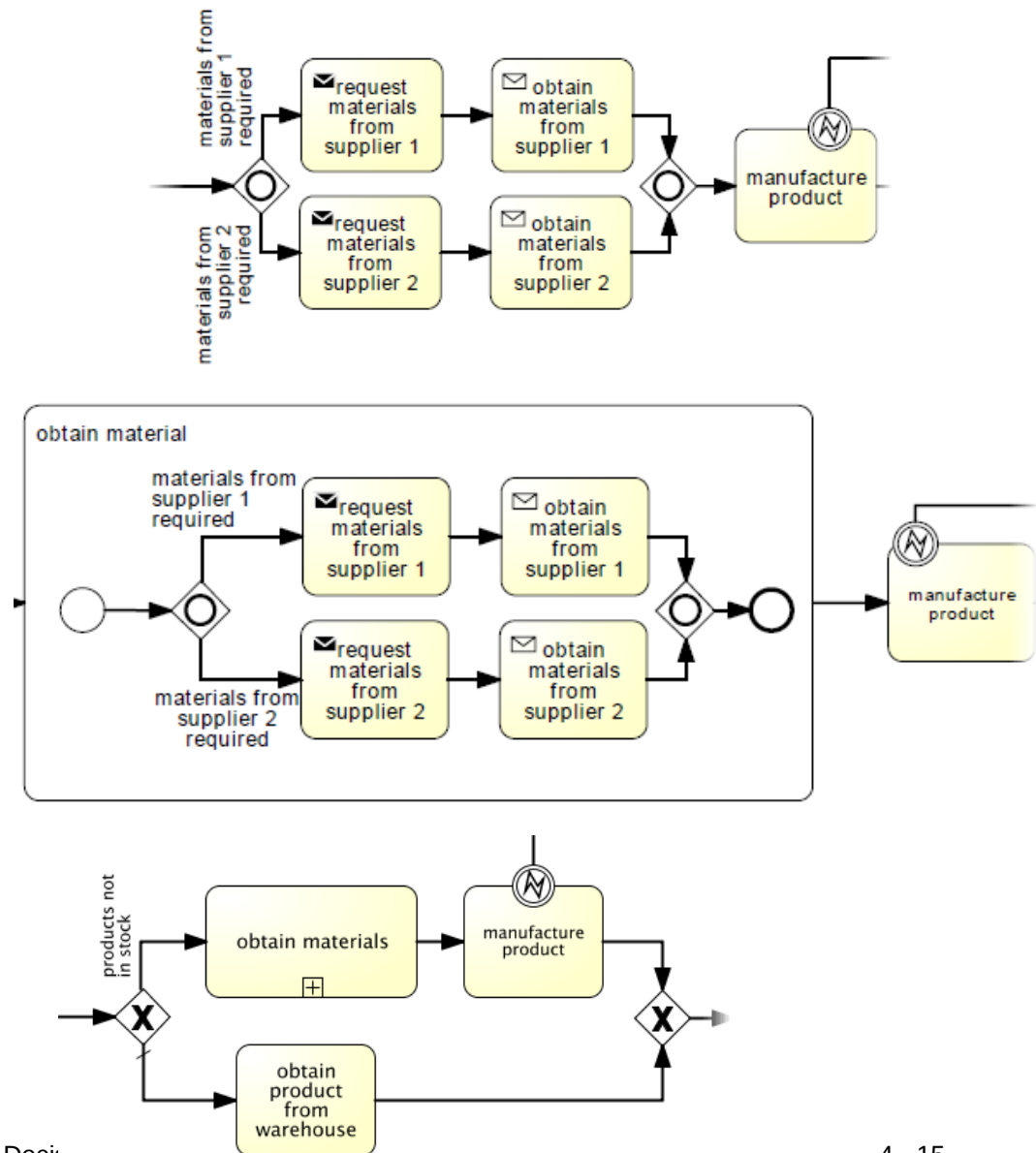


Expanded Sub-Process



Sub-Processes and Abstraction

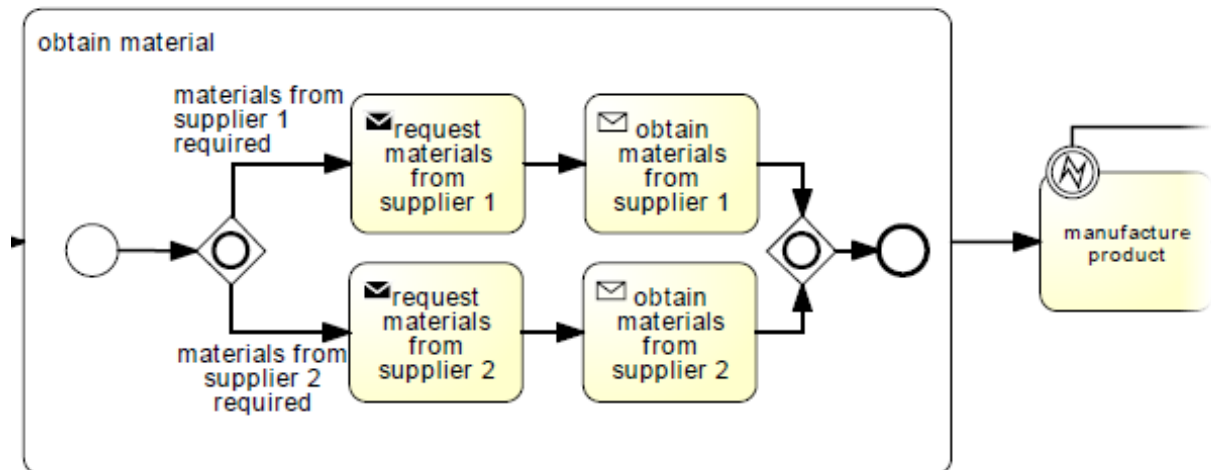
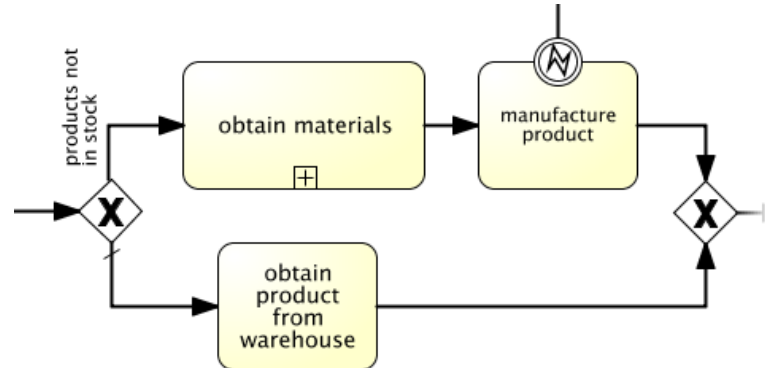
- Detailed, flat process model
- Expanded sub-processes provide logical grouping
- Collapsed sub-processes provide hierarchical decomposition and hide complexity



Video Clip 4.1

Sub-Processes

- Collapsed sub-processes
 - Hierarchical decomposition
 - Information hiding, reduced complexity
- Expanded sub-processes
 - Logical grouping of parts of process models



