University of Applied Sciences Northwestern Switzerland
School of Business

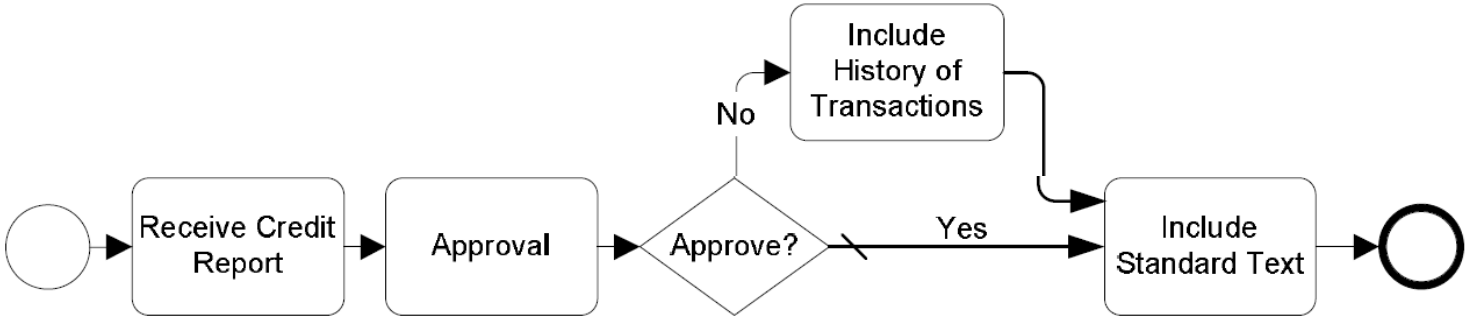
BPMN –
Business Process Modelling Notation

Knut Hinkelmann

This chapter is based on the BPMN Tutorial of Stephen A. White and the book
White, S.A., Miers, D. (2008) BPMN - Modeling and Reference Guide. Future Strategies Inc.

What is BPMN?

- BPMN is flow-chart based notation for defining Business Processes



- BPMN is an agreement between multiple modeling tools vendors, who had their own notations, to use a single notation for the benefit of end-user understand and training
- BPMN provides a mechanism to generate an executable Business Process (BPEL) from the business level notation
  - A Business Process developed by a business analyst can be directly applied to a BPM engine instead of going through *human* interpretations and translations into other languages

History of BPMN

- In 2001 BPMI.org began developing BPML (Business Process Modelling Language), an XML-based process execution language
- BPMN published by BPMI in early 2005
  - Graphical representation: Single modelling notation
  - Exports BPEL (Business Process Execution Language) to drive process automation engines
- BPMI merged with OMG in June 2005
  - First OMG revision (1.1) published January 2008
- Over 50 tools support BPMN
- BPMN 2.0 submission selection under way
  - Aims to preserve BPMN syntax, introduce formal metamodel
  - Next review stage in December 2008

