

BPMN – Business Process Modelling Notation

Knut Hinkelmann



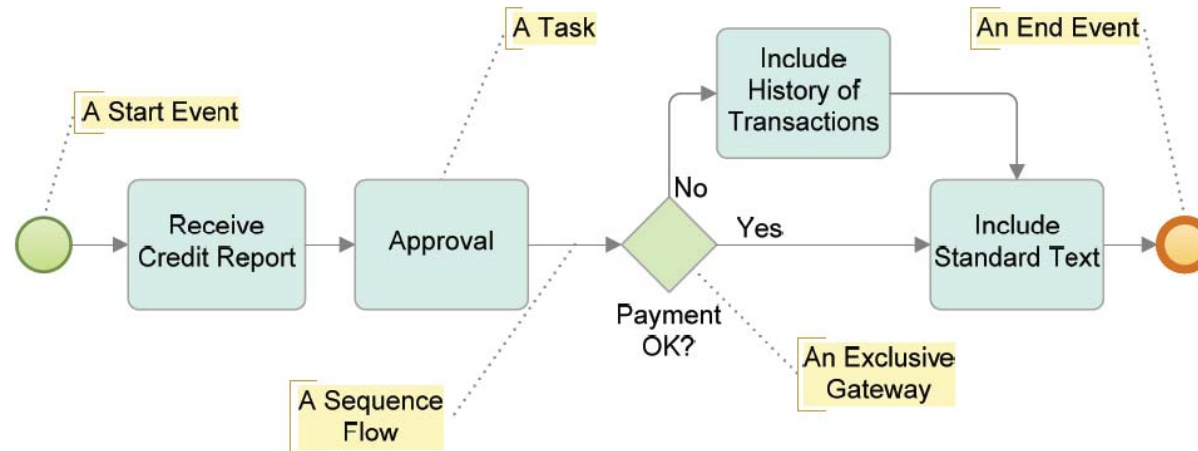
Literature

This chapter is mainly based on the following literature:

- Bridgeland David M.; Zahavi, Ron (2009): Business Modeling - A Practical Guide to Realizing Business Value. Morgan Kaufman Publishers. Chapter 5
- White, S.A., Miers, D. (2008) BPMN - Modeling and Reference Guide. Future Strategies Inc. Part 1: Understanding BPMN
- Stephen A. White (2006): Introduction to BPMN, http://www.bpmn.org/Documents/OMG_BPMN_Tutorial.pdf
- OMG Business Process Modeling Notation - BPMN, <http://www.omg.org/spec/BPMN/1.1/PDF>

What is BPMN?

- BPMN is a graphical modeling notation for business processes that is independent of a specific implementation environment



- BPMN was officially adopted as an OMG specification in 2006, updated in 2008 and now available in version 2.0 (Beta 2 released May 2010: <http://www.omg.org/spec/BPMN/2.0/Beta2>)
- BPMN provides a standardized bridge for the gap between the *business process design* and *process implementation*

Objectives of BPMN

- BPMN has two somehow contradictory objectives
 - ◆ to provide an **easy to use process modelling notation**, accessible to business users and business analysts
 - ◆ provide facilities to **translate models into an executable form** (such as BPEL – Business Process Execution Language)
- To meet the requirements of the first goal, BPMN is structured with a
 - ◆ small set of elements (Activities, Events and Gateways) that have
 - ◆ distinct shape (rectangle, circle and diamond).

Events



Activities










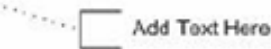



Gateways



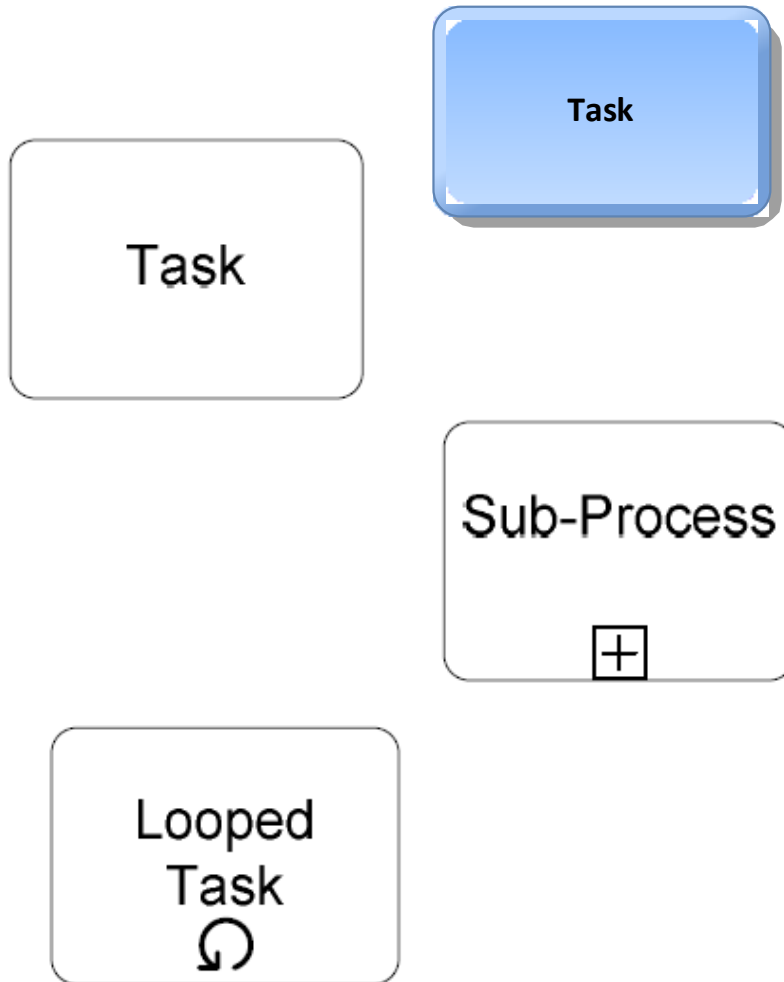
This small set supports simplicity and readability of models

Elements of BPMN

Elements of BPMN can be divided into 4 categories:

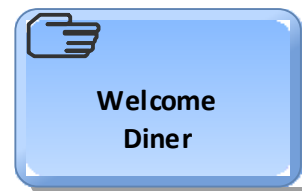
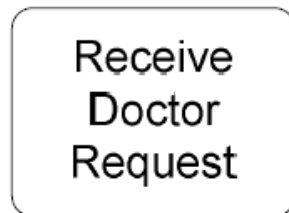
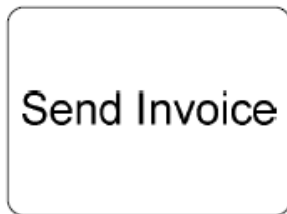
Flow Objects	Connectors	Artefacts	Swimlanes
<div>Events</div>  <div>Activities</div>  <div>Gateways</div> 	<div>Sequence Flow</div>  <div>Message Flow</div>  <div>Association</div> 	<div>Data Object</div>  <div>Name (State)</div> <div>Text Annotation</div>  <div>Group</div> 	<div>Pool</div>  <div>Lanes (within a Pool)</div> 

Activities



- An activity is work that is performed within a business process.
- Typically an activity is one step of a larger business process.
- There are two types of activities :
 - ◆ **Task** (atomic)
 - ◆ **Sub-Process** (compound, i.e. consisting of several activities)
- Activities are rounded rectangles (some tools use colors)
- Activities can be performed once or can have internally defined loops

Tasks



- A Task is an atomic activity.
- A Task is used when the work is not broken down to a finer level of detail
- There are specialized types of tasks, e.g.
 - ◆ manual
 - ◆ sending
 - ◆ receiving
- Markers or icons can be added to Tasks to help identify the type of Task
- Markers must not change the footprint of the Task
 - ◆ In the example the task **Fill Order** has a person as a marker to indicate that it is executed by a person

Attributes of Tasks

Every task has attributes, which capture details of the work

■ name:

- ♦ typically short – consisting of a verb and an object/resource
- ♦ Example: **Check Reservations**

■ description

- ♦ details about the work, what it means and how it is performed and mention the applications used
- ♦ Example:
Check the reservation book to see whether the reservation exists. Verify that the party arrived before the reservation time

■ participant/resource

- ♦ A task is typically performed by a person, who is a resource. The resource is not mentioned as a single person but a role
- ♦ Example: **Waiter**

■ cost

- ♦ details about costs of resources can be helpful for process analysis

The actual attributes available depend on the modeling tool

(Bridgeland & Zahavi 2009, p. 107f)

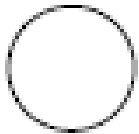
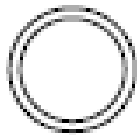
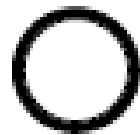


Sequence Flow

- A Sequence Flow is used to show the order that activities will be performed in a Process
- The source and target must be one of the following objects
 - ◆ Events
 - ◆ Activities
 - ◆ Gateways
- In a sequence of activities, the subsequent activity is performed after the previous activities is finished



Events

Start**Intermediate****End**

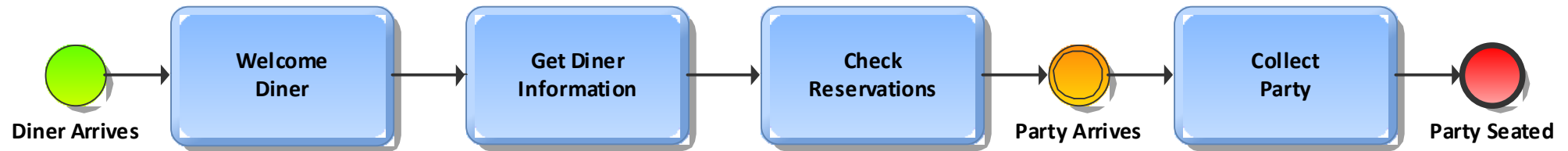
- A process begins with a start event and ends with an end event
- Events are states that affect the flow of the process
 - ◆ they start, interrupt and finish the flow
 - ◆ they can trigger an activity or are its result
- Events are represented as circles. The type of boundary determines the type of Event
 - ◆ Start Event
 - ◆ Intermediate Event
 - ◆ End-Event
- Events can have descriptions, just as tasks.

Example: A simple End-to-End Process



- **Diner Arrives** is the start event
- **Diner Seated** is the end event
- Note that the names of the events are different from the names of the activities
 - ◆ Activity names are typically imperative sentences, they sound like command. The verb is at the beginning of the name.
 - ◆ Event names are typically declarative sentences, describing a state or something that happens

Example with an Intermediate Event

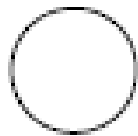


- An intermediate event happens after the process starts and before it ends
- In this example the events models a delay: But when the first diner of a party arrives the host checks the reservations but does not seat the diner until the rest of the party arrives.