Video Clip 4.2

Sub-Processes and Boundary Events

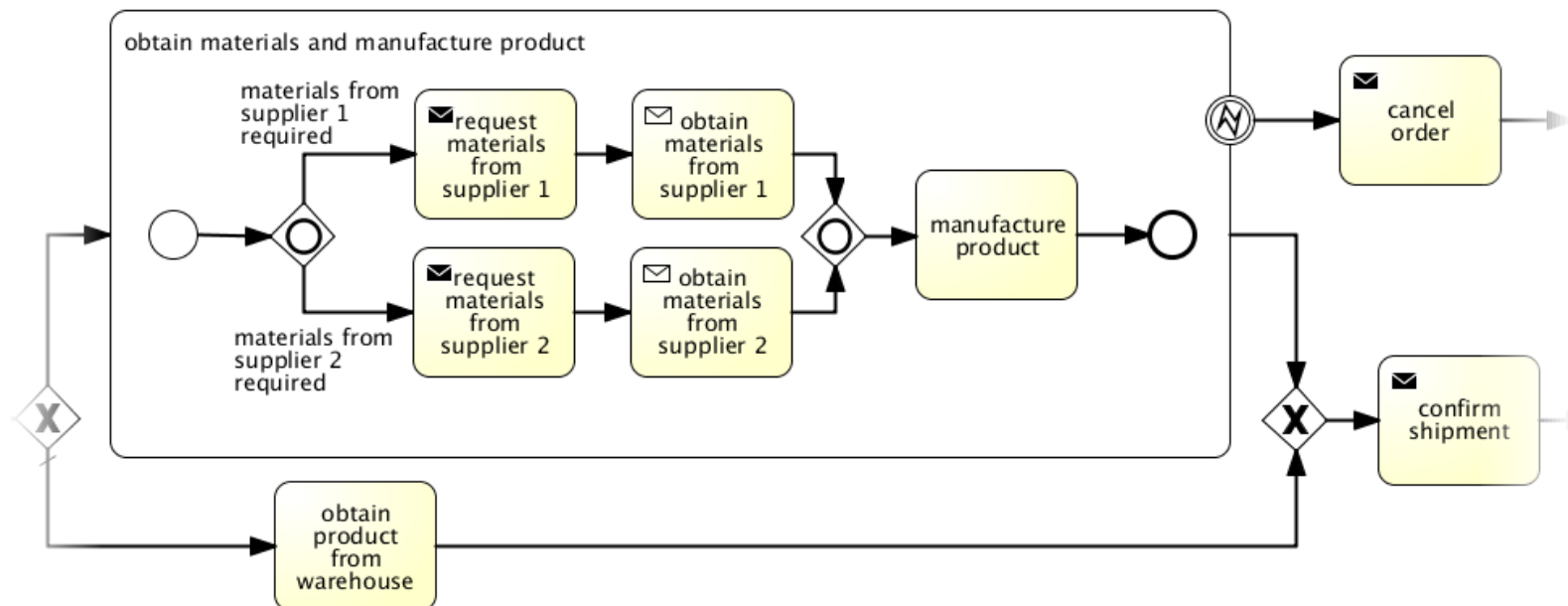
BPMN Meets DMN: Business Process and Decision Modeling:
An openHPI Course by Mathias Weske

Sub-Processes and Boundary Events

- Sub-processes can have boundary events
 - Same semantics as discussed for boundary events of tasks in Week 2
- If a boundary event occurs during a sub-process,
 - the outgoing flow of the event is signaled and either
 - the sub-process is interrupted („interrupting boundary event“), or
 - the sub-processes continues („non-interrupting boundary event“)
- When the sub-process starts
 - The process listens to the event
- When the sub-process terminates
 - The process stops listening to the event

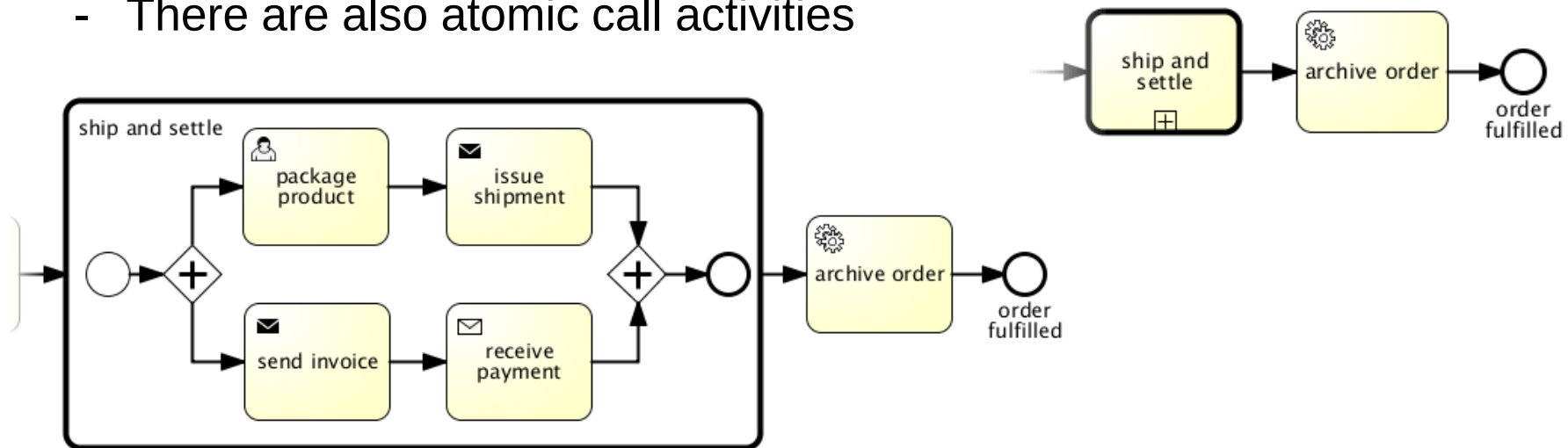
Sub-Processes and Boundary Events

If during execution of the expanded sub-process „obtain material and manufacture product“ a boundary error event occurs, the sub-process is interrupted.
(Notice that this is a variation of the process discussed before.)



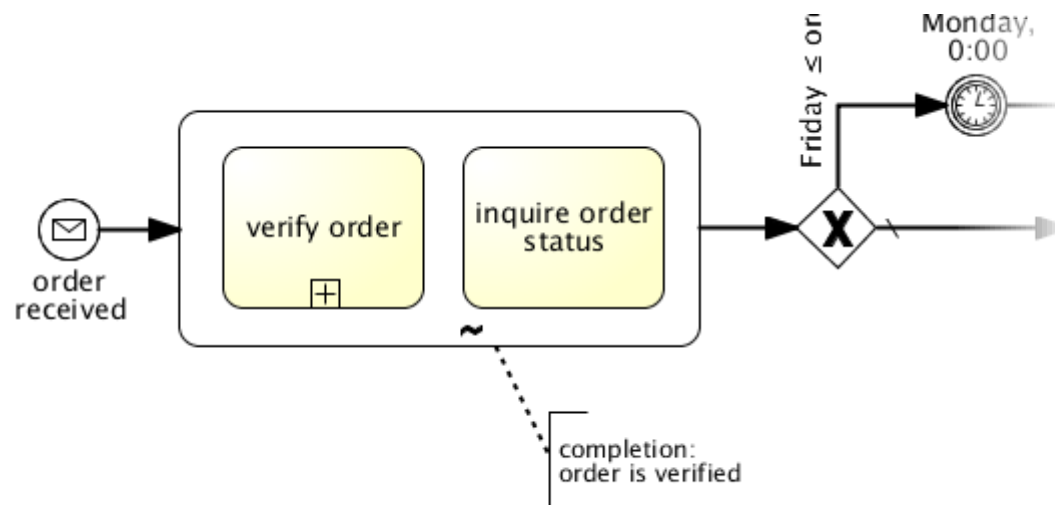
Sub-Processes and Reuse

- Sub-processes can be reused in a number of process models by using call activities
- A call activity is globally known and can be referenced in any business process model, several times
 - Notation: activity has a bold border
 - Upon execution, the call activity is invoked in the context of the calling process
 - There are also atomic call activities



Ad-hoc Sub-Processes

- Sometimes activities may appear in any order and be repeated until some condition is satisfied
- The ad-hoc sub-process groups a number of activities
 - Their ordering and the number of executions of activities is determined by the performer of the ad-hoc sub-process
 - Activities do not need to be sequential
 - A completion condition defines, when the sub-process terminates (stored as attribute; annotation only for explanation)



Modeling Guideline

Ensure consistency of events of a sub-process with those of the parent process.

- Motivation
 - A sub-process can be started, when the sub-process activity is signaled, i.e., when it is in the ready state
 - A sub-process may only start with a blank start event or with an activity.
 - Sub-processes with more than one alternative blank end event shall be followed by an XOR gateway in the parent process
 - Non-blank end events, e.g., error end events, must be caught by boundary events attached to the sub-process activity

Sub-Process and End Events

- Sub-processes with more than one alternative blank end event shall be followed by an XOR gateway in the parent process

Sub-Process and End Events

- Events can be thrown from within the sub-process to be caught by the sub-process.