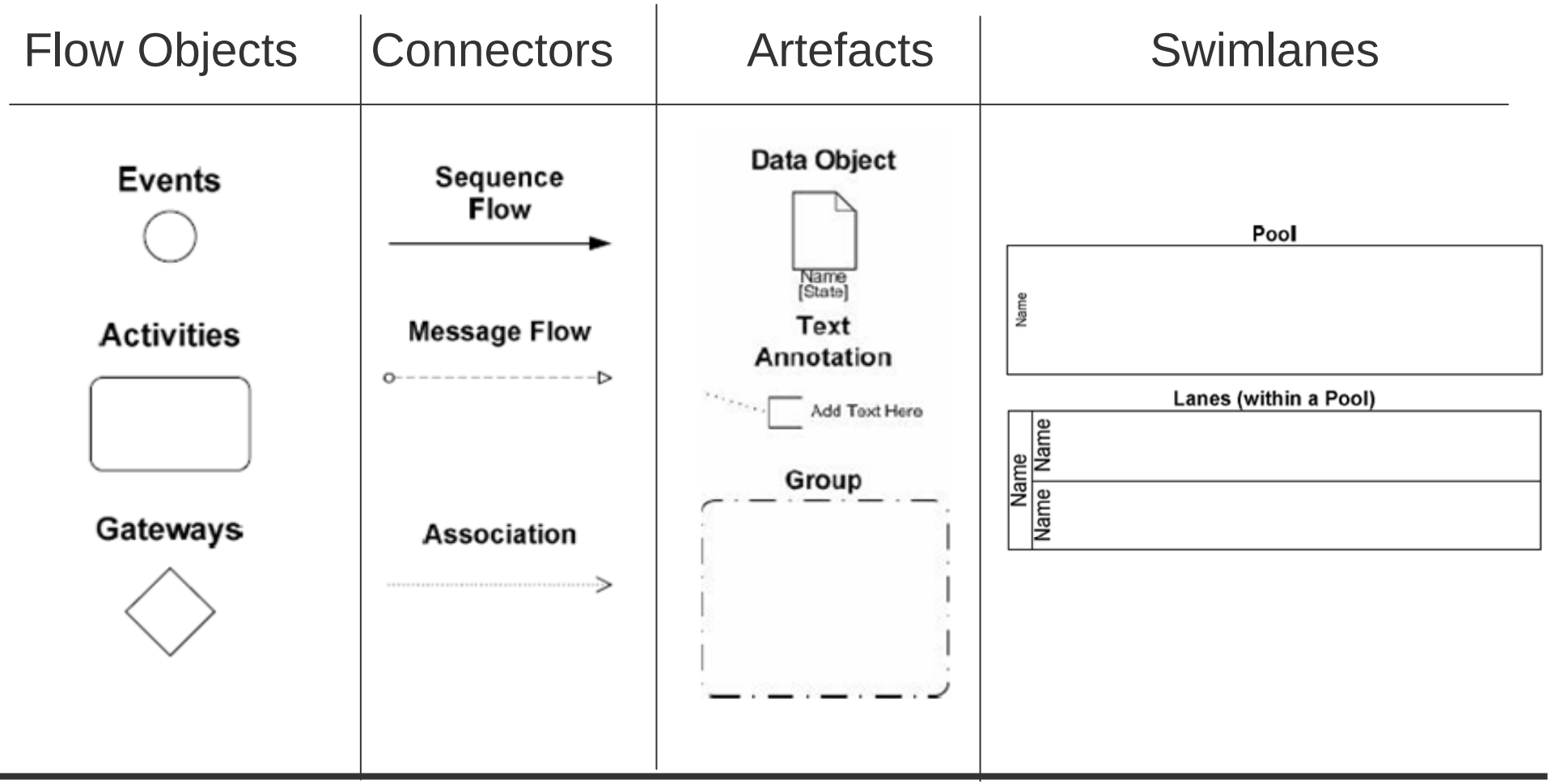
Objectives of BPMN

- 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 ***tanslate 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).

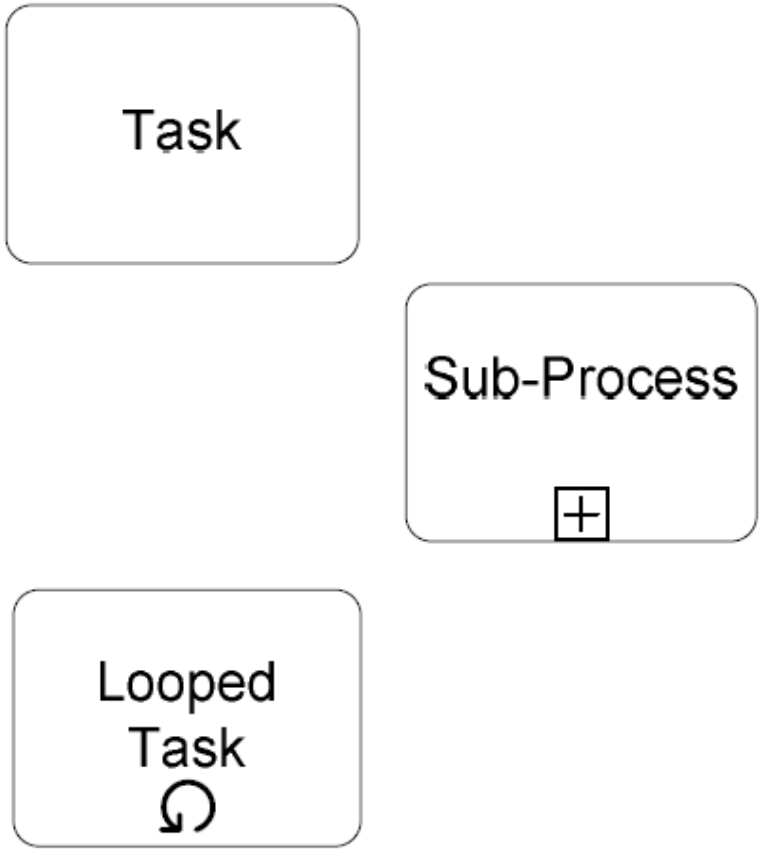
  This small set supports simplicity and readability of models
- To meet the requirements of the second goal, the main elements are
  - specialized for specific purposes, each of which carry further information and/or
  - supported with more elements to allow the modelling of the required behavior

Elements of BPMN

Elements of BPMN can be divided into 4 categories:

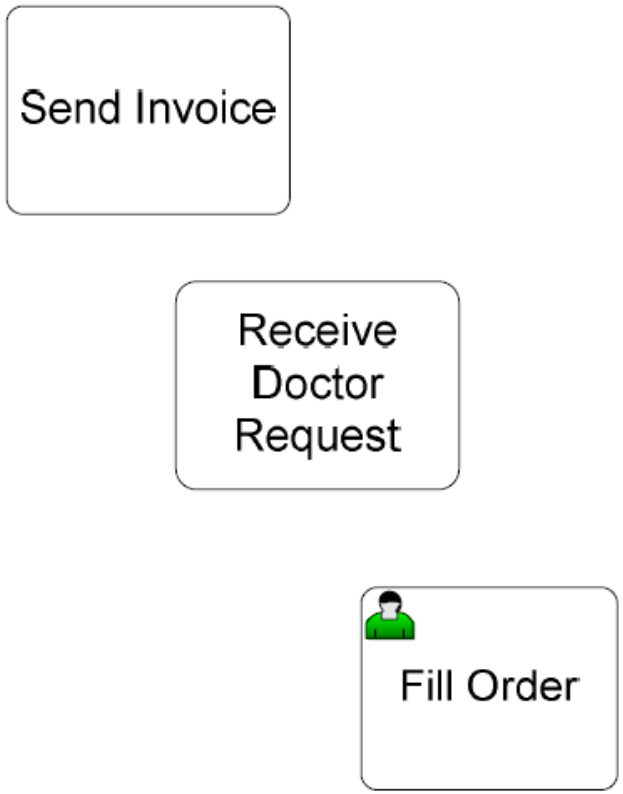


Activities



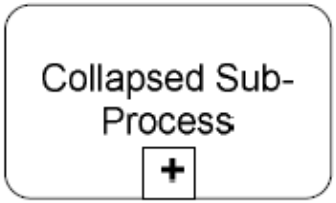
- An activity is work that is performed within a business process.
- An activity can be atomic or non-atomic (compound).
- The types of activities that are a part of a Process Model are:
  - **Task** (atomic)
  - **Sub-Process** (compound)
- Activities are rounded rectangles
- They can be performed once or can have internally defined loops