Event Types

For start, intermediate and end events there are specific types:

Start



None



Timer



Message



Signal



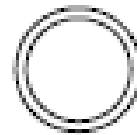
Conditional



Multiple



Intermediate



None



Message



Timer



Error



Compensation



Rule



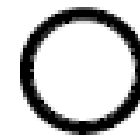
Link



Multiple



End



None



Message



Error



Compensation



Link



Terminate



Multiple



Start Events

Core Start Events

None



Timer



Message



Signal



Advanced Start Events

Conditional


















Multiple



- Start Events indicate when a Process will begin
- There are different “Triggers” that indicate the specific circumstances that start the process
 - None** – no trigger defined
 - Timer** – specific date and time of regular date-time cycle (e.g. first Friday of the month at 8am)
 - Message** - Arrival of a message from another business entity or role
 - Signal** – signal broadcast from another process (e.g. change of a prevailing Interest Rate triggering any number of other processes)
 - Conditional** – condition expression that must be satisfied for the process to start
 - Multiple** – two or more triggers that can be any of the other triggers (OR combined)
- Start events can only have outgoing sequence flows

Intermediate Events

Core Intermediate Events

	Catching	Throwing
None		
Timer		
Message		
Signal		
Error		
Cancel		
Compensation		
Conditional		
Link		
Multiple		

- Intermediate Events occur after a process has been started and before a process is ended
- There are different “Triggers” that indicate the specific circumstances of the Event
- Each type of event can either catch or throw an event

Intermediate Events

■ Core Intermediate Events

None – no trigger defined

Timer – specific date and time of regular date-time cycle

Message - Arrival of a message from another business entity (participant) in the process. If shown in the diagram the entities are represented as Pools.

Signal – signal that is broadcast or received.

■ Advanced Intermediate Events

Error – Event that usually disrupts the Process or requires correction

Cancel – used to cancel a TransactionSub-Process

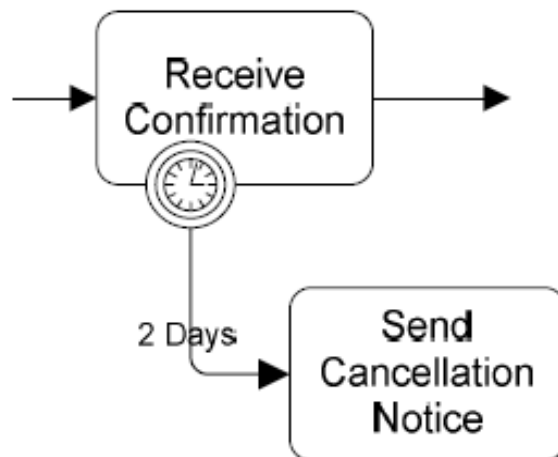
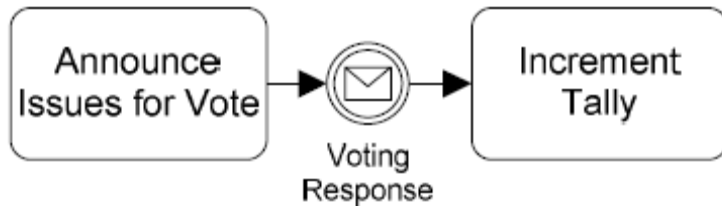
Compensation – establish the behaviour to undo Activities

Conditional – condition expression that must be satisfied for the process to start

Link – visualize a "goto" mechanism

Multiple – two or more triggers that can be any of the other triggers (OR combined)

Intermediate Events



- Intermediate Events can be placed in the normal flow of the Process or attached to the boundary of an activity
- Events that are placed within the process flow represent things that happen during the normal operations of the process
 - ◆ They can represent the response to the Event (i.e., the receipt of a message)
 - ◆ They can represent the creation of the Event (i.e., the sending of a message)
- Events that are attached to the boundary of an activity indicate that the activity should be interrupted when the Event is triggered
 - ◆ they can be attached to either Tasks or Sub-Processes
 - ◆ They are used for error handling, exception handling, and compensation
 - ◆ Events that can interrupt an Activity are Timer, Message, Error, Cancel, Conditional, Signal

End Events

Core End Events

None



Message



Signal



Terminate



Advanced End Events

Error



Cancel



Compensation



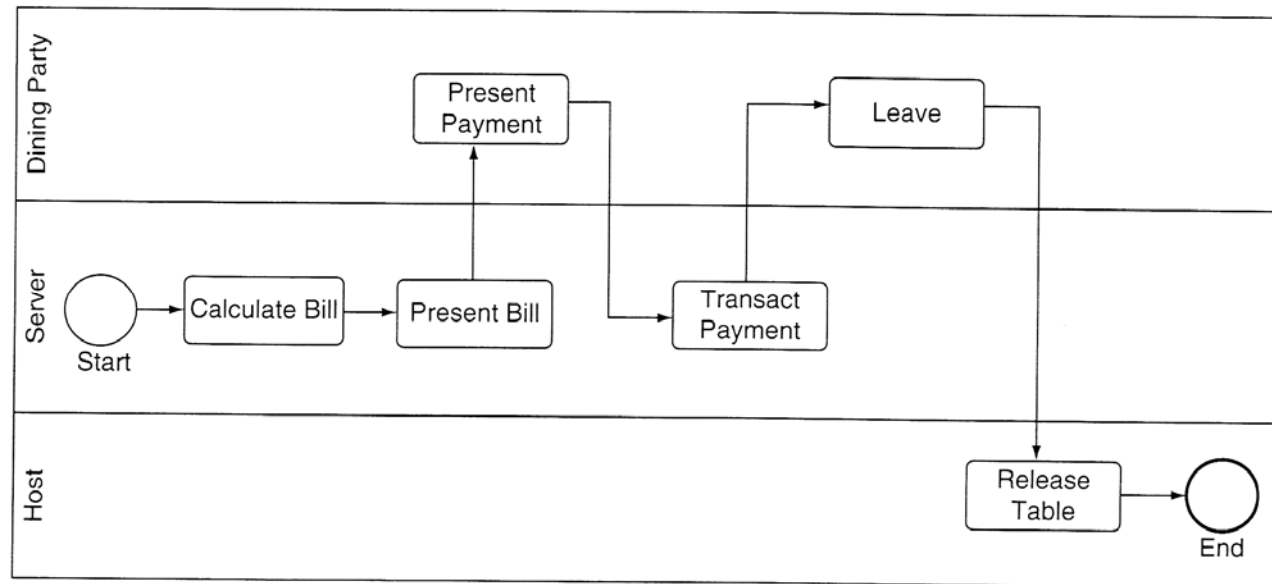
Multiple



- End Events indicates where a process will end
- There are different “Results” that indicate the specific circumstances that end the Process
- Basis End Events
 - None** – no result is defined
 - Message** – Communication to another business Entity (participant of Process)
 - Signal** – Defines a broadcast event that any other Process can see and react to
 - Terminate** – Stops all Process Activities, even if they are ongoing in parallel paths
- Advanced End Events
 - Error** – disrupt the Process or require correction
 - Cancel** – causes cancellation of the Transaction Sub-Process. It is a Throw for the Catch
 - Compensation** – Also used as part of the Transaction Sub-Process behaviour
 - Multiple** – Defines two or more of Message, Error, Compensation or Signal results.

Lanes

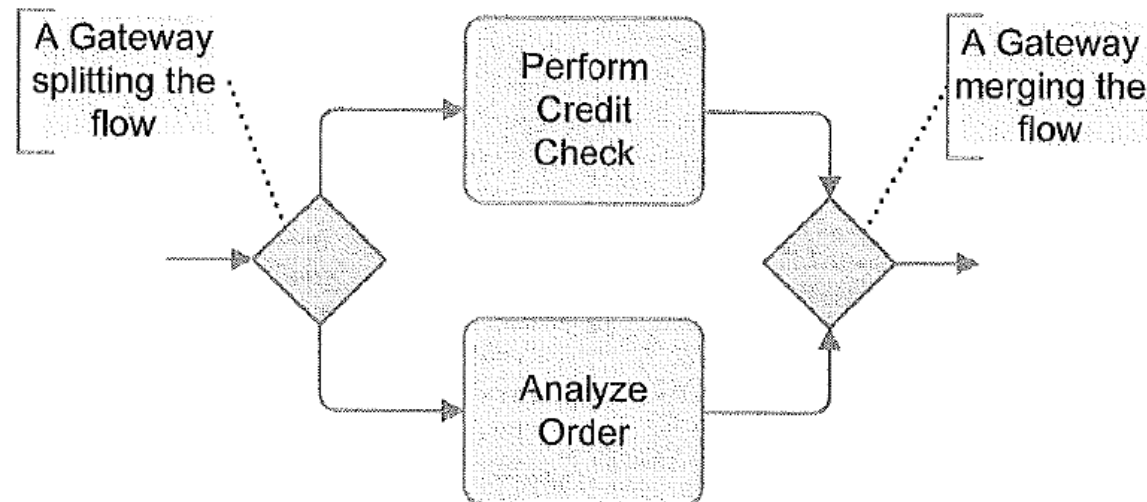
- A business process model graphically shows who performs which activity
- Each role that performs activities in a process has a lane – a horizontal stripe (like a line in a swimming pool)
- Each lane has the name of the role or organisation who performs the work
- Sequence flows can cross boundaries of lanes



(Bridgeland & Zahavi 2009, p. 110f)

Gateways

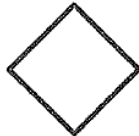
- **Gateways** model sequence flow alternatives, i.e. they represent points of control
- They split and merge the flow of a Process
- All types of Gateways are diamonds
- The underlying idea is that Gateways are unnecessary if the Sequence Flow does not require controlling



Types of Gateways

Core Gateways

Exclusive



Event



Parallel



Advanced Gateways

Inclusive



Complex



Exclusive

Splitting: only one outgoing paths can be taken when the process is performed, depending on the evaluation of Sequence Flow conditions (decision)

Merging: Control flow of any incoming path is passed through

Event

Splitting: the path that is taken depends on the occurrence of a specified event

Merging: Flow of any incoming path is passed through

Parallel

Splitting: all outgoing paths are taken.

Merging: the Gateway waits for all incoming paths (synchronization)

Inclusive

Splitting: one to all of the outgoing paths are taken depending on the evaluation of all Sequence Flow conditions.