Video Clip 4.2

Sub-Processes and Boundary Events

- Boundary events
 - Throw an event in a sub-process
 - Catch it at its boundary
- Re-use through call activities
- Ad-hoc sub-processes
 - Can express unstructured process parts
- Modeling guidelines

Video Clip 4.3

Activity Modifiers

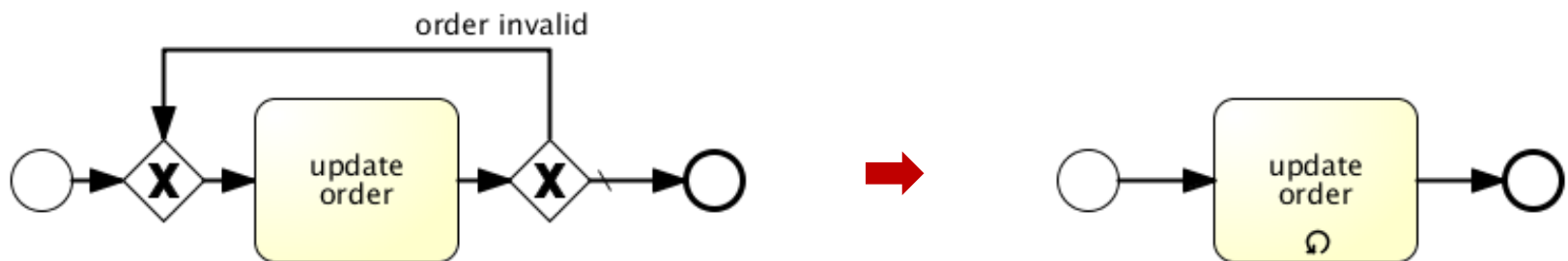
BPMN Meets DMN: Business Process and Decision Modeling:
An openHPI Course by Mathias Weske

Activity Modifiers

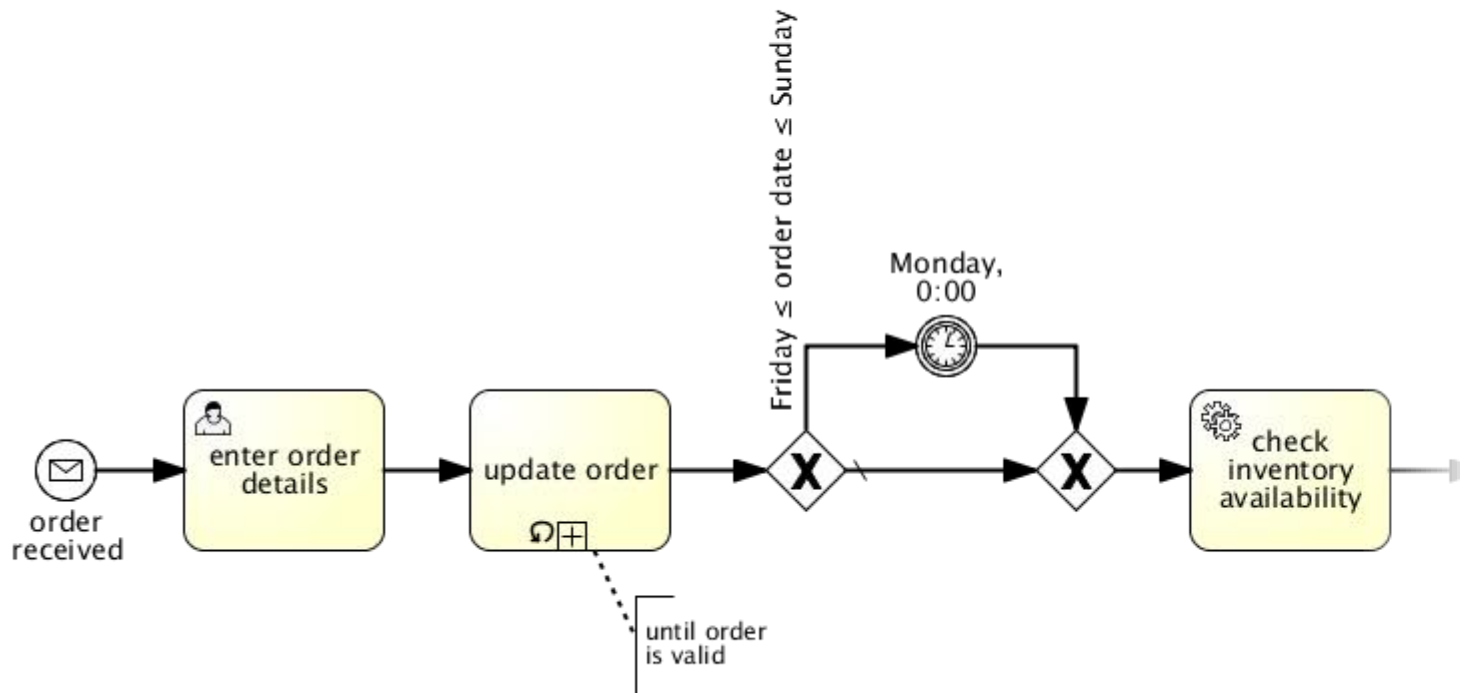
- Activities can be enriched with markers that represent specific execution semantics
- Activity modifiers can be attached to
 - Atomic tasks and
 - Sub-processes

Loop Activities

- Repetition of an activity can be expressed by a loop activity marker
- Number of repetitions can be configured by a loop condition (stored as attribute)
 - Condition can use number of loops (loop counter) or values of data objects (discussed in Week 5)
 - For-loop / while-loop
 - Condition can be tested before or after execution of the activity
 - While-loop / repeat-until-loop



Application Scenario (cont.)



Multiple Instance (MI) Activities

- Compact representation of activities that shall be executed multiple times for one process instance
- Different activity modifiers for

- sequential execution



- parallel execution



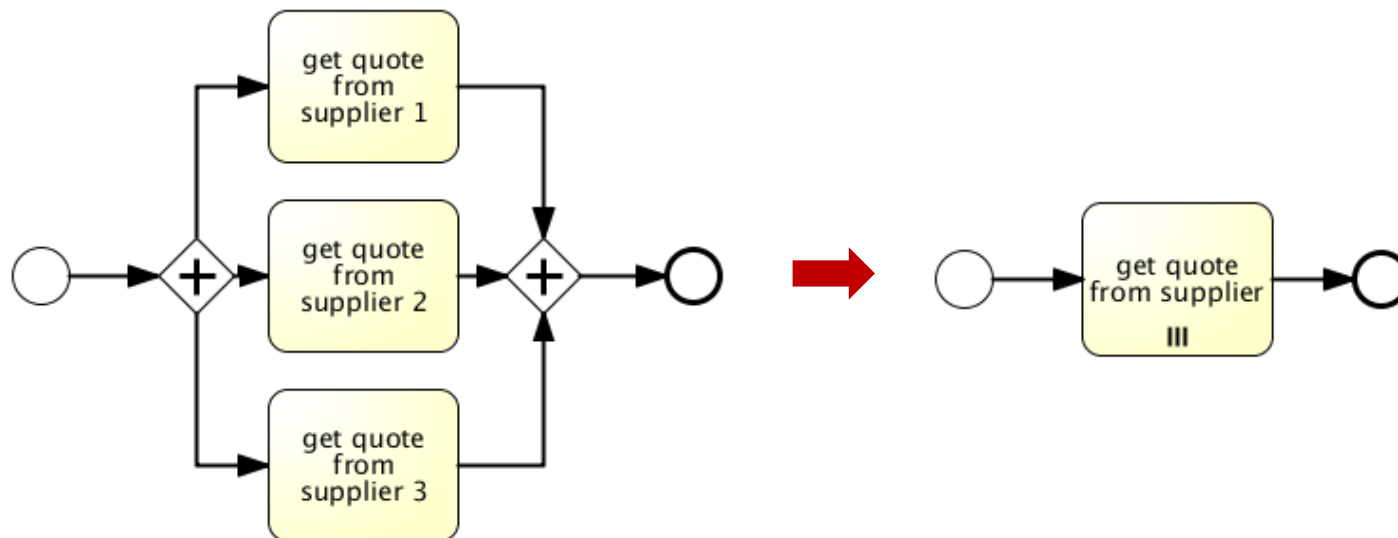
- MI activities are synchronized, i.e., process continues only after all MI activity instances have been terminated