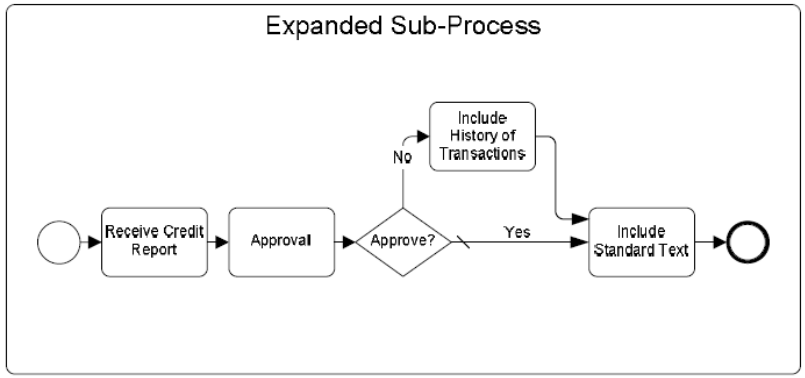
Tasks



- A Task is an atomic activity that is included within a Process.
- A Task is used when the work in the Process is not broken down to a finer level of Process Model detail
- There are specialized types of Tasks for sending and receiving, or user-based Tasks, etc.
- Markers or icons can be added to Tasks to help identify the type of Task
- Markers must not change the footprint of the Task or conflict with any other standard BPMN element

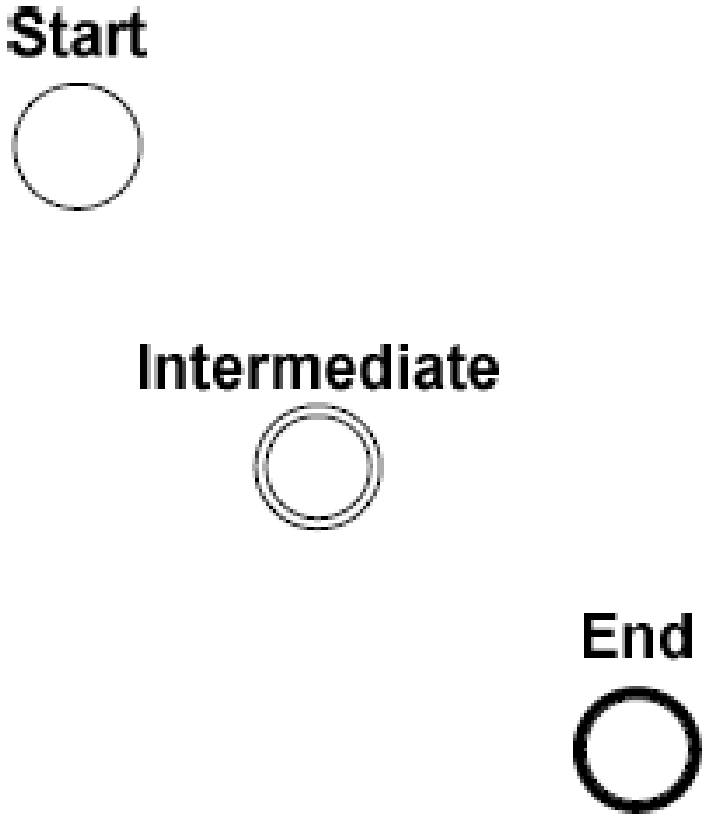
Sub-Processes





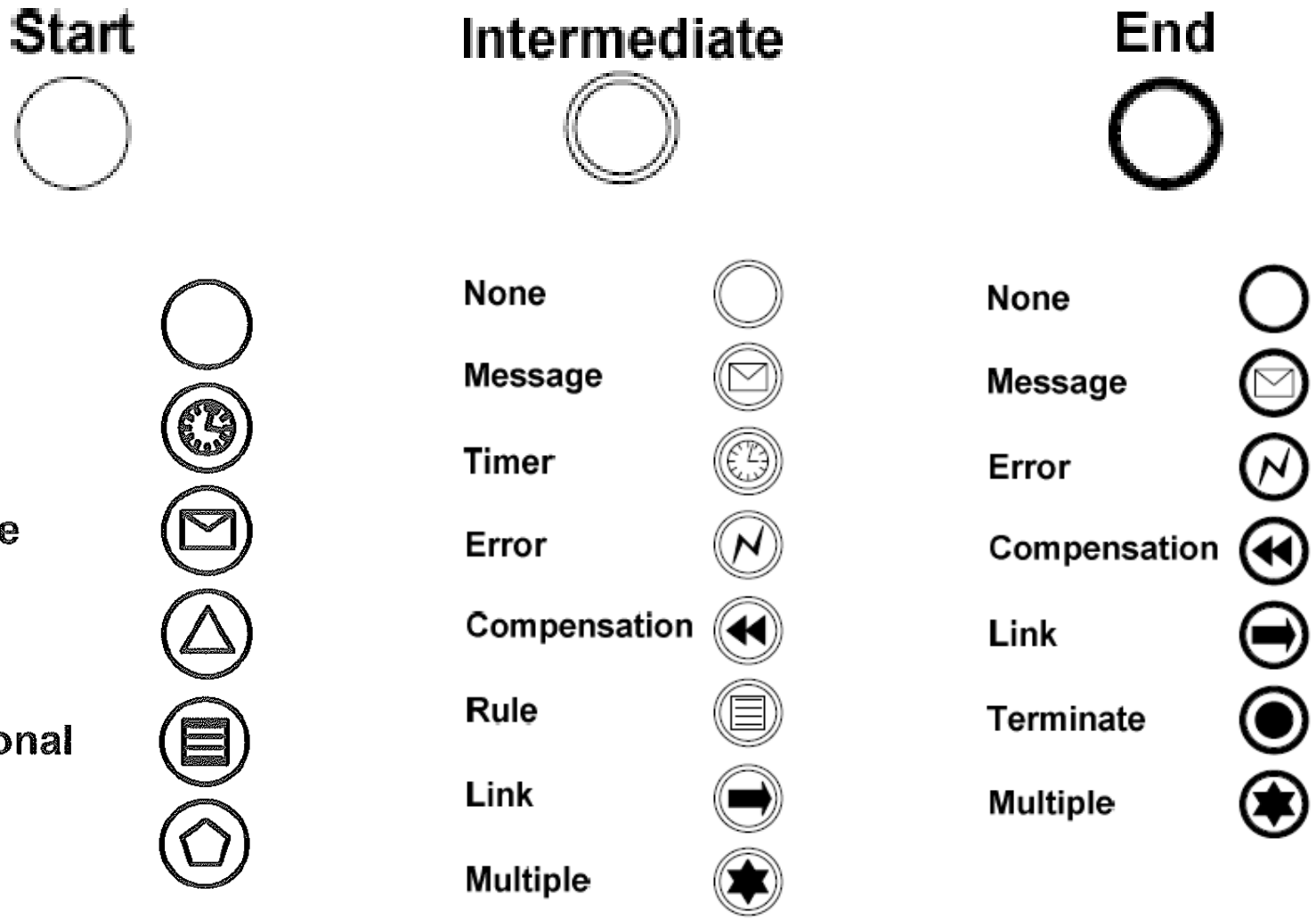
- Sub-Processes enable hierarchical Process development
- A Sub-Process is a compound activity that is included within a Process. It is compound in that it can be broken down into a finer level of detail (a Process) through a set of sub-activities
- For a collapsed version of a Sub-Process, the details 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.
- For an expanded version of a Sub-Process, the details (a Process) are visible within its boundary.
- There are two types of Sub-Processes:
  - Embedded and
  - Independent (Re-usable)