Merging: Flow of incoming path is passed through

Complex

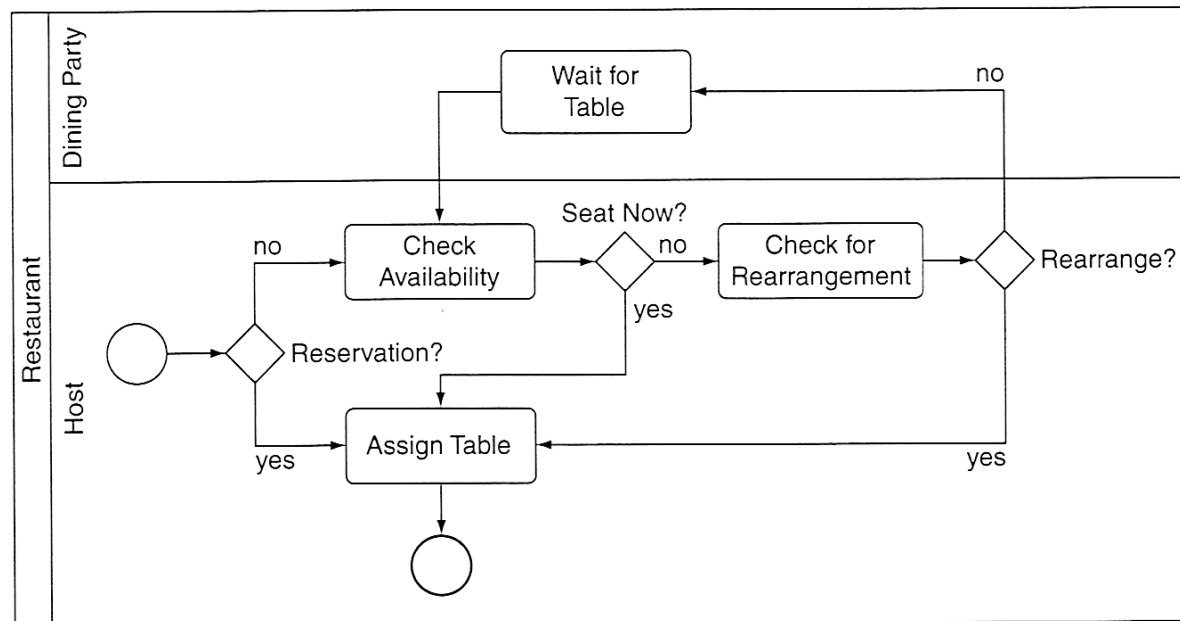
Splitting: one to all of the outgoing paths are taken depending on the evaluation of a single Gateway condition

Merging: Flow of incoming path is passed through depending on evaluation of a single Gateway condition

Complex Gateways provide a way for the modeler to specify (program) any behaviour

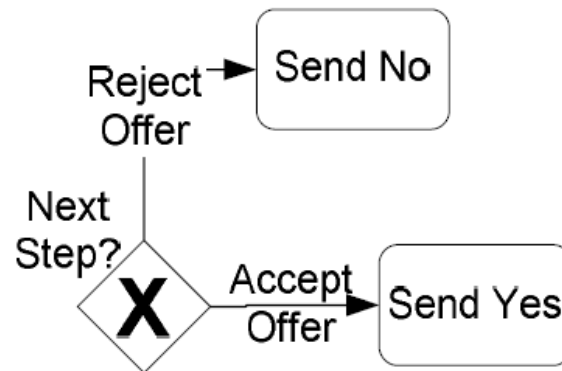
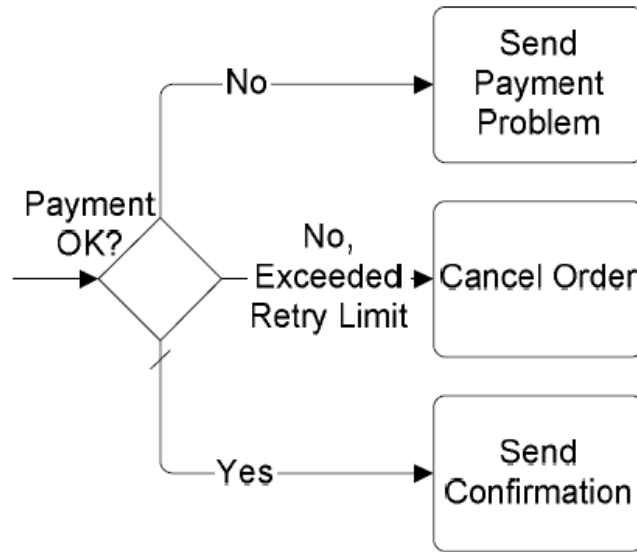
Exclusive Gateways

- For exclusive Gateways exactly one of the following sequence flows is selected
- The name of the gateway is a question with the alternative answers to the questions as labels on the outgoing sequence flows



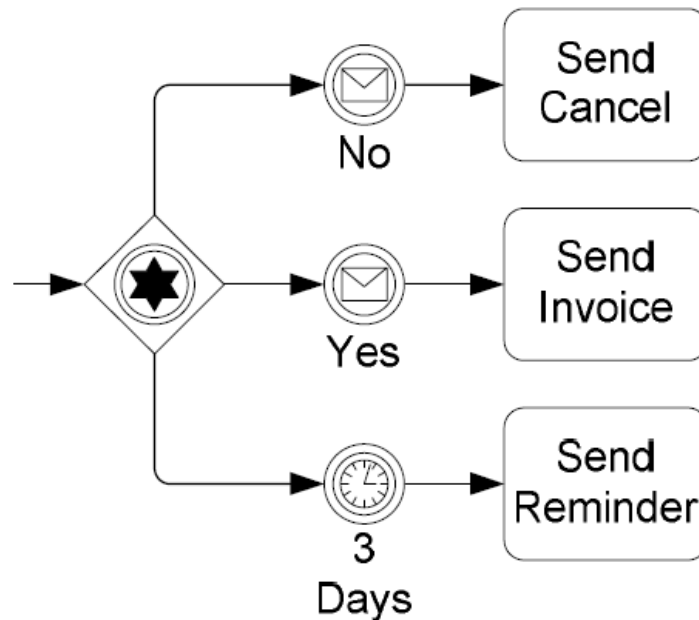
(Bridgeland & Zahavi 2009, p. 113f)

Exclusive Gateways based on Data



- The Gateway (Decision) creates alternative paths based on defined conditions.
- Exclusive Gateways based on Data are the most commonly used Gateways
- They can be shown with or without an internal „X“ marker. Without is the most common use.

Exclusive Gateways based on Events

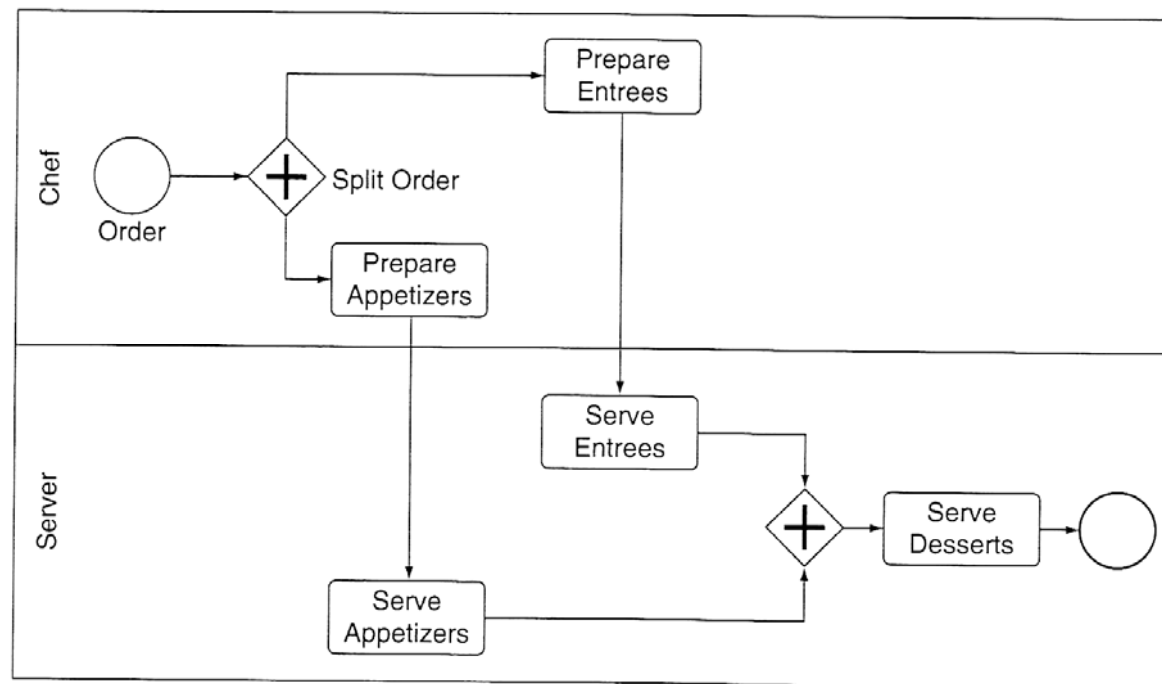


- Alternatives in this Decision are based on events that occur at the point in the process rather than conditions
- The Multiple Intermediate Event is used to identify this Gateway
- The Events that follow the Gateway Diamond determine the chosen path
 - ◆ The first Event triggered wins