Multiple Instance Activities

- Sequential execution

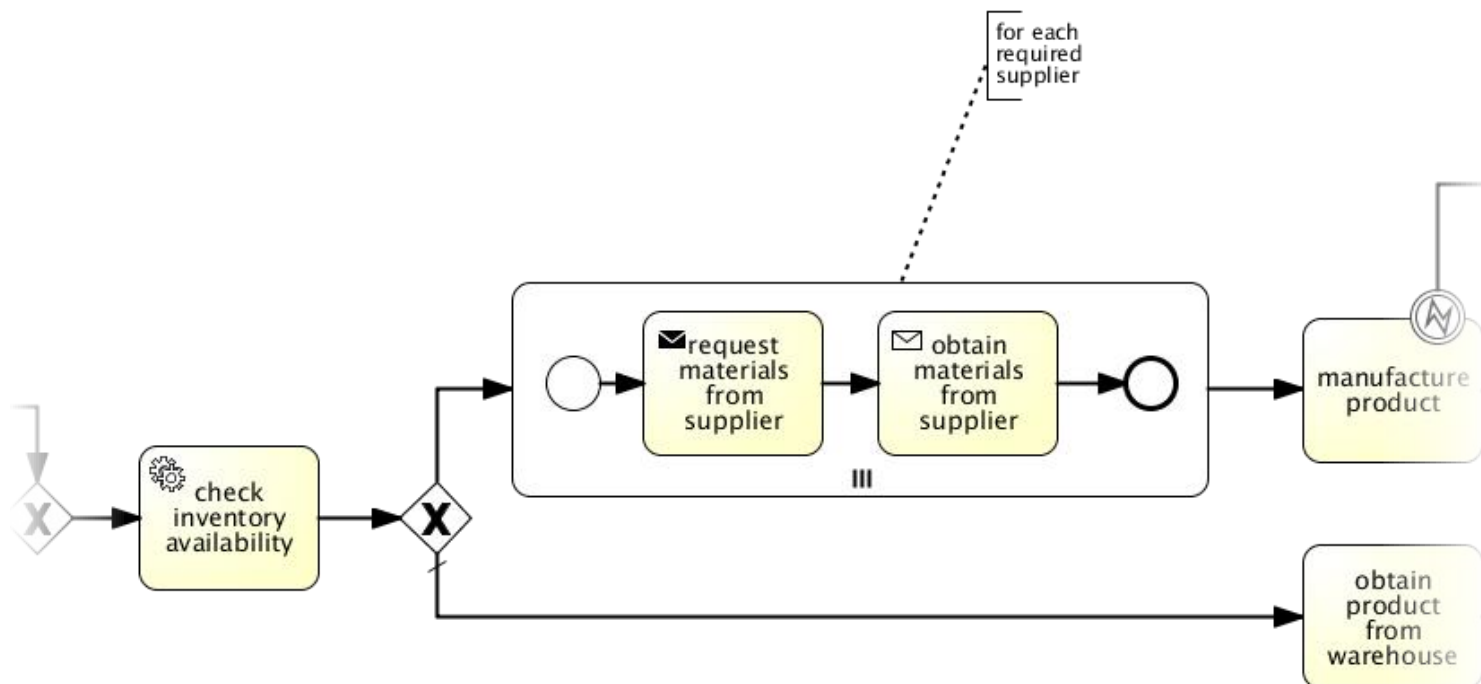


- Parallel execution



Application Scenario (cont.)

- Expanded sub-processes can also have a multiple instances marker
 - The sub-process provides a scope for sending and receiving messages (to be discussed in Week 6)



Video Clip 4.3

Activity Modifiers

- Modifiers add execution semantics to activities
- Provide abstract representation of process behavior
- Loop
 - Hide looping details in attributes
- Multiple instances
 - Parallel
 - Sequential

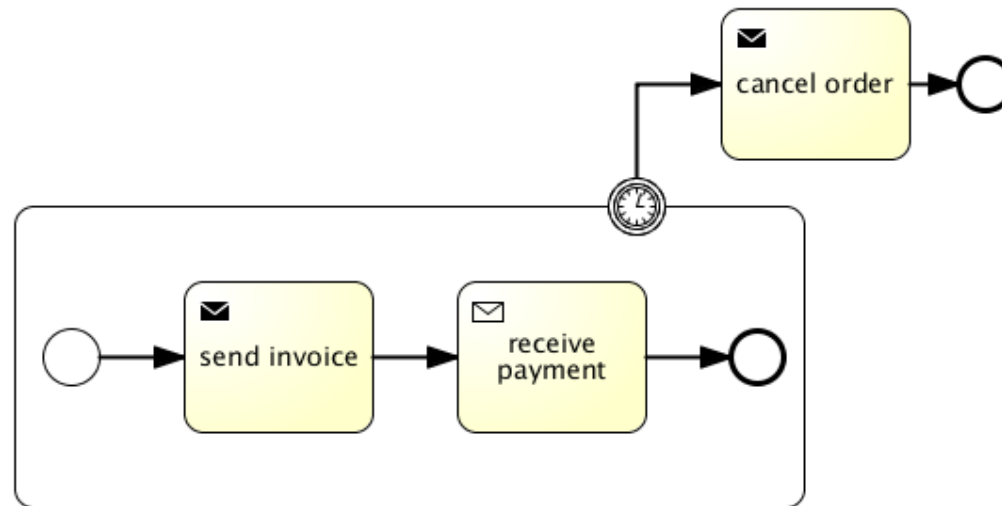
Video Clip 4.4

Racing Events

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

Racing Events

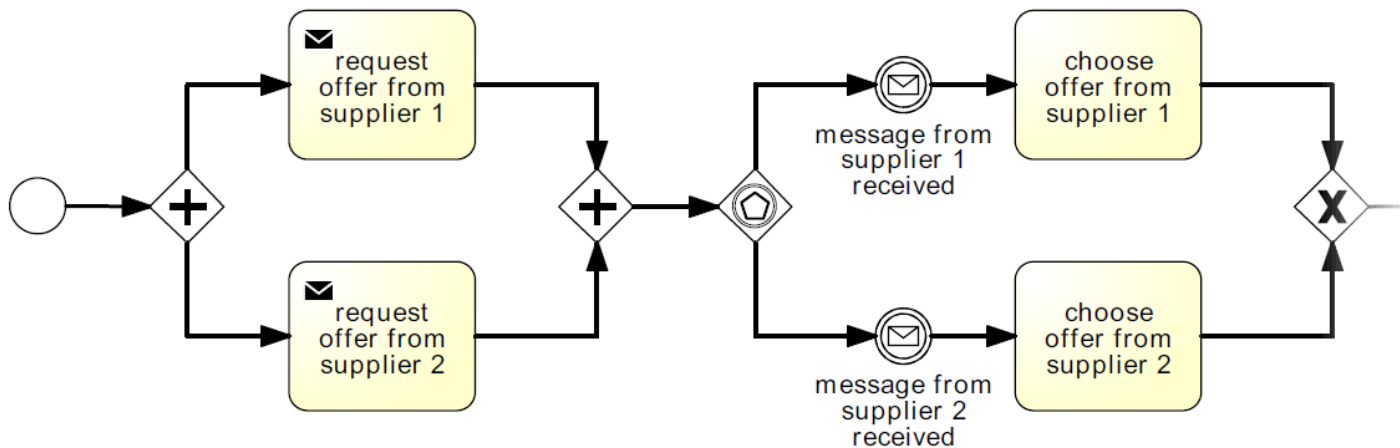
- Recap of boundary events from Week 2
 - A boundary interrupting event can cancel a sub-process, if it is observed while the sub-process executes
- Race condition between boundary event and sub-process termination



- There are further race conditions between events in process models, and a dedicated gateway

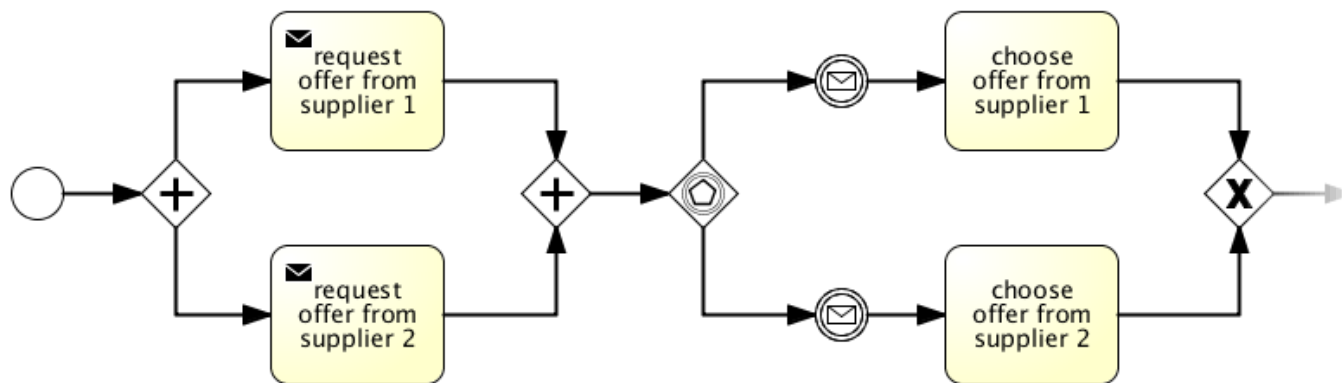
Event-based Gateway

- BPMN provides a particular gateway which is used to select from a number of racing events the first one that occurs
- Event-based gateway
 - Can only be used as a split gateway
 - Can safely be merged using an XOR join gateway, as it has exclusive split semantics