Events



- An Event is something that happens during process execution
- These Events affect the flow of the Process and usually have a trigger or a result.
- They start, interrupt and finish the flow
- Events are represented as circles. The type of boundary determines the type of Event
  - Start Event
  - Intermediate Event
  - End-Event
- In addition, different Event types are distinguished

Event Types


Start
Intermediate
End
None
Timer
Message
Signal
Conditional
Multiple
None
Message
Timer
Error
Compensation
Rule
Link
Multiple
None
Message
Error
Compensation
Link
Terminate
Multiple

Start Events

Core Start Events

None

Timer

Message

Signal

Advanced Start Events

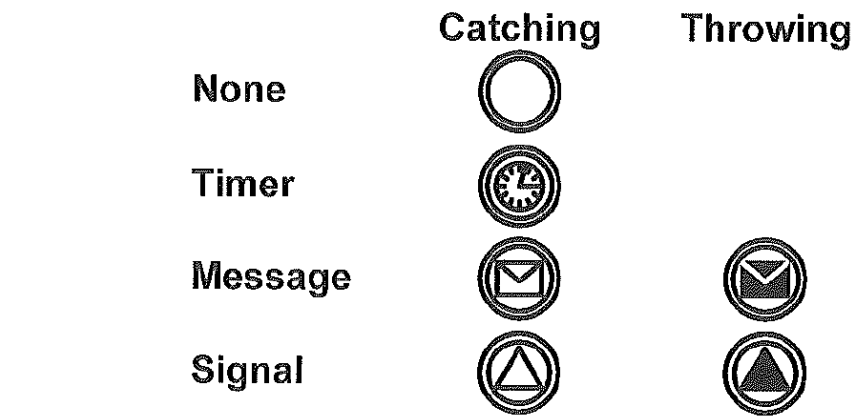
Conditional

Multiple

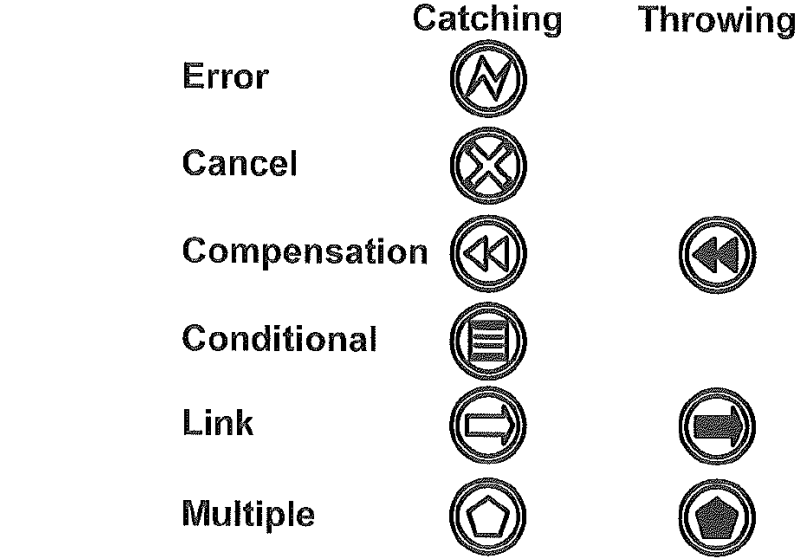
- Start Events indicate when a Process will begin
- There are different “Triggers” that indicate the specific circumstances that start the process
  - **None** – no trigger defined
  - **Timer** – specific date and time of regular date-time cycle (e.g. first Friday iof the month at 8am)
  - **Message** - Arrival of a message from another business entitiy 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