Parallel Gateway

■ A parallel gateway

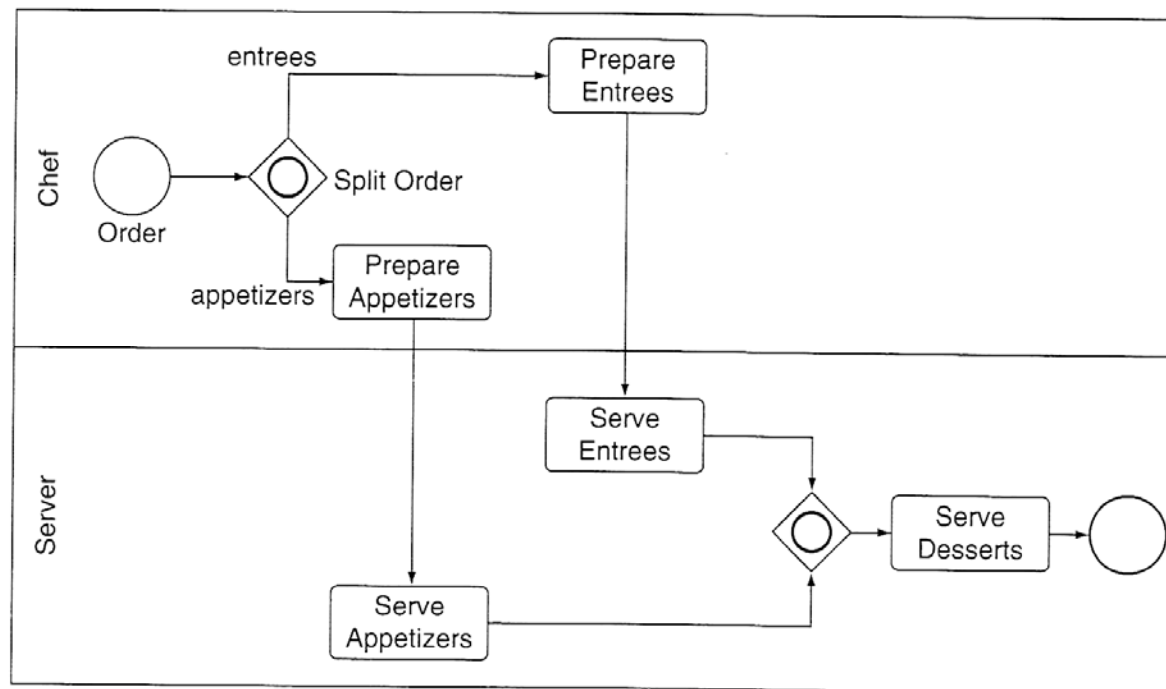
- ◆ starts parallel work, i.e. two (or more) sequence flows that then progress at the same time
- ◆ parallel flows can be joined back together by another parallel gateway



(Bridgeland & Zahavi 2009, p. 114f)

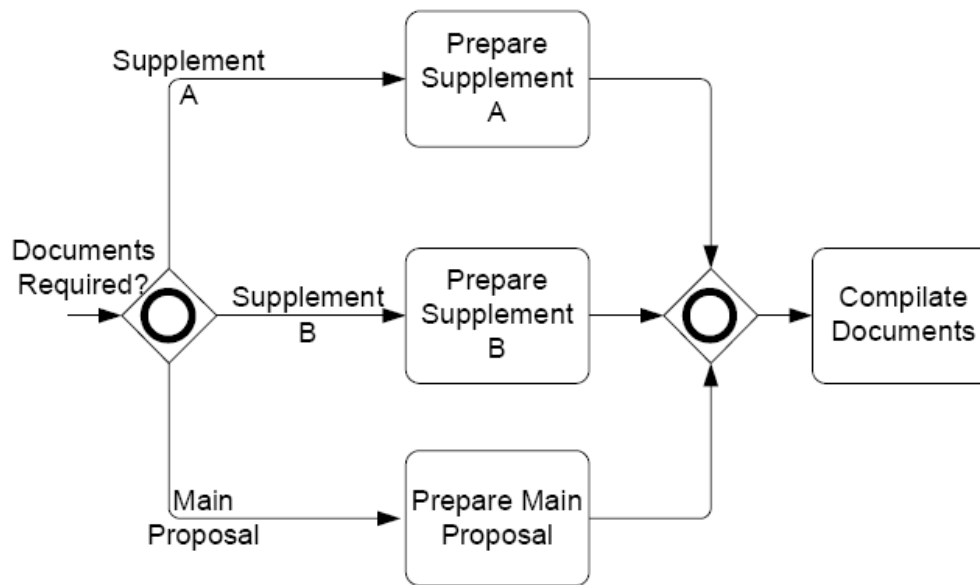
Inclusive Gateway

- An inclusive gateway allows either of the outgoing sequence flow to be taken or several in parallel.
- Example: The following process shows a process where the guests do not have both appetizers and entrees but can have only one of them.



(Bridgeland & Zahavi 2009, p. 114f)

Inclusive Gateways

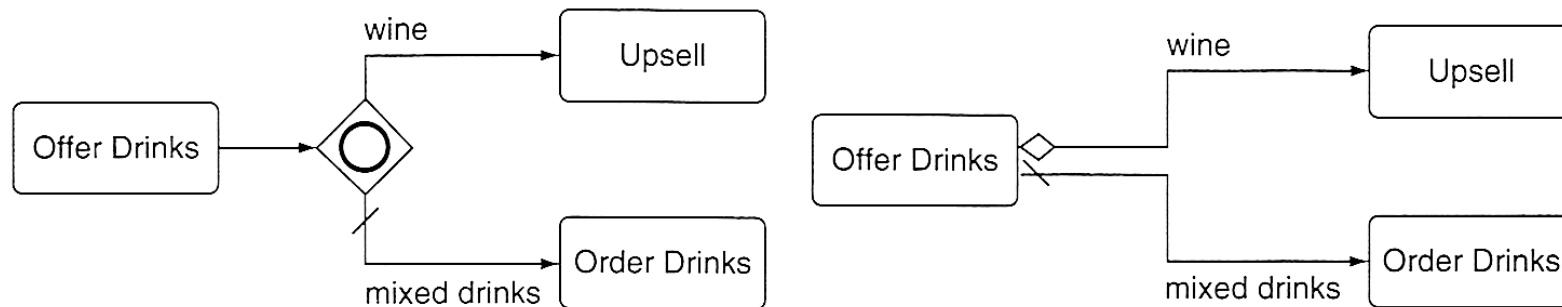


- Inclusive Gateways are Decisions where there is more than one possible outcome
- The „O“ marker is used to identify this Gateway
- They usually are followed by a corresponding merging Inclusive Gateway

Default Sequence Flow and Conditional Sequence Flow

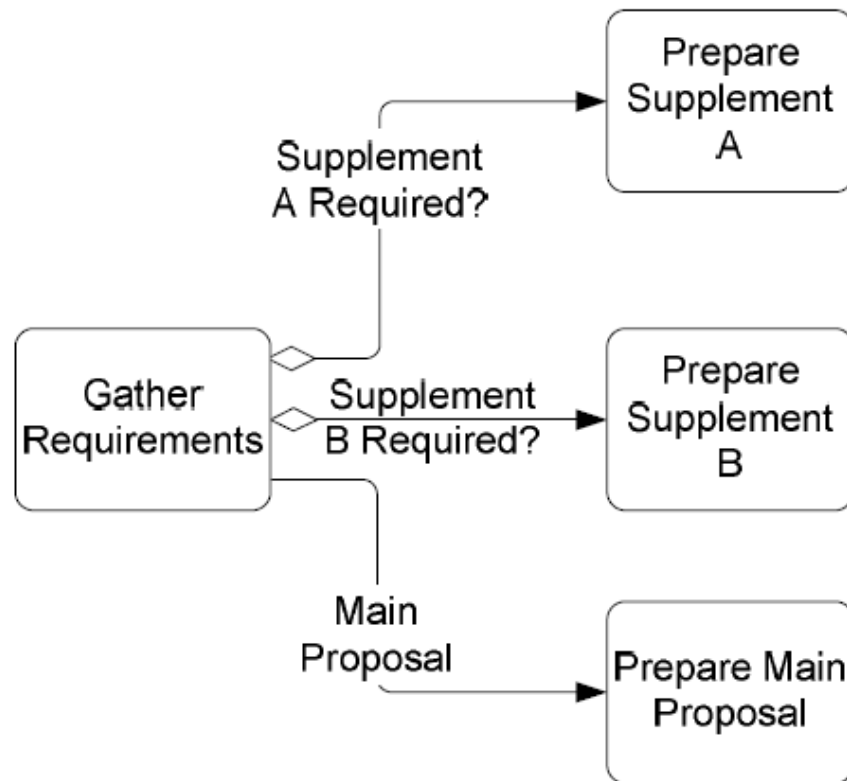
- One of the outgoing sequence flows from a gateway can be marked as default – the one that is taken if there is no reason to take another sequence flow.
- The default is modeled with a short line crossing the sequence flow.
- The same can be modeled without a gateway using a conditional sequence flow.
- A conditional sequence flow is a sequence flow that includes a condition

Example: Identical process with a gateway and with conditional sequence flow



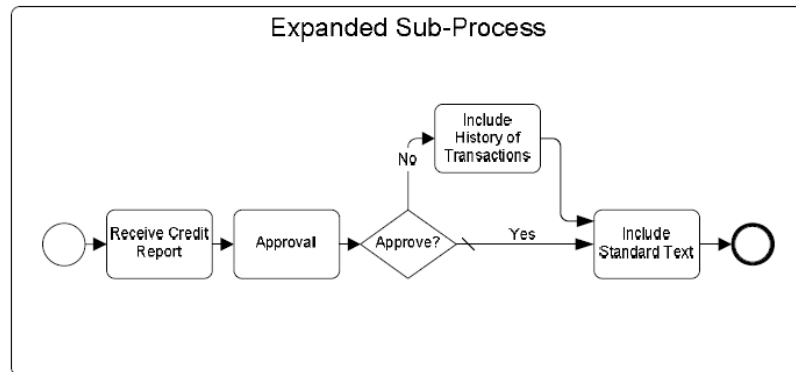
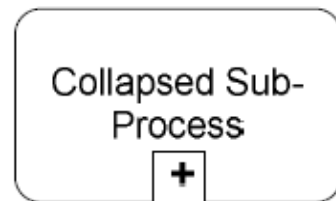
(Bridgeland & Zahavi 2009, p. 116)

Conditional Sequence Flow



- The condition of a sequence flow has to be true to allow the flow to continue down the Sequence Flow
 - ◆ A mind-diamond shows that the Sequence Flow has a condition
- At least one of the outgoing Sequence Flows must be chosen during Process performance
- Sequence flows without condition are followed in any case
 - ◆ In the example main proposal is prepared
- Defaults can be used to select a flow if no other condition is true

Sub-Processes



- A Sub-Process is a compound activity that is included within a Process.
 - ◆ A process can be broken down into a finer level of detail through a set of sub-activities
- Two kinds of representation
 - ◆ Collapsed: the details of the Sub-Process are not visible in the Diagram. A “plus” sign in the lower-center of the shape indicates that the activity is a Sub-Process and has a lower-level of detail.
 - ◆ Expanded: the details (a Process) are visible within its boundary.