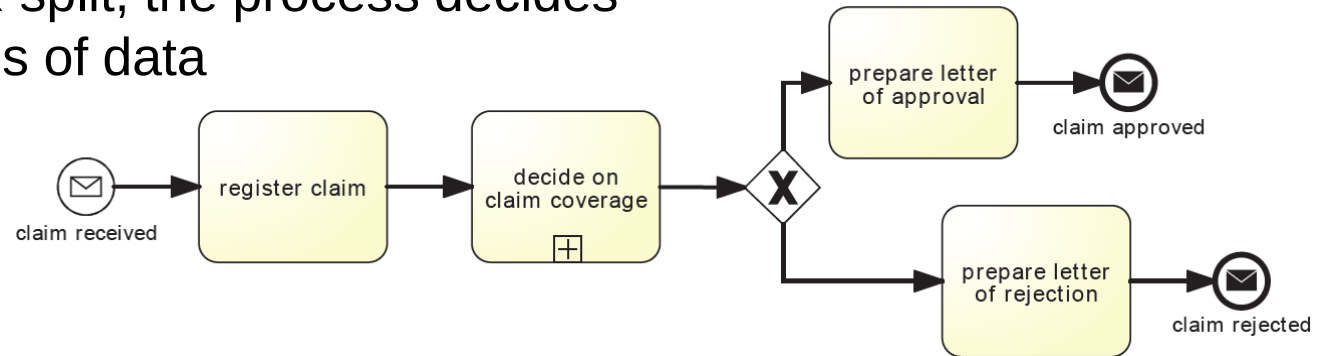
Event-based Gateway

- The event-based gateway must be followed by intermediate events or receive tasks
- When firing, the gateway enables all events on its outgoing edges
 - Upon the occurrence of either event, the others are disabled
 - Only the outgoing path of the observed event is then signaled

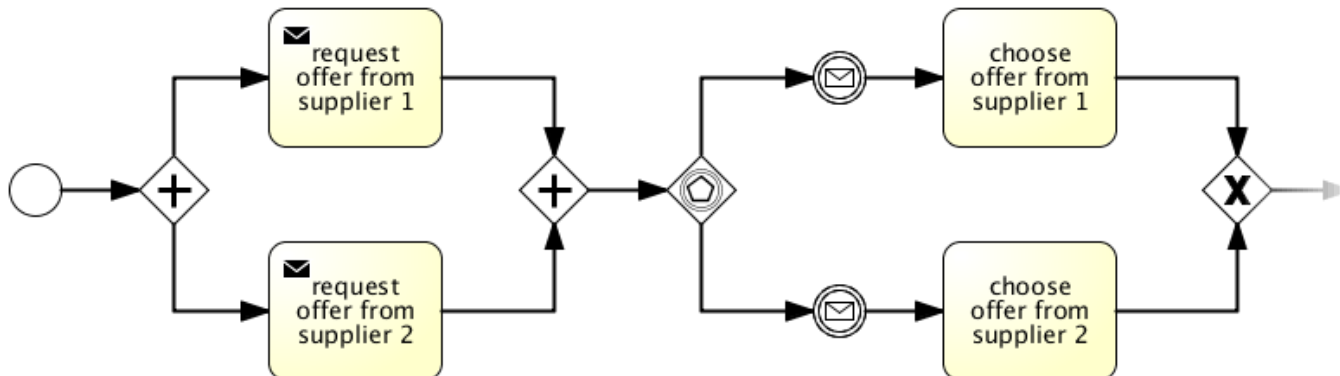


Difference to XOR Split

- XOR split
 - In the XOR split, the process decides on the basis of data

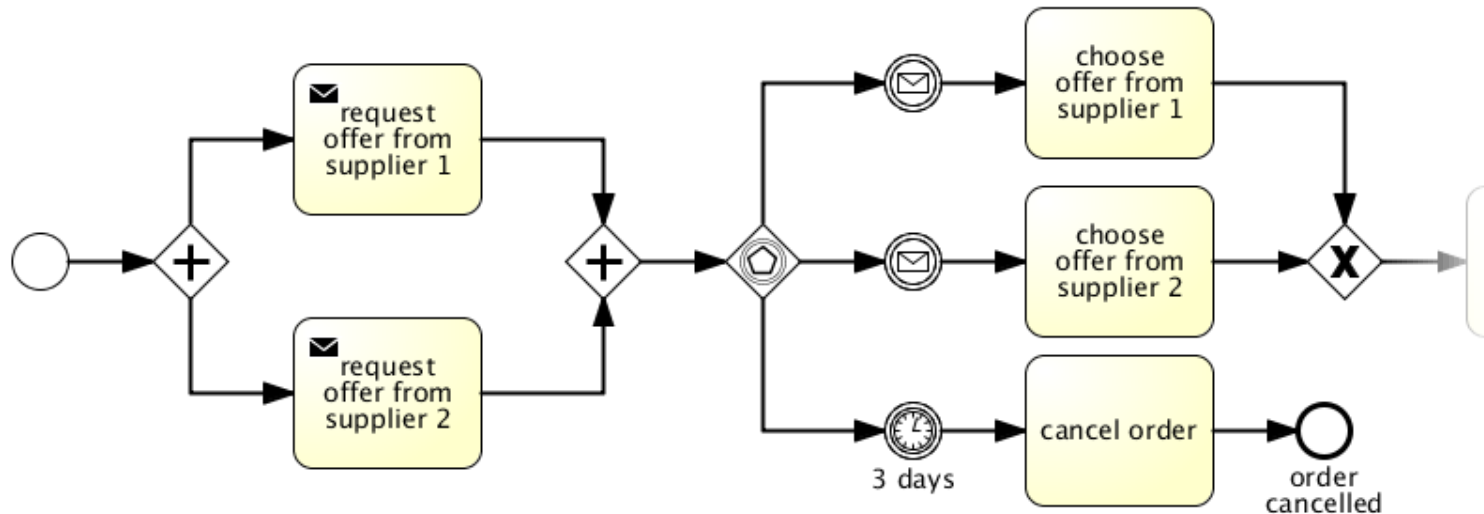


- Event-based gateway
 - Decision is taken by the process environment
 - Depending on which event occurs first



Event-based Gateway

- To avoid the process getting stuck, a timer event should be used
 - This is similar to the default flow of XOR gateways



Video Clip 4.4

Racing Events

- Boundary Events
 - Race between termination event and boundary events
- Event-based Gateway
 - Race between multiple events
 - Decision is taken by the environment

Video Clip 4.5

Modeling Organization

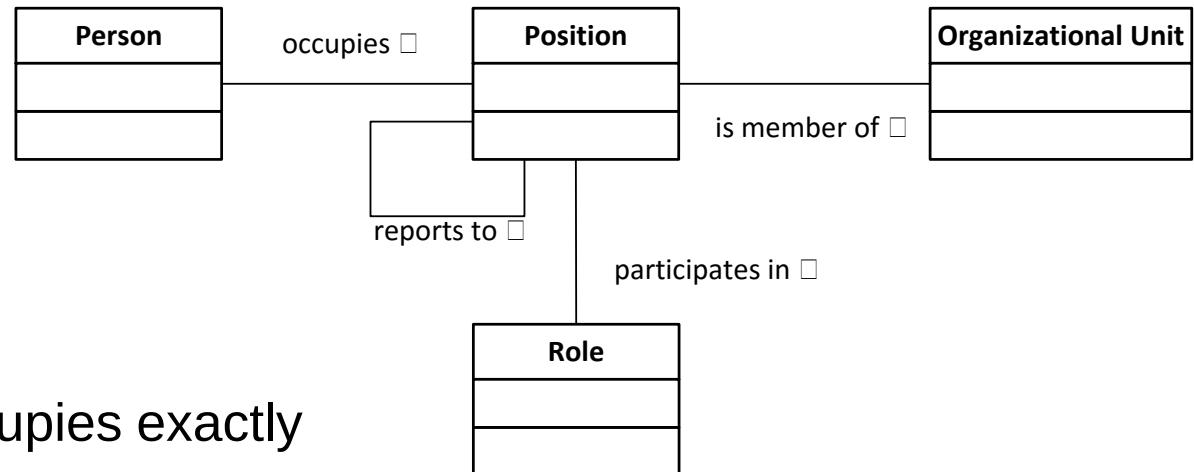
BPMN Meets DMN: Business Process and Decision Modeling:
An openHPI Course by Mathias Weske

Coordination of Work among People

- Recall
 - “A business process consists of a set of activities that are *performed in coordination in an organizational [...] environment.*”
 - Distribution and coordination of work among people is an important task of a BPM system
- Prerequisites
 - Information about the structure of an organization is required
 - This information needs to be connected to process diagrams