Advanced Intermediate Events

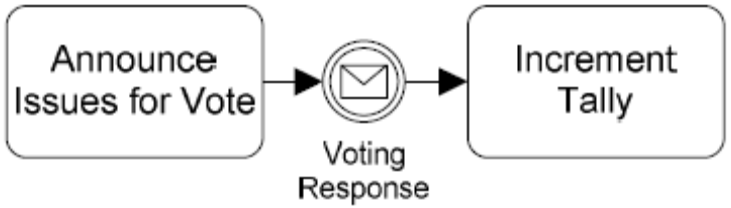


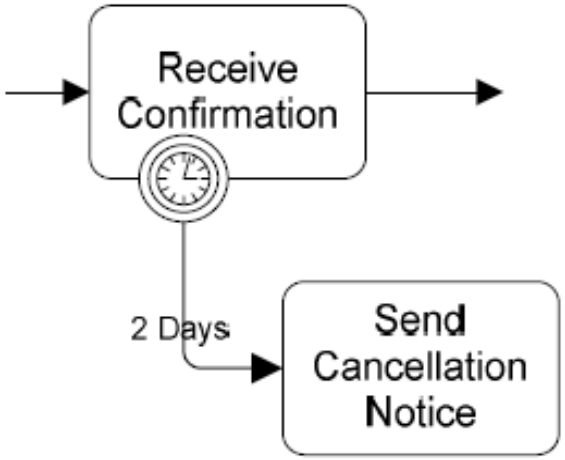
- Intermediate Events occur after a process has been started and before a process is ended
- There are different “Triggers” that indicate the specific circumstances of the Event
- Each type of event can either catch or throw an event

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) inthe process.If shown in the diagram the entities are represented as Pools.
  - **Signal** – signal that is broadcast or received.
- Advanced Intermediate Events
  - **Error** – Event the usually disrupts the Process or requires correction
  - **Cancel** – used to cancel a TransactionSub-Process
  - **Compensation** – establish the behaviour to undo Acitvities
  - **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)

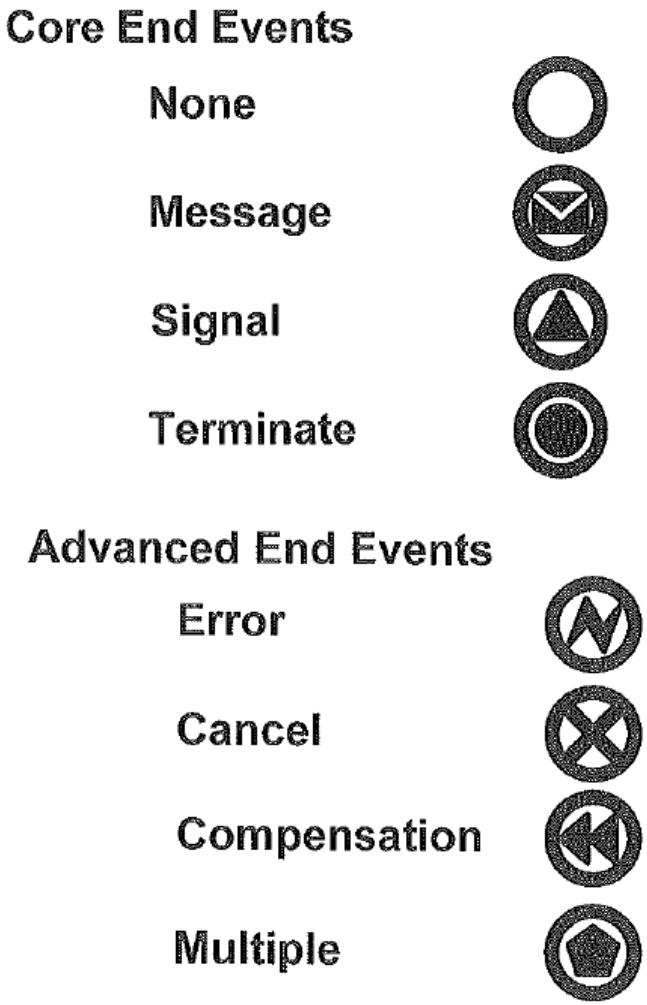
Intermediate Events





- 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

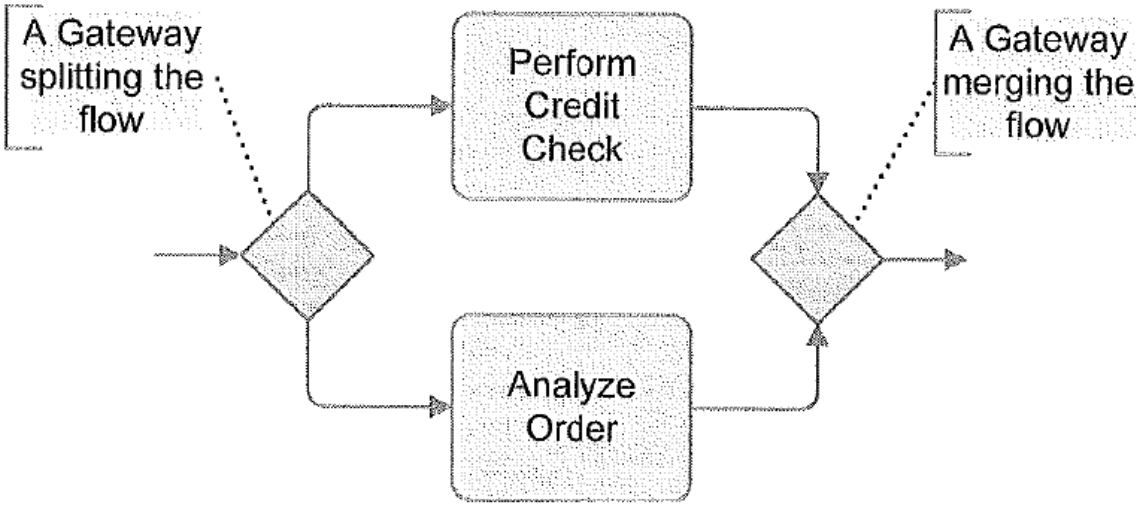
End Events



- 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 resutl is defined
  - **Message** – Communication to another business Entity (participant of Process)
  - **Signal** – Defines a broadcast event taht any other PRocess cansee 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.