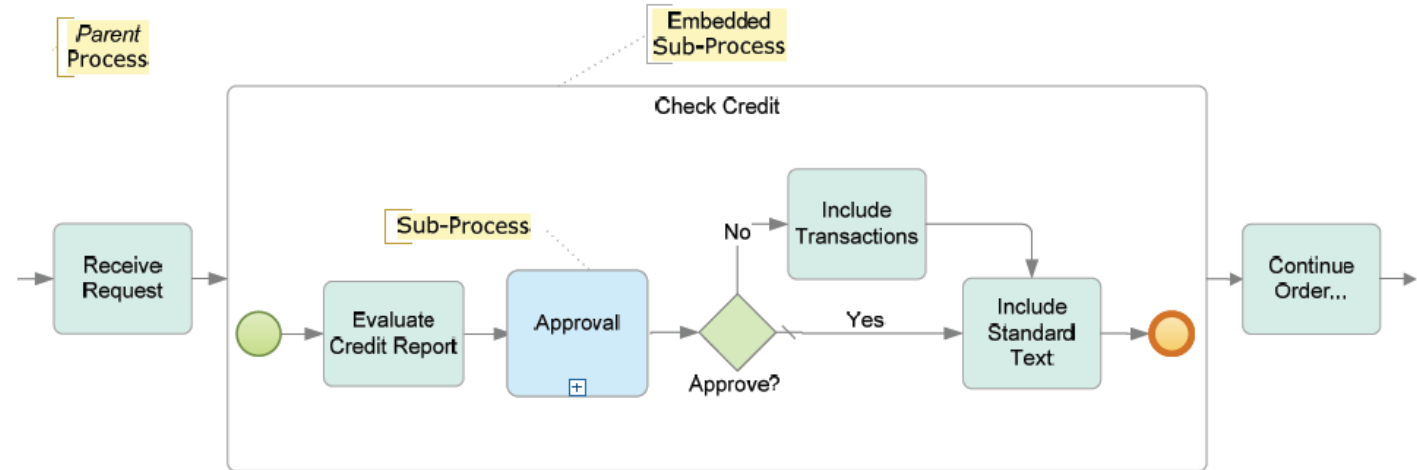
Embedded and Independent Sub-Processes

- Embedded - A modeled Process that is actually part of the *parent* Process.
 - ◆ *Embedded* Sub-Processes are not re-usable by other processes.
 - ◆ All "process relevant data" used in the *parent* Process is directly accessible by the *embedded* Sub- Process (since it is part of the *parent*).
- Independent¹⁾ - A separately modeled Process that could be used in multiple contexts.
 - ◆ Example: checking the credit of a customer
 - ◆ Any data must be transferred specifically between the *parent* and Sub-Process.
 - ◆ An independent Sub-Process can also be called Top-level process.

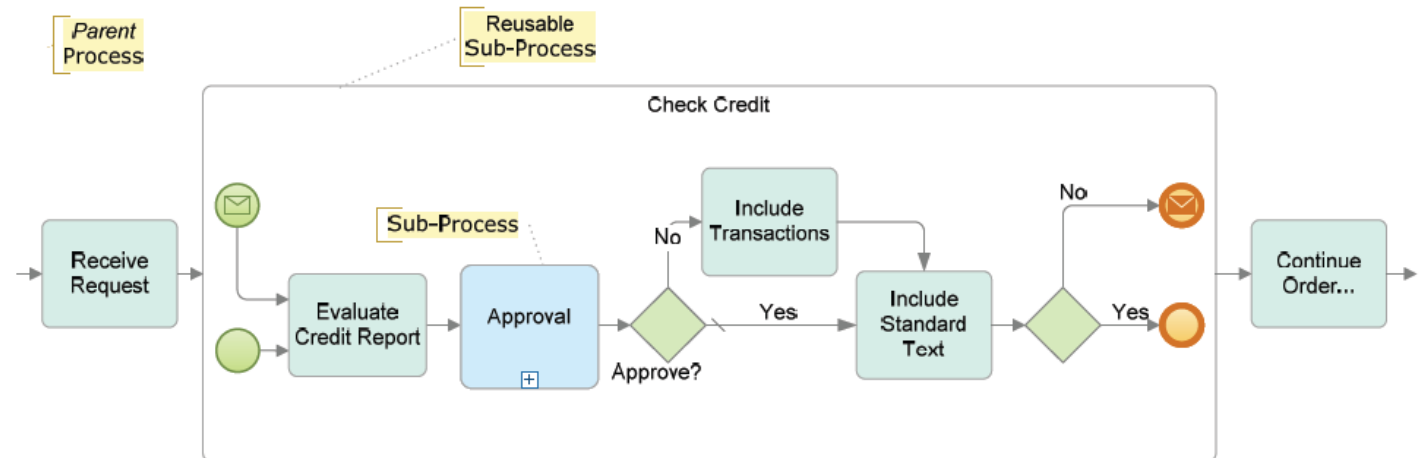
¹⁾ Independent Sub-Processes where called Reusable in BPMN 1.0

Examples of Embedded and Independent Sub-Processes

Embedded Sub-Process (expanded)



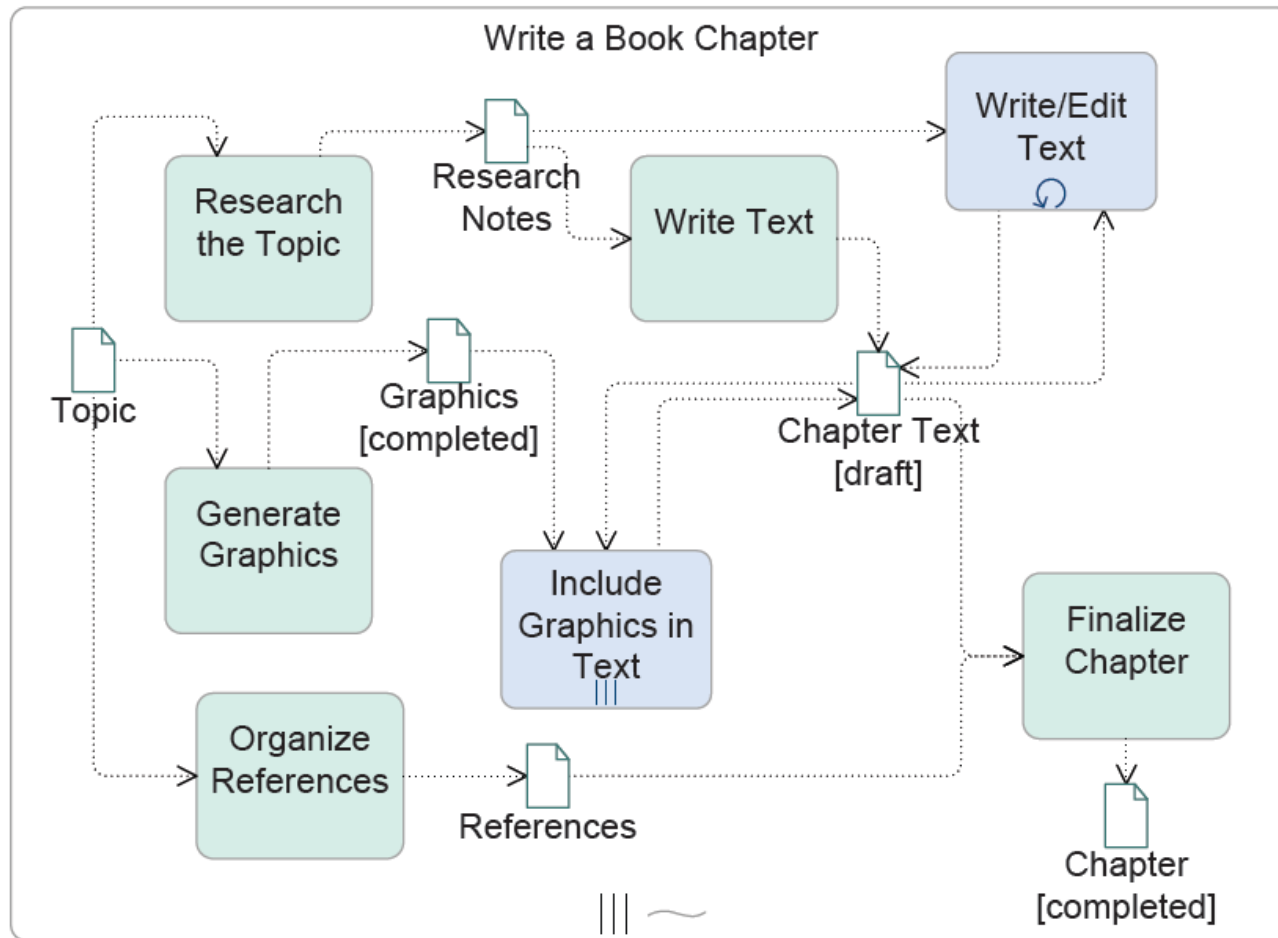
Independent Sub-Process (expanded)



Ad hoc Processes

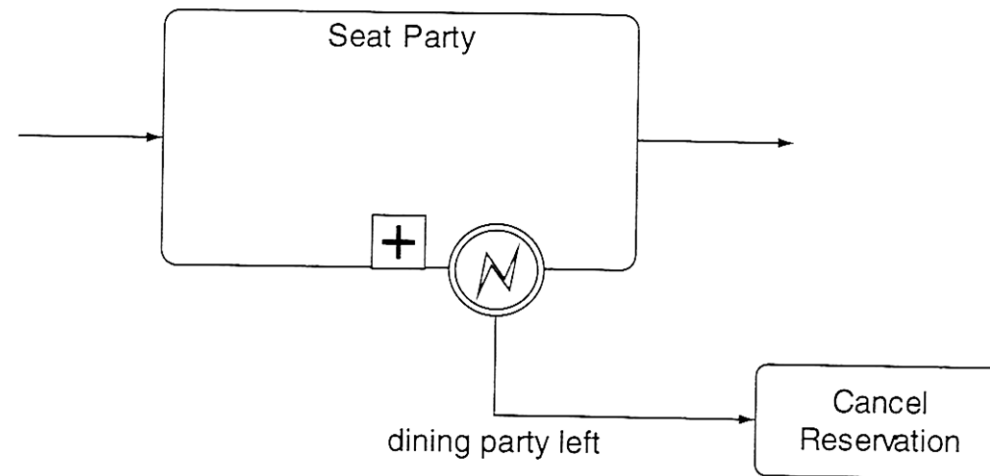
- The Ad hoc process represents processes where the activities can occur
 - ◆ in any order
 - ◆ In any frequency
- There is no specific ordering or obvious decisions
- It has a tilde (~) to show that it is ad hoc
- Typically, the activities in an ad hoc process involve human performers to make decisions as to what activities to perform, at which time and how many time
- It is possible, however, to use occasional sequence flow between some activities, but sequence flow does not imply that there are explicit start and end events.
- The ad hoc process has a non-graphical completion condition attributes. When the attribute becomes true (by updating the date expressed in the condition), the process terminates.