Organizational Modeling

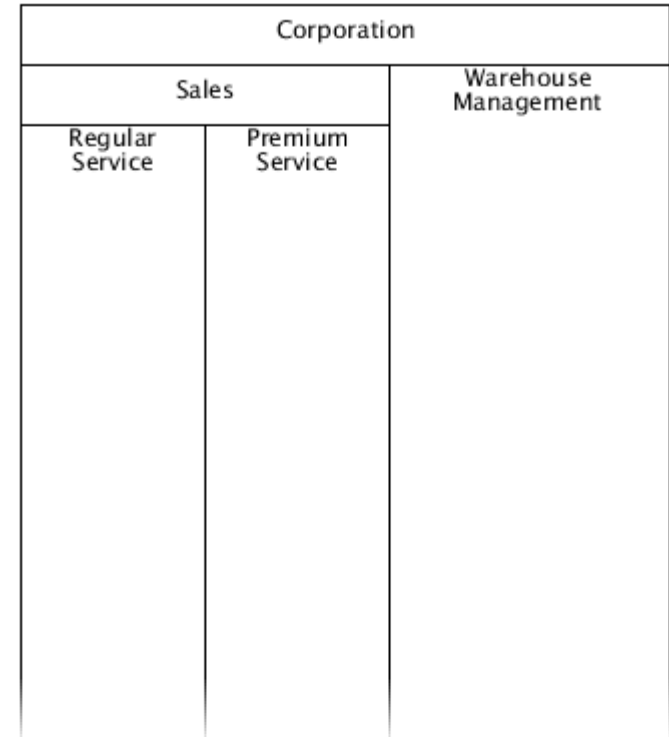
- We use a simple organizational model



- Assumptions
 - Each person occupies exactly one position
 - Duties and rights are associated with positions, not directly with persons
 - Positions can participate in roles
 - Organizational units are permanent groupings of persons based on their positions

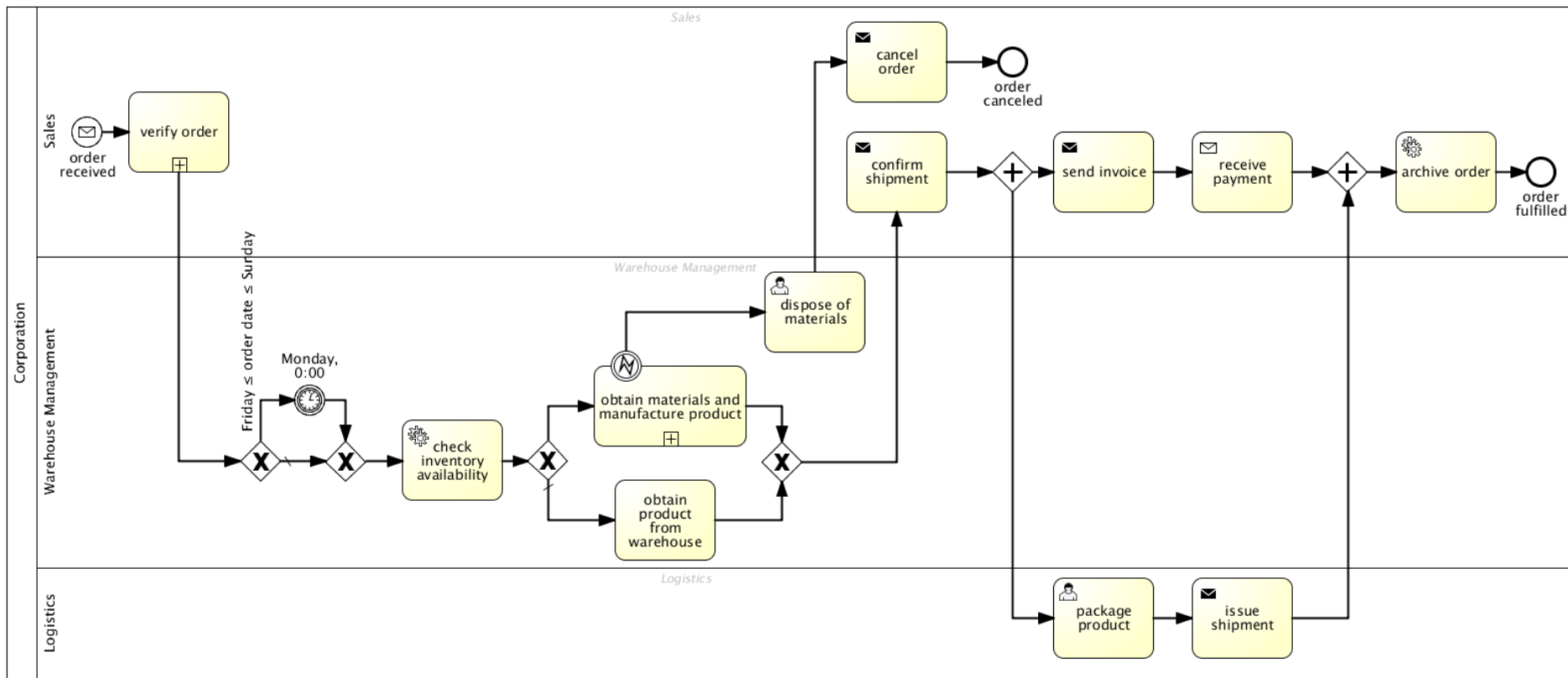
Pools and Swimlanes

- In business process diagrams, organizational information is represented by pools and swimlanes
- Pools represent organizations
 - Each pool contains a business process
 - Each business process is contained in a pool (might be implicit)
- Lanes represent resources within an organization
 - Positions, roles, organizational units, but also persons
 - Lanes can be nested (pools cannot)



Resource Assignment

- The assignment of activities to resources is done by placing them in the respective pool or lane
- Sequence flow may cross lanes, indicating handover of work

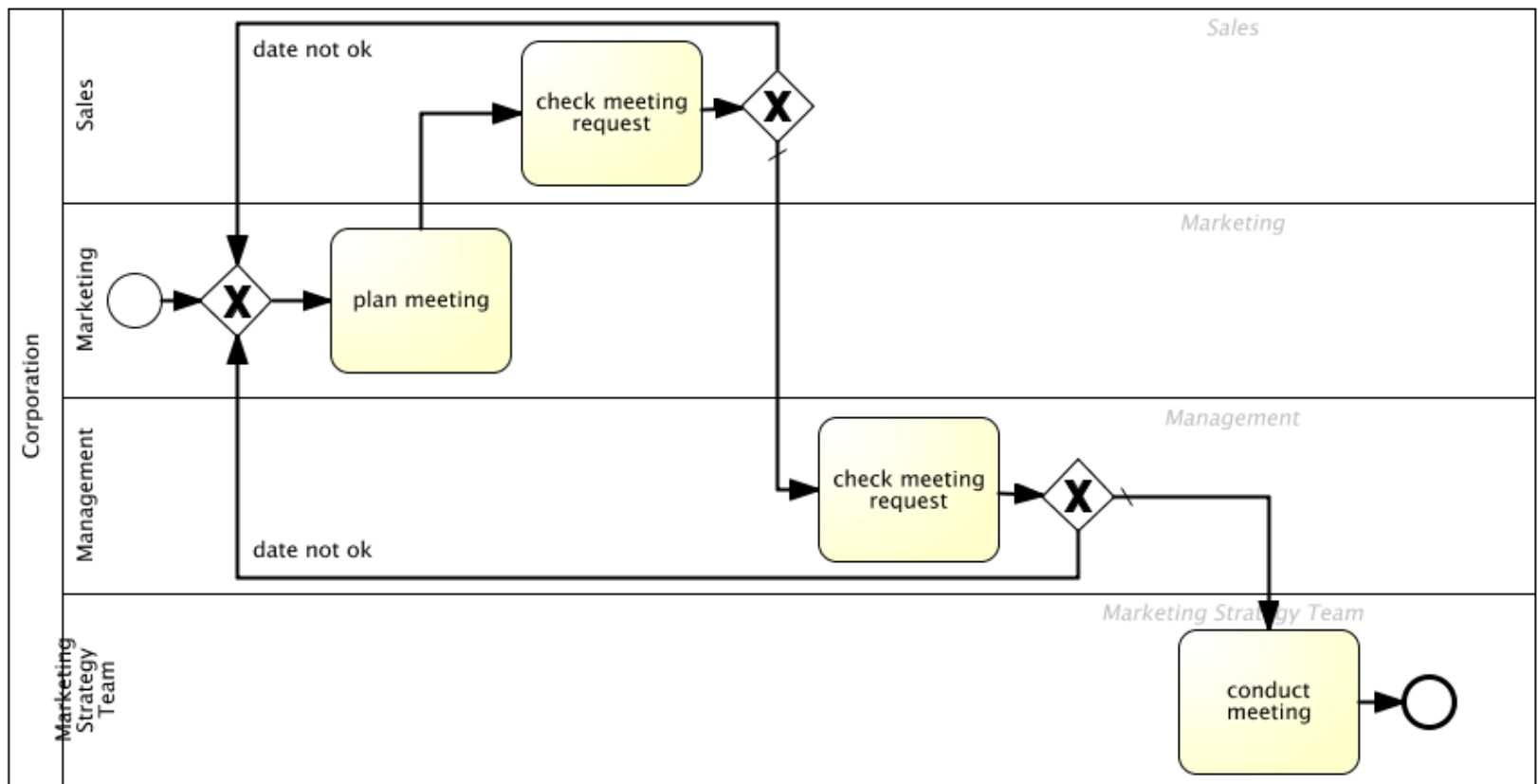


Modeling Guideline

If several roles participate in one activity, create a new role. Do not replicate the activity for each role.