Gateways

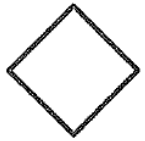
- **Gateways** are modeling elements that control how the Process diverges or converges, i.e. they represent points of control
- They split and merge the flow of a Process how
- All types of Gateways are diamonds
- The underlying idea is that Gateways are unnecessary if the Sequence Flow does not require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



Types of Gateways

Core Gateways

Exclusive



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

Event

Splitting: the path that is taken depends on the occurence of a specified event
Merging: Flow of any incoming path is passed through

Parallel

Parallel

Splitting: all outgoing paths are taken.
Merging: the Gateway waits for all incoming paths (synchronization)

Advanced Gateways

Inclusive

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

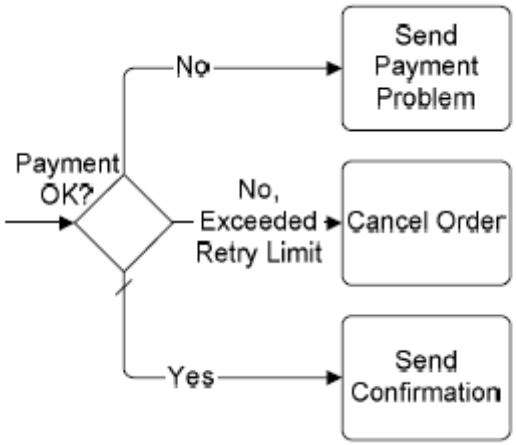
Complex

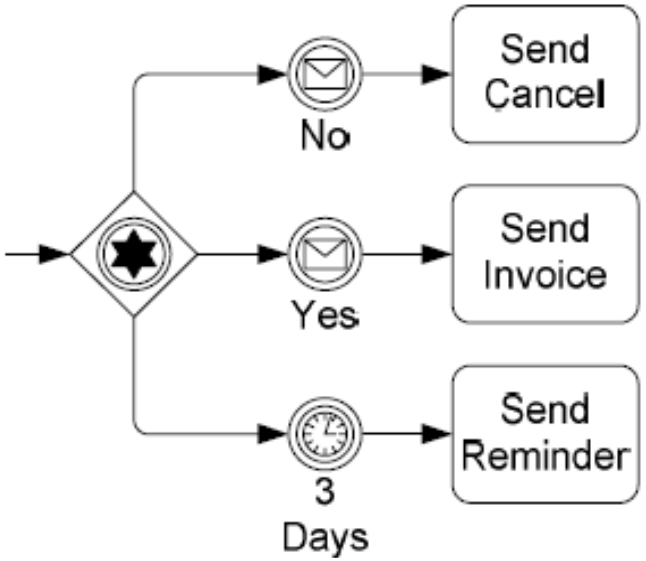
Complex

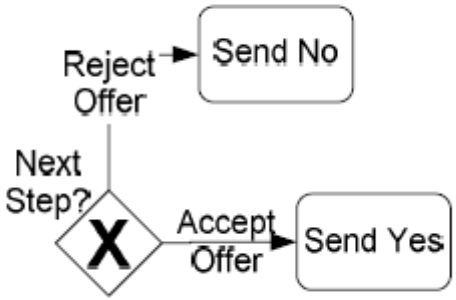
Splitting: one to all of the outgoing paths are taken depending on the evaluation os single Gateway condition
Merging: Flow of incoming path is passed through depending on evaluaion of a single Gateway condtion

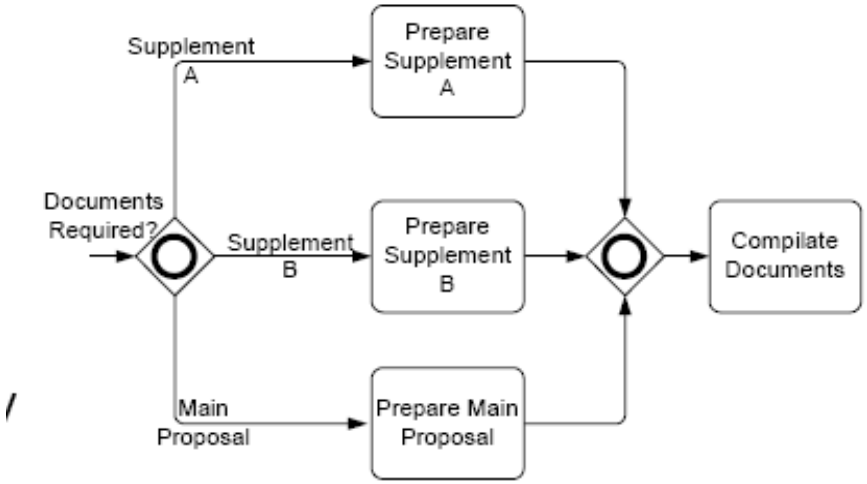
Complex Gateways provide a way for the modeler to specify (program) any behaviour