Example of an Ad hoc Process



Exception Flow

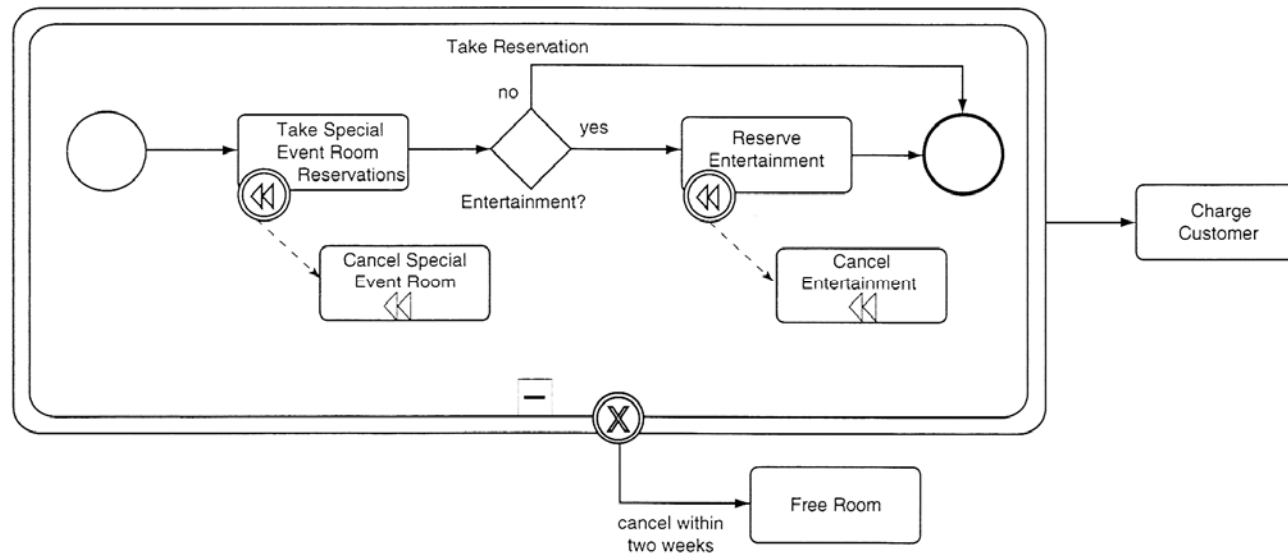
- Exception Flow is a sequence flow triggered by an exception
 - ◆ an intermediate event that occurs some time during the course of the process
 - ◆ modeled as an event on the border or the subprocesses
- Example: If the dining party leaves at any part of the subprocess their reservation is cancelled.



Cancellation and Compensation

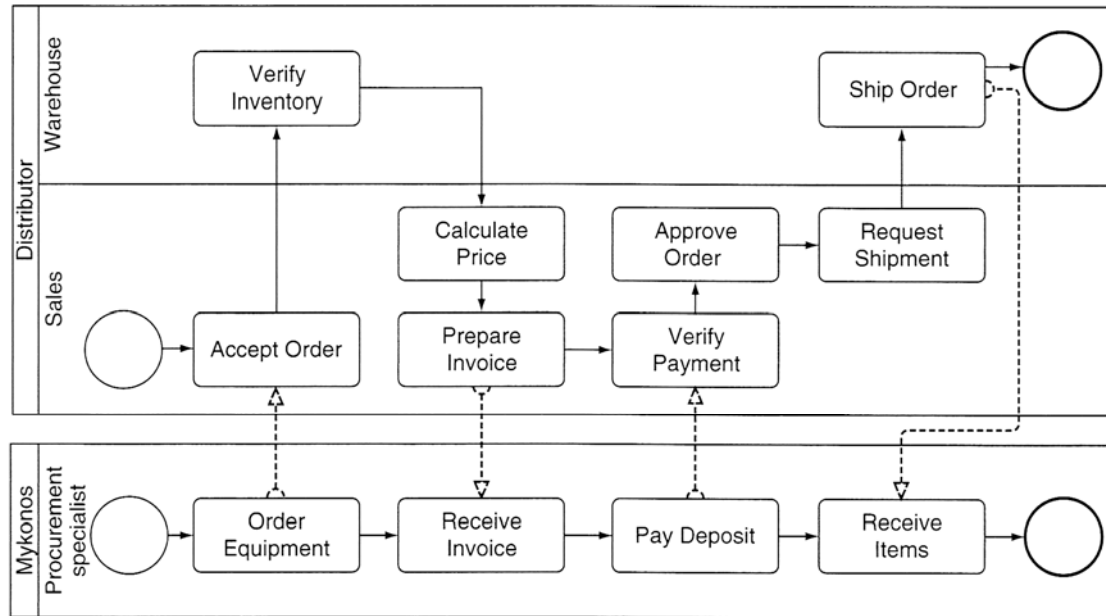
- A business transaction is a collection of activities that must either complete successfully or must be rolled back in their entirety.
- The following process shows a Business Transaction modelled with compensation and cancel events:

If the reservation is cancelled (cancel event on the border of the process) then both the room reservation and entertainment have to be cancelled. This is modeled with the compensation events.



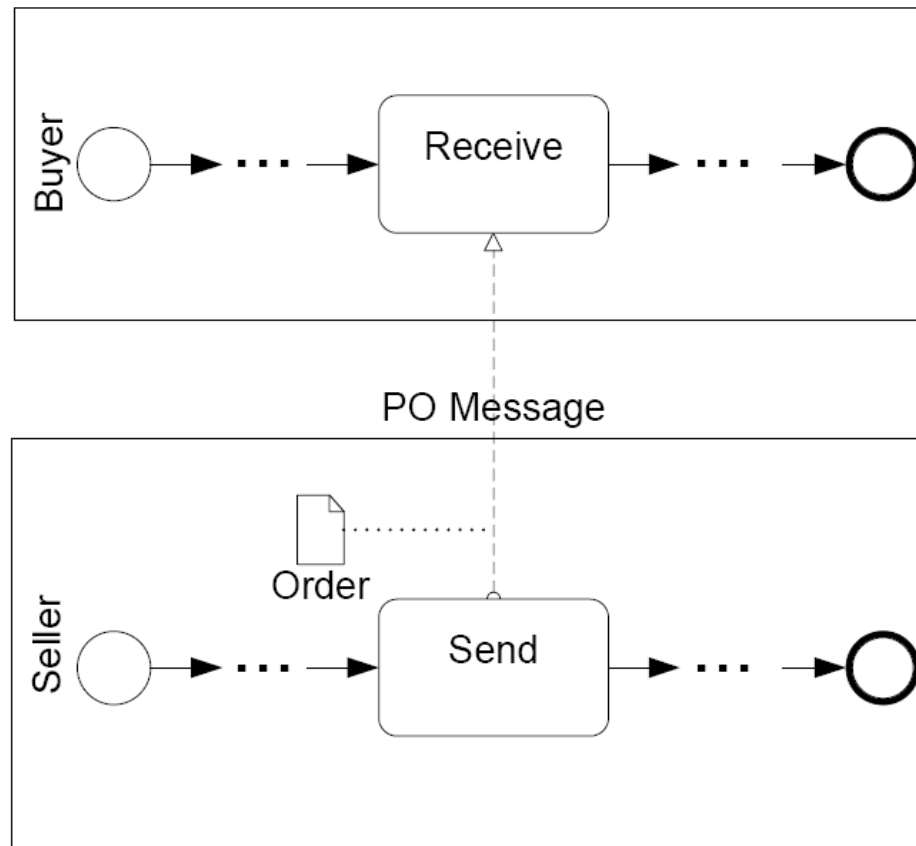
Swimlanes

- Swimlanes partition and organise activities
- There are two main types of swimlanes: Pool and Lane
 - ◆ Pools represent Participants in an interactive (B2B) Business Process Diagram
 - ◆ Lanes represent sub-partitions for the objects within a Pool – they represent participants of a process



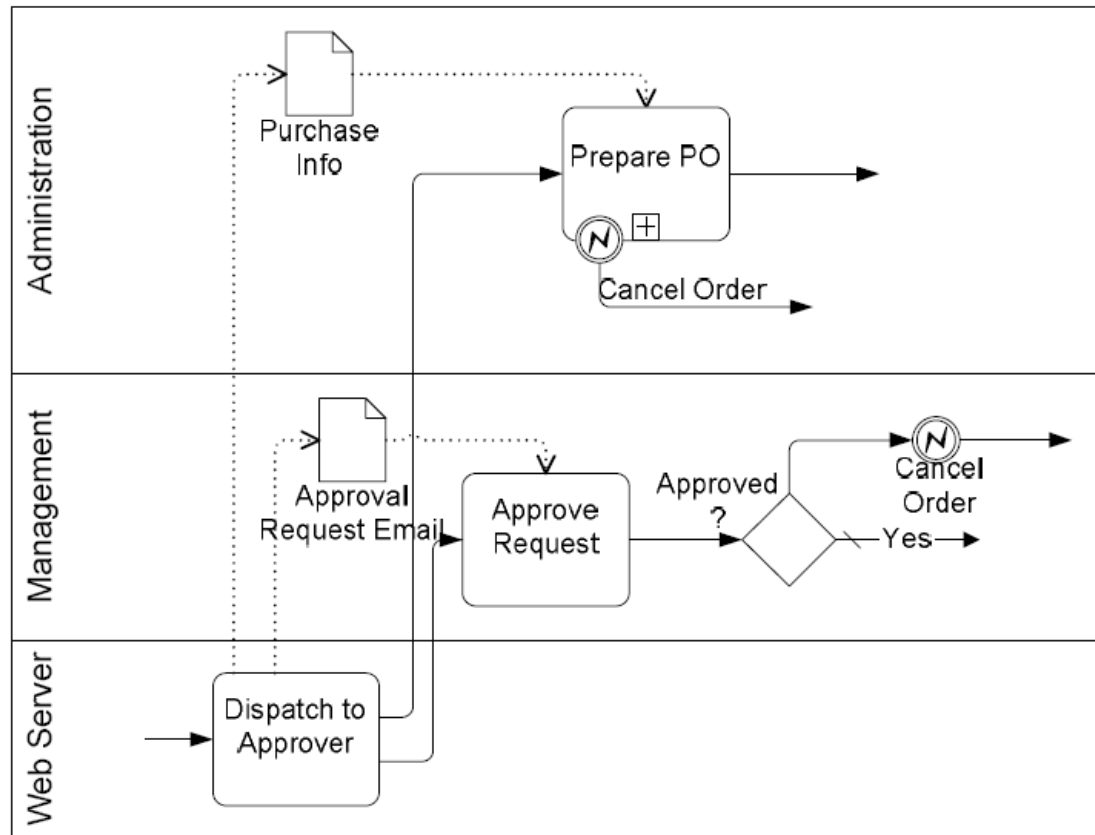
(Bridgeland & Zahavi 2009, p. 123)