- Motivation
 - Activities are assigned exactly to one swimlane
 - Some activities require the collaboration of several roles, e.g., in a meeting
 - Replication of the activity would lead to several activity instances during execution, although only one instance exists
 - Instead, create a new role that captures the duties and responsibilities of the collaborative task
 - This is feasible as each person can participate in several roles

Modeling of Collaborative Tasks



Video Clip 4.5

Modeling Organization

- Employees are organized in
 - Roles
 - Positions
 - Organizational units
- Process models contain organizational information
 - Pools and swimlanes
 - Can be used to analyze handover between resources
- Collaborative tasks

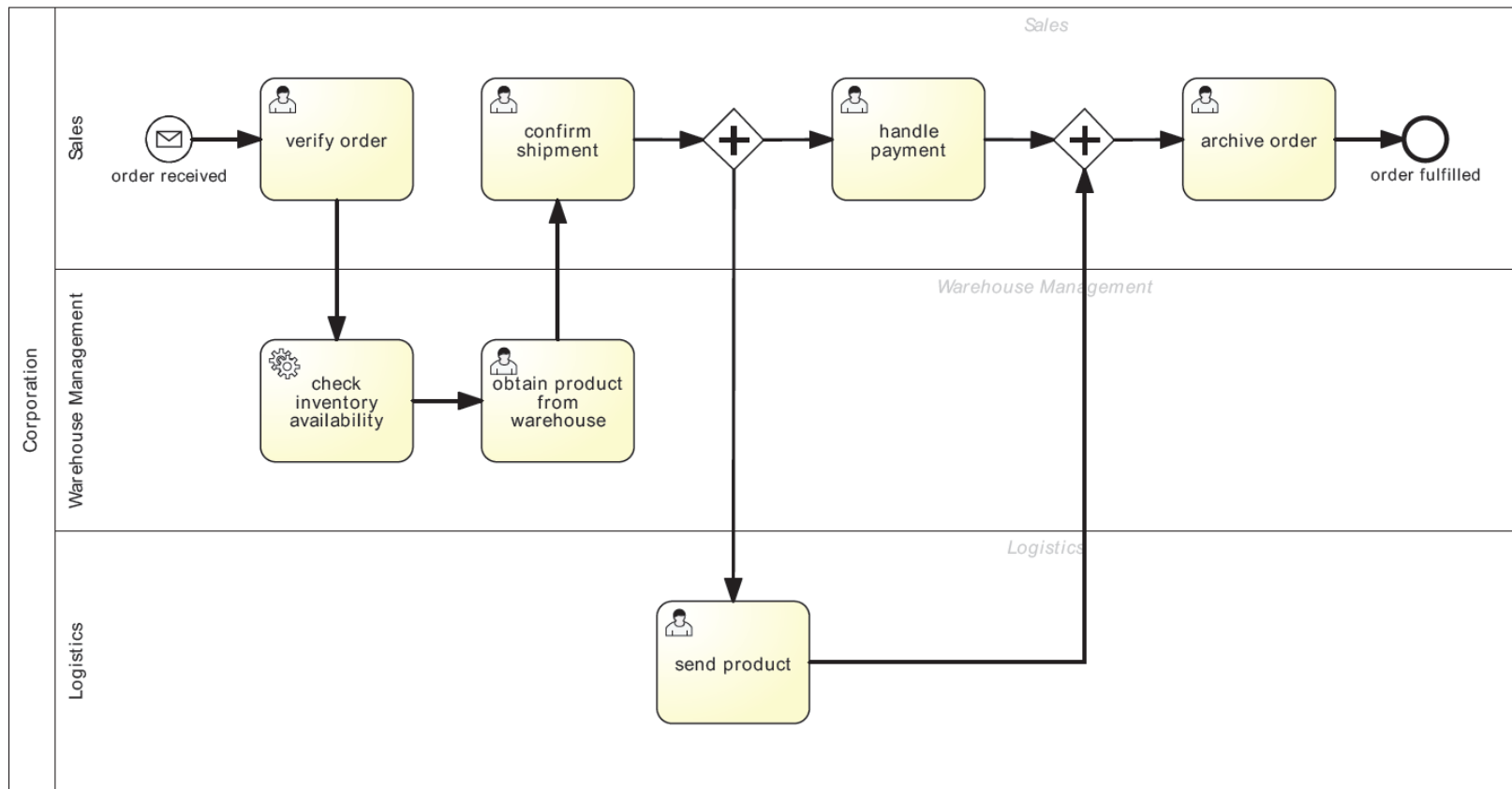
Video Clip 4.6

Resource Allocation Patterns

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

Resource Allocation

- During execution, activity instances need to be *allocated* to resources, most importantly, employees