Examples for the Use of Gateways



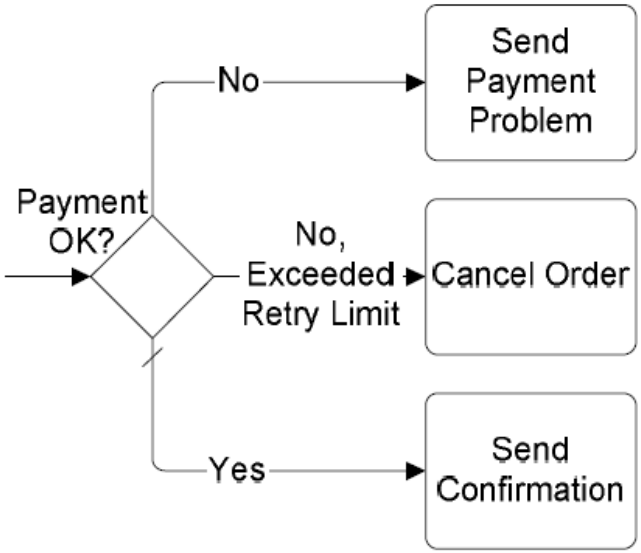


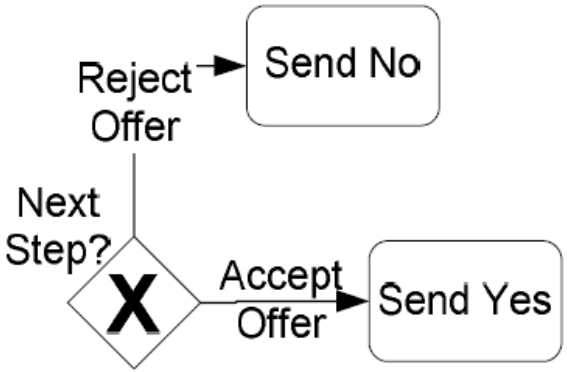




Exclusive Gateways base on Data

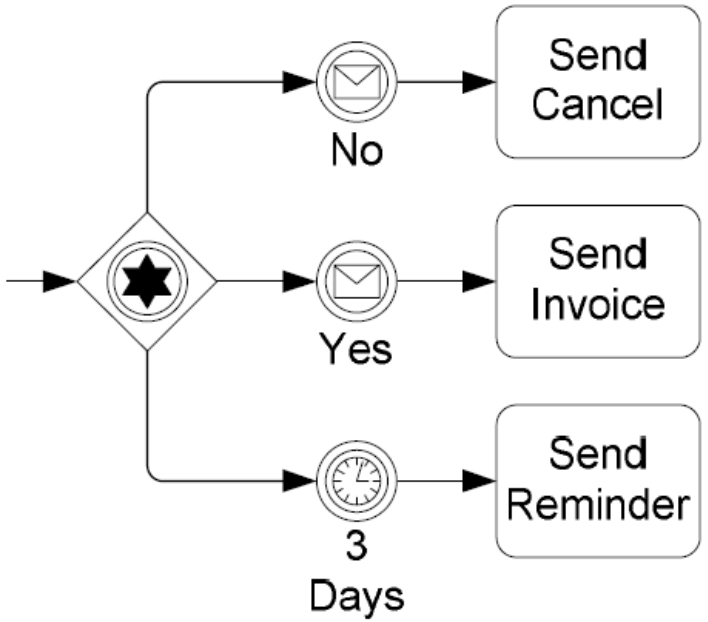
- These are the most commonly used type of Gateways.
  - They can be shown with or without an internal “X” marker. Without is the most common use.
- The Gateway (Decision) creates alternative paths based on defined conditions





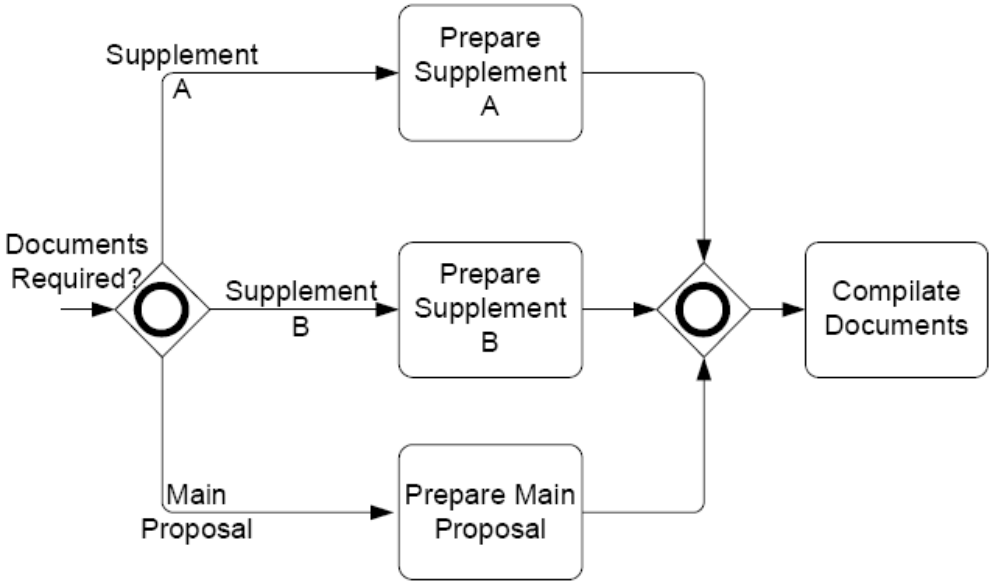
Exclusive Gateways based on Events

- This type of Decision represents a branching point in the process where the alternatives are based on events that occurs at that point in the Process, rather than conditions
- The Multiple Intermediate Event is used to identify this Gateway
- The Event that follow the Gateway Diamond determine the chosen path
  - The first Event triggered wins

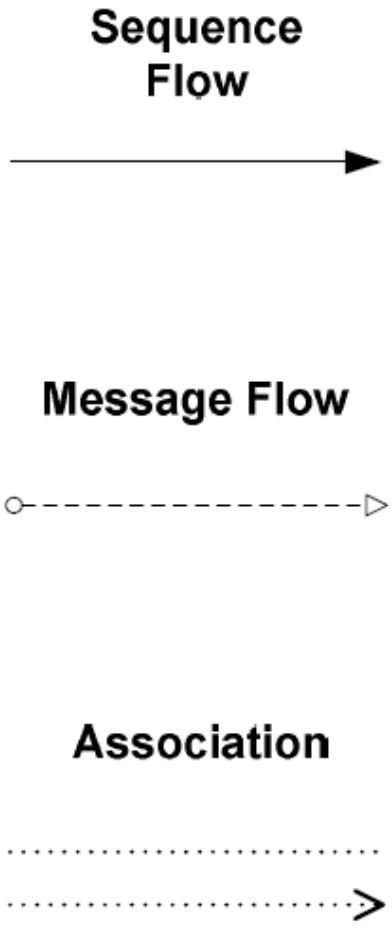


Inclusive Gateways

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



Connectors



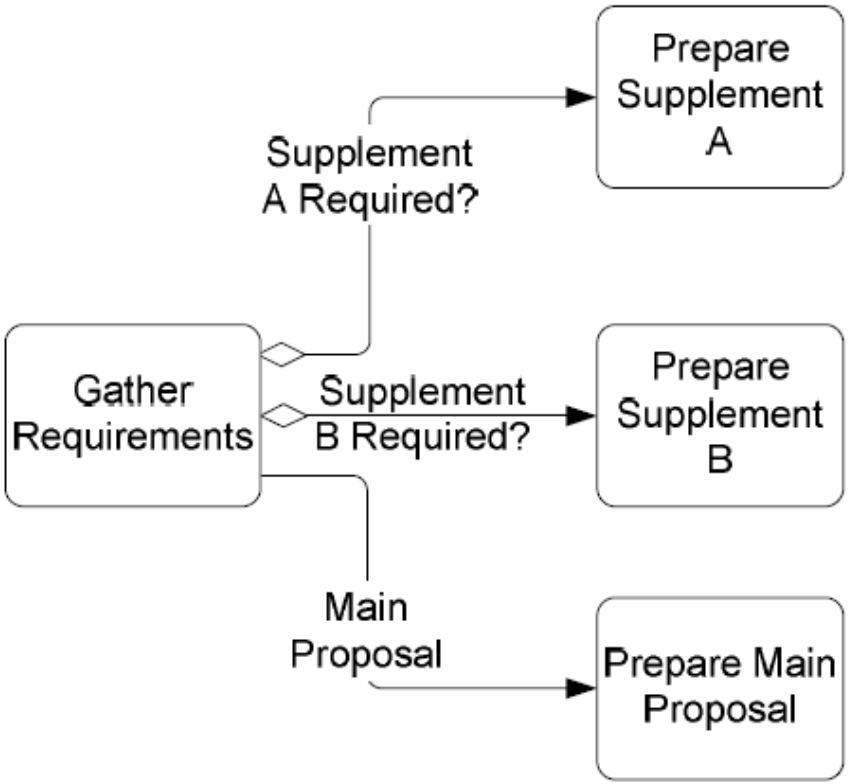
- A **Sequence Flow** is used to show the order that activities will be performed in a Process
- A **Message Flow** is used to show the flow of messages between two entities that are prepared to send and receive them
- An **Association** is used to associate data, information and artifacts with flow objects

Sequence Flow

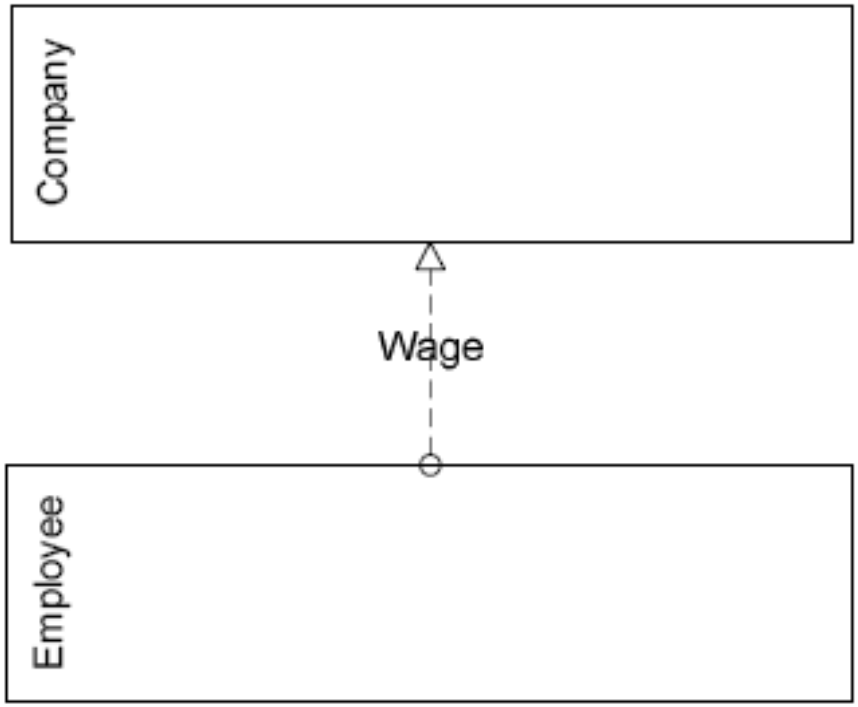
- A Sequence Flow is used to show the order that activities will be performed in a Process
- The source and target must be one of the following objects: Events, Activities, and Gateways
- A Sequence Flow cannot cross a Sub-Process boundary or a Pool boundary