Pools



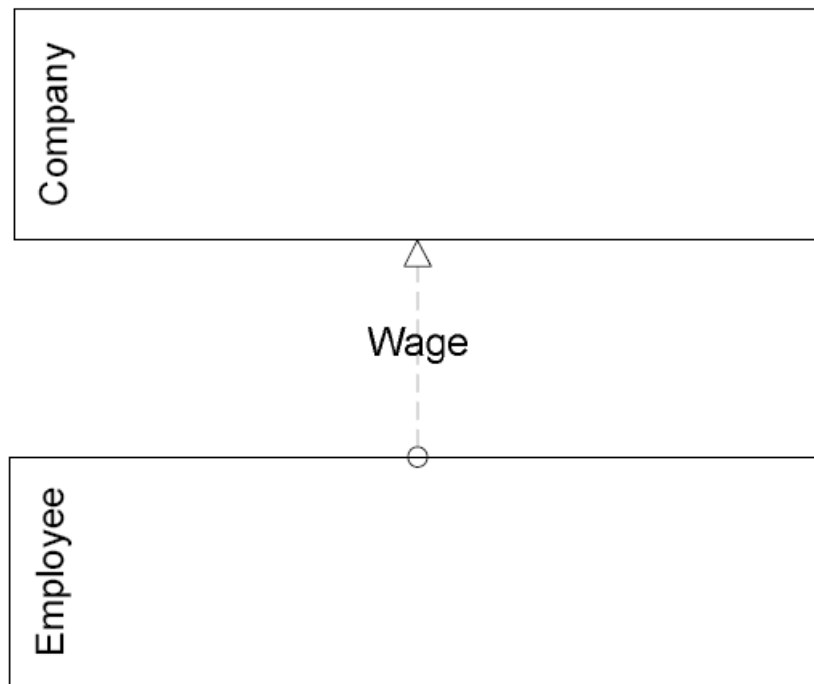
- Pools represent independent Participants in an interactive (B2B) Business Process
 - ◆ A Participant may be a business role (e.g. „buyer“ or „seller“) or a business entity (e.g. „IBM“ or „OMG“)
- A Pool may be a „black box“ or may contain a Process
- Interaction between Pools is handled through Message Flow
- Sequence Flow cannot cross the boundary of a Pool (i.e. a Process is fully contained within a Pool)

Lanes



- Lanes represent sub-partitions for the objects within a Pool
- They often represent organisation roles (e.g. manager, associate), but can represent any desired Process characteristic
- Sequence Flow can cross lane boundaries

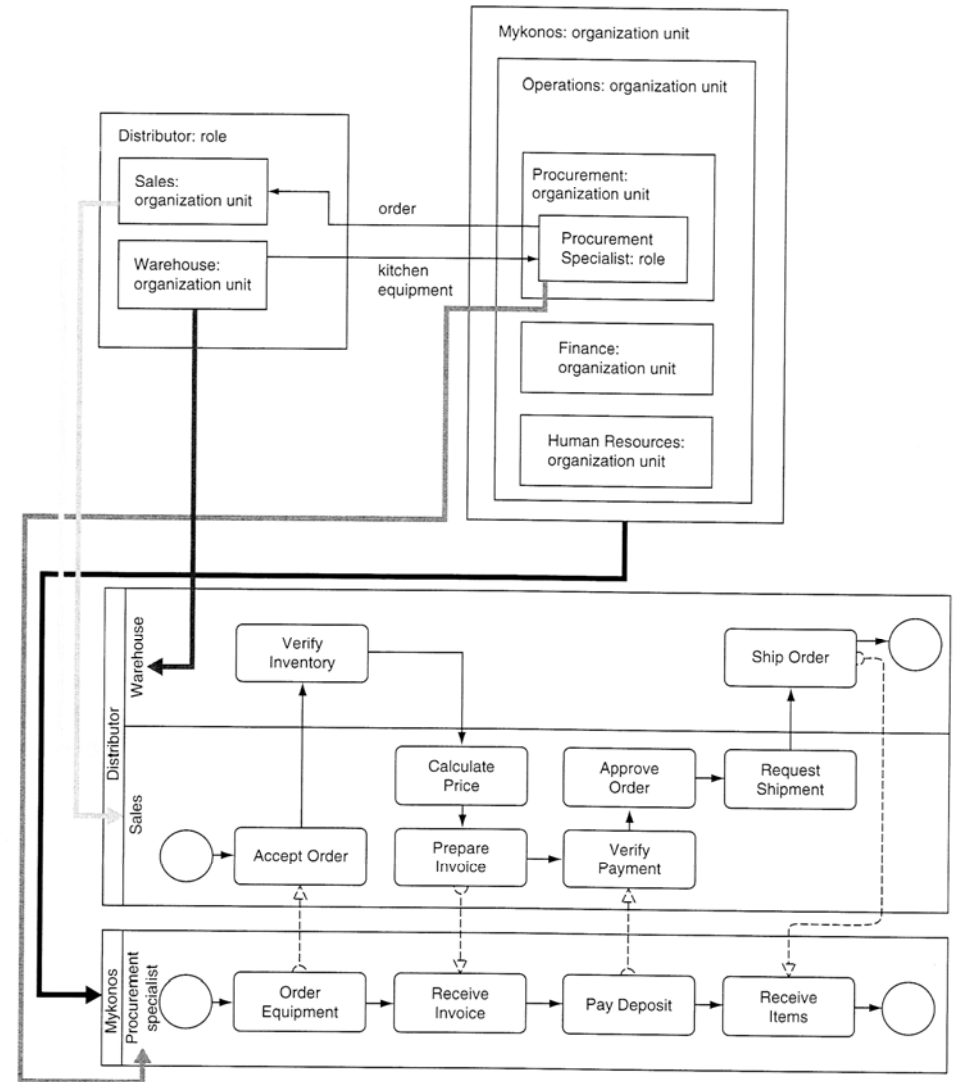
Message Flow



- A Message Flow is used to show the flow of messages between two Pools of a Process
- A Message Flow can connect to the boundary of the Pool or to an object within the Pool
- Message Flows are not allowed between objects within a single Pool

Business Processes, Organisations, and Interactions

- A pool contains a process
 - ◆ The pool is labeled with the participant who manages this process
- A lane in a process model is labeled with the participant who performs the action
 - ◆ an role or organisation in the pool
- Interactions to external roles/organisations are modeled as message flows in a process

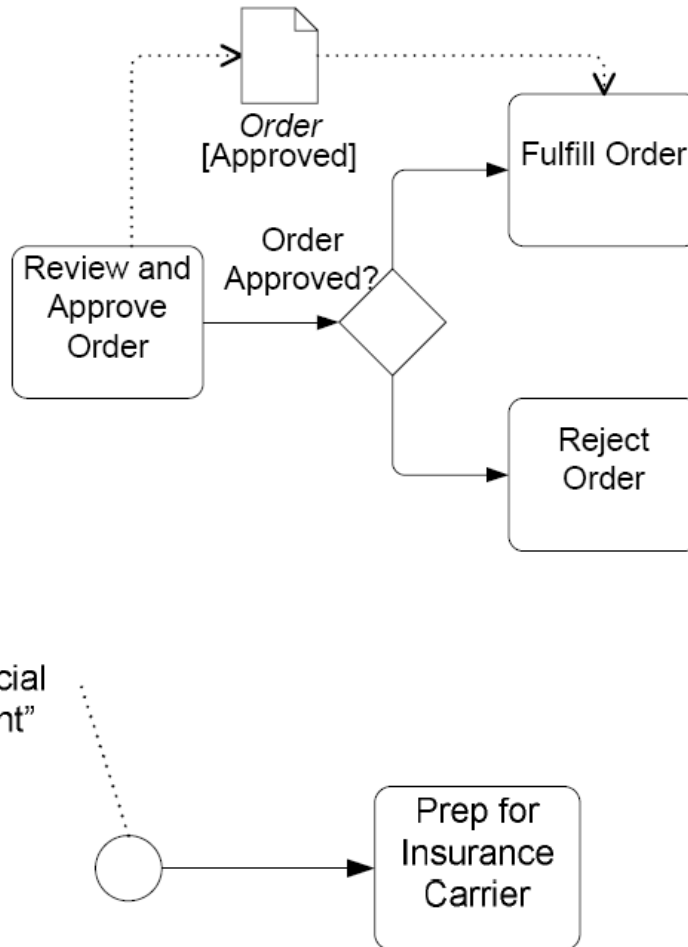


(Bridgeland & Zahavi 2009, p. 130f)

Artifacts

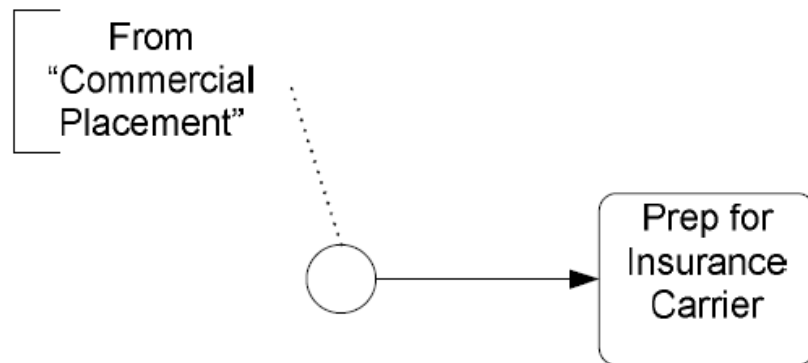
- Artifacts provide the capability to show information beyond the basic flow-chart structure of the Process
- There are currently three standard Artifacts in BPMN:
 - ◆ Data Objects
 - ◆ Groups
 - ◆ Annotations
- A modeler or tool can extend BPMN by defining new Artifacts