Sales

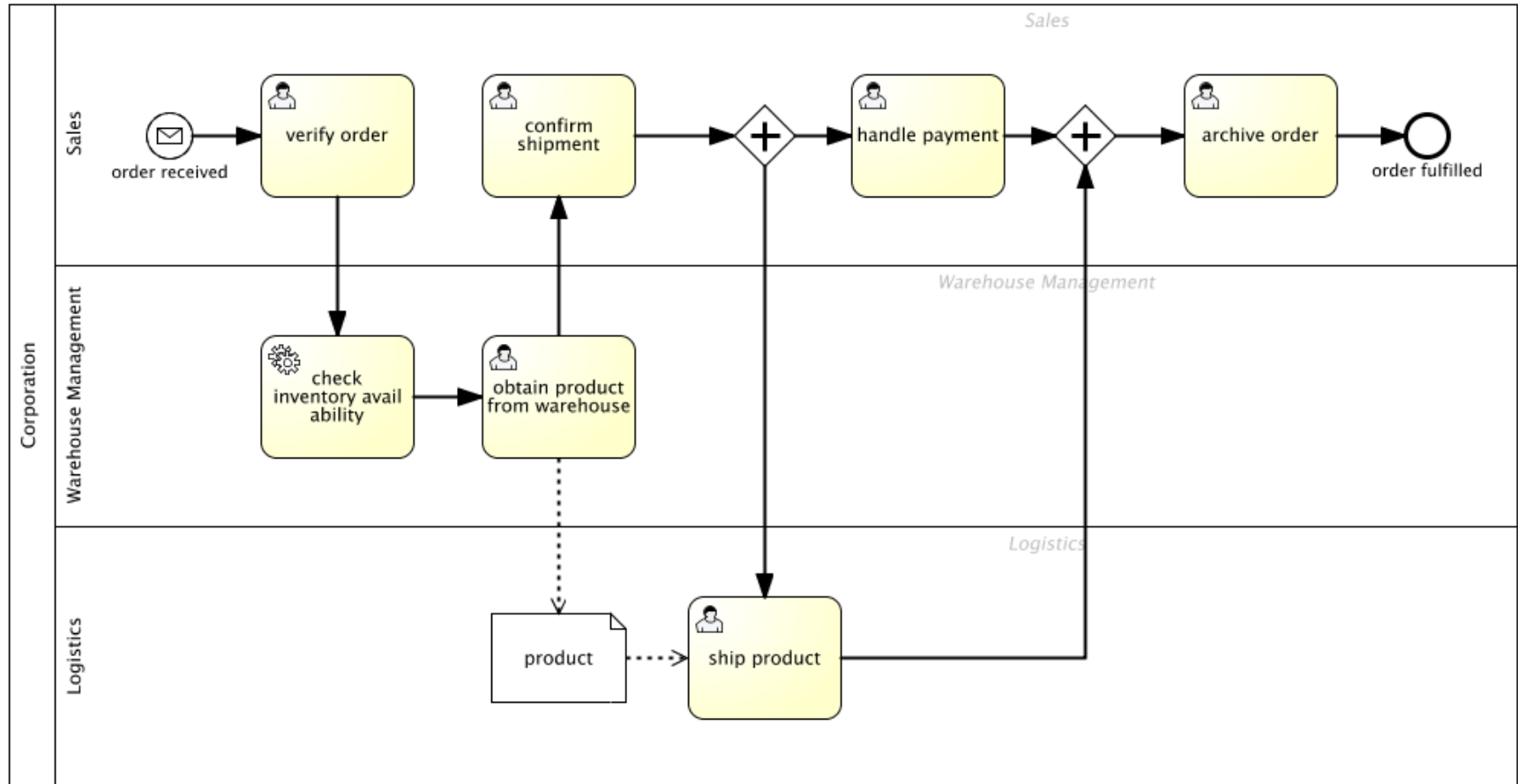
- Anne
- Bridget
- Charles

Warehouse Mgt

- Daniel
- Eva

Logistics

- Frank
- George



Role-based Allocation

- Process model specifies a role, each member of the role is capable of performing the task
- During runtime, the activity instance is offered to all persons available who are members of that role
- Example
 - Verification of a purchase order is offered to each member of the role “sales”
 - One employee of that role will take the activity instance and perform it

Direct Allocation

- An individual person is specified in the process model
- Useful, if skills of a particular person are required
 - E.g., changes to a legacy software system often require the participation of the original developer regardless of his or her current position in an organization



Direct Allocation

- Discussion
 - How about roles with one position and one person in that position?
 - Isn't that exactly direct allocation?

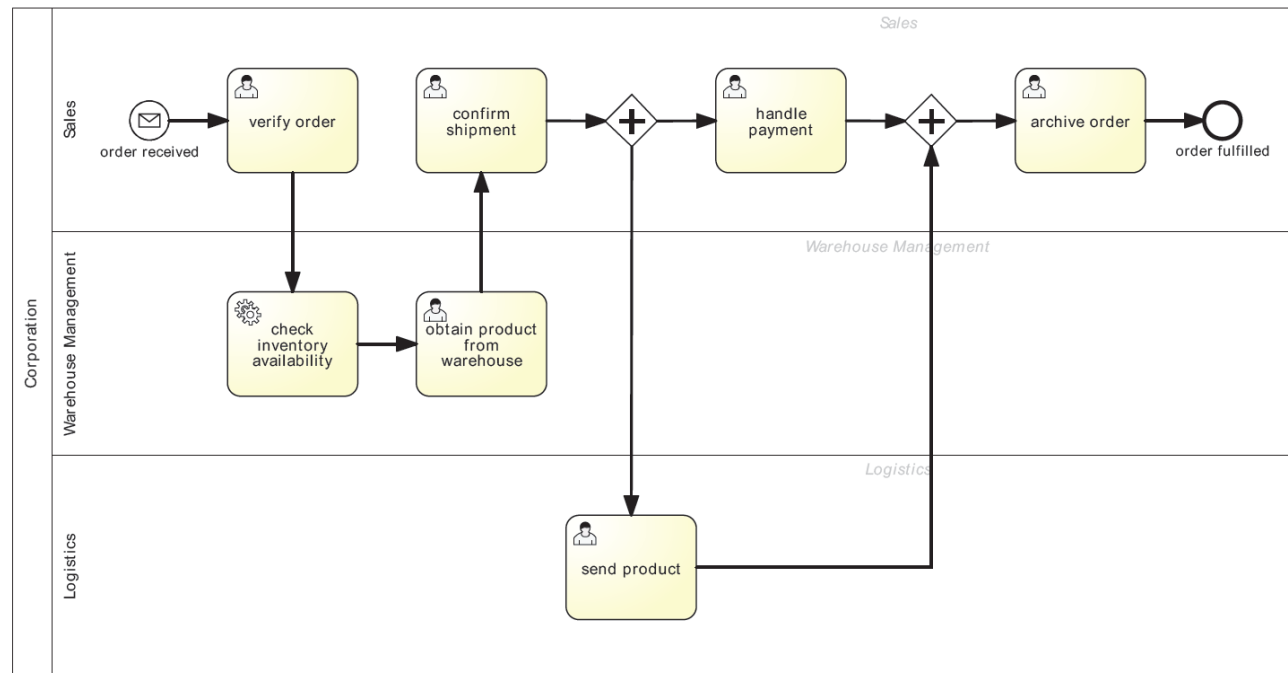


Case Handling

- Activities of one process instance (one “case”) shall be performed by the same person
 - The person who performed the first activity will also be assigned to the remaining activities of that instance
 - Might also be limited to a specific role

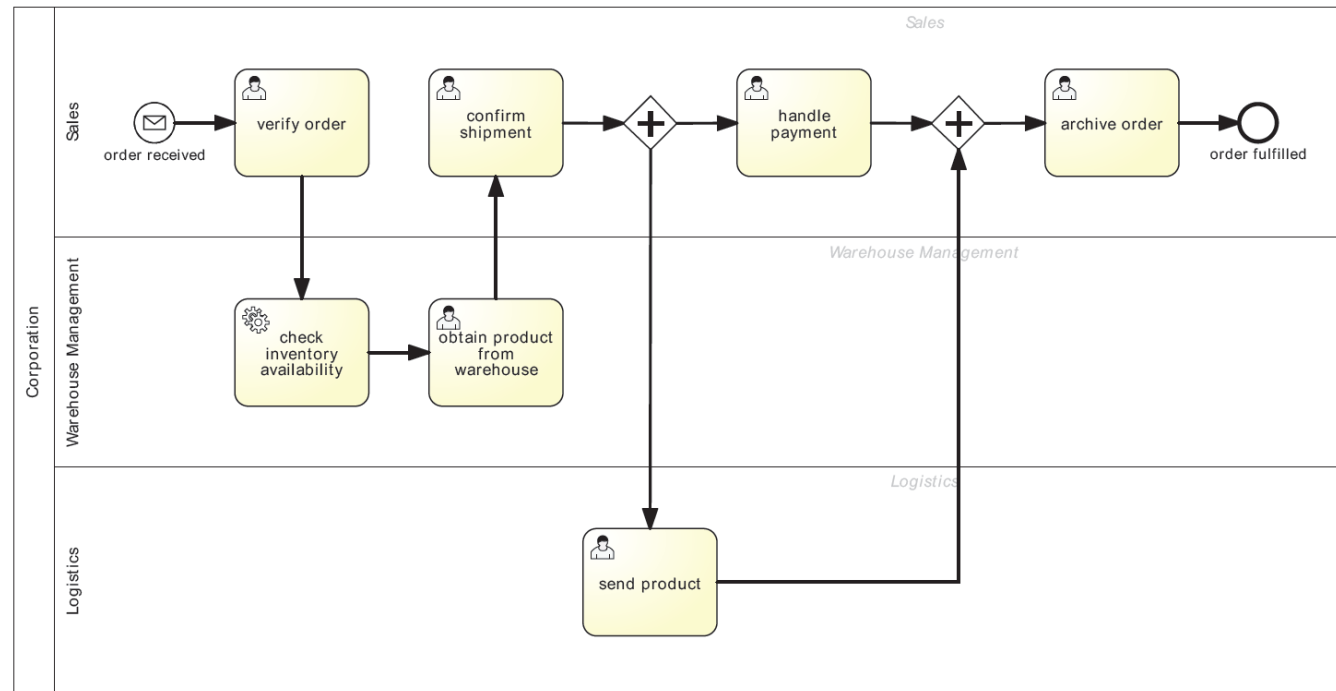
Example

- All interactions with a customer shall be carried out by the same sales person



History-based Allocation

- Resources are allocated based on their participation in previously executed process instances
- Example
 - Interaction with a customer shall be carried out by the same sales representative in all cases (“One Face to the Customer”)



Clip 4.6

Resource Allocation Patterns

- Work items encapsulate task instances
- Task assignment determines responsibilities
 - Roles
 - Positions
- Resource allocation distributes work items to employees of an organization
 - Role-based allocation
 - Direct allocation
 - Case handling
 - History-based allocation

Summary of Week 4

- Activities
 - Hierarchical decomposition and reuse with sub-process
 - Ad-hoc sub-process
 - Activity modifiers: multiple instance, loop
- Racing events
 - Interrupting boundary events
 - Event-based gateway
- Resource allocation
 - Organizational structures
 - Pools and swimlanes
 - Resource allocation patterns