Associations

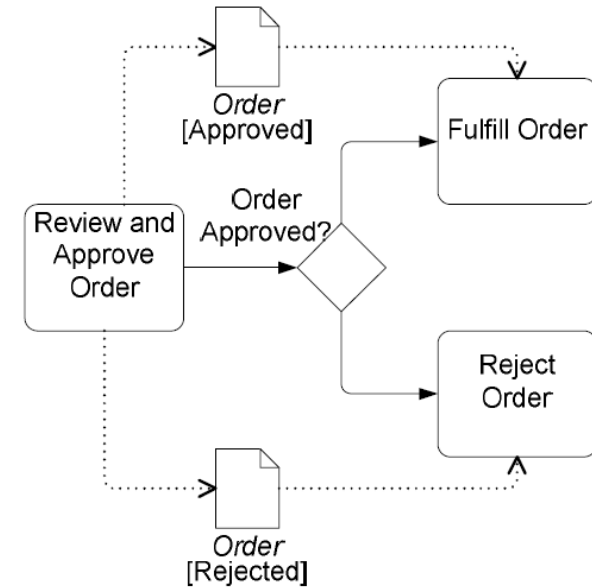


- An Association is used to associate data objects and artifacts with flow objects
- Associations are used to show how data is input to and output from Activities
- Text Annotations can be associated with objects

Text Annotations and Data Objects



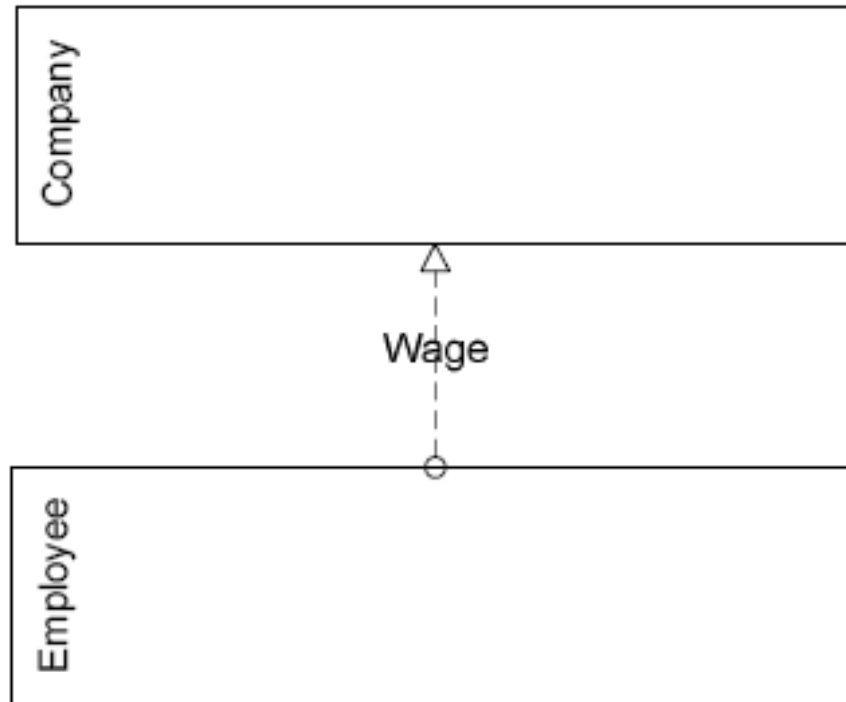
- Text Annotations are a mechanism for a modeler to provide additional information about a Process
- Text Annotations can be connected to a specific object on the Diagram with an Association



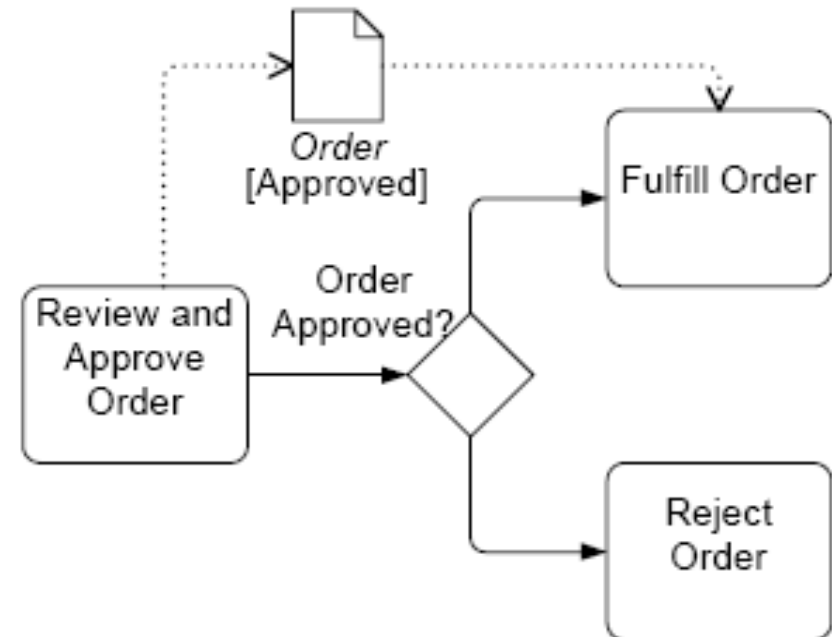
- Data Objects can be used to define inputs and outputs of activities
- Data Objects can be given a "state" that shows how a document may be changed or updated within the Process

Data Transfer with Message Flow und Associations

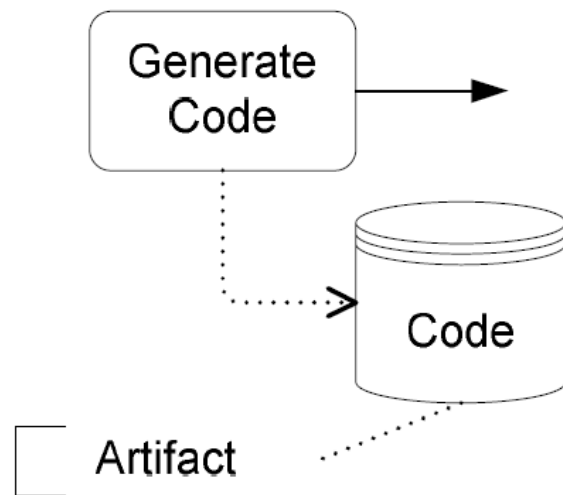
Message Flow between pools:



Associations within a pool:



Artifacts are Extendible

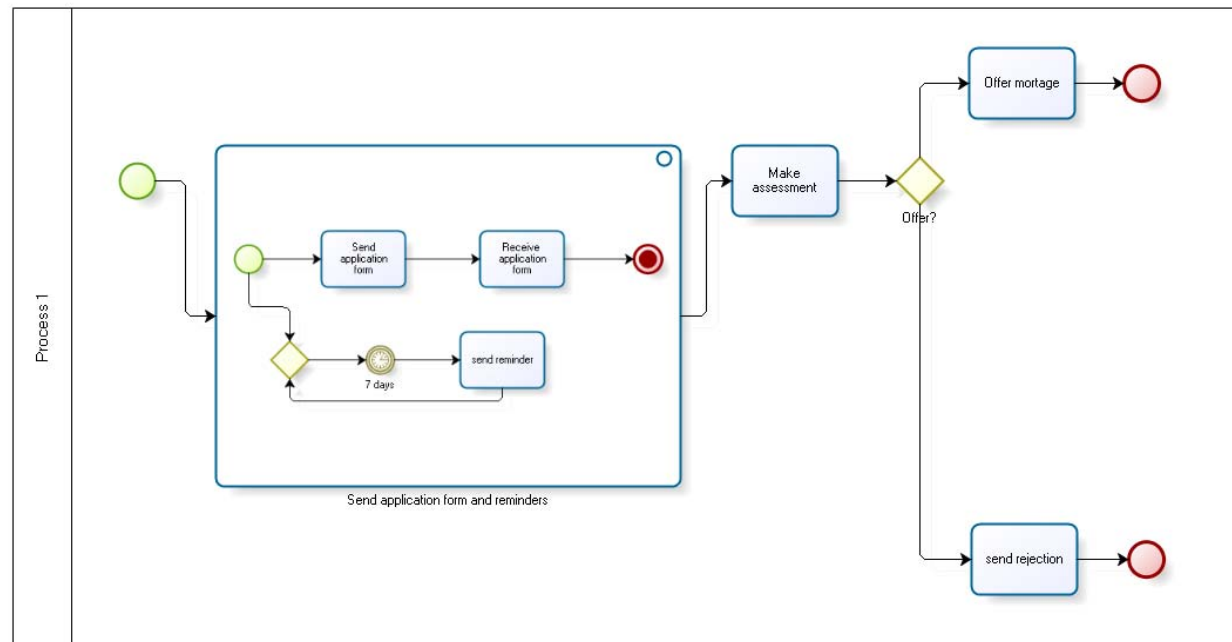
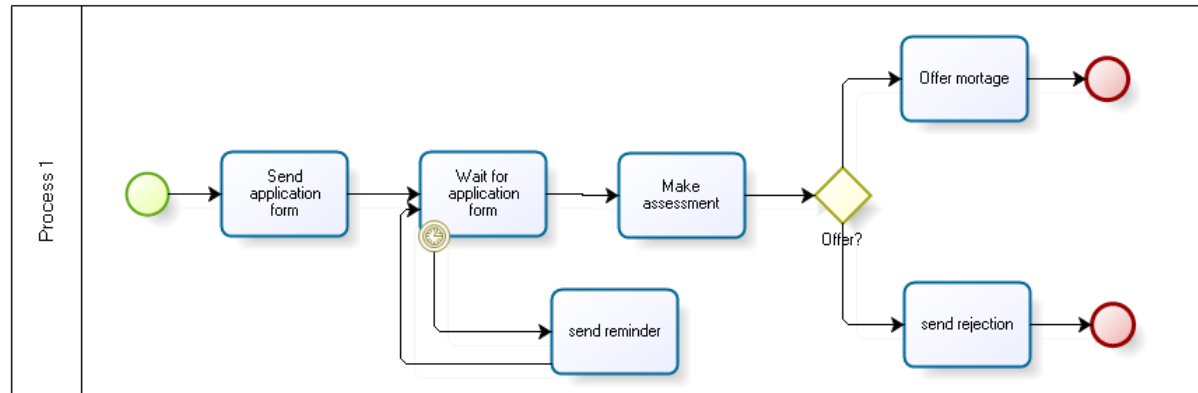


- Modelers and Modeling tools can add new artifacts to a diagram
 - ◆ Specific industries or markets may have their own set of artifacts
- Their shapes must not conflict with existing shapes
- They are not part of normal flow, but can be associated with other elements

Exercise

- What are the possible uses of process models?
- From this derive: What would you want to represent in a process model?
- → Brainstorming

Exercise – Timer



The following models represent a loop with a timer event.

- ◆ Are the two representations equivalent?
- ◆ Why do we need a subprocess in the second solution?
- ◆ Why do we need a termination end event in the subprocess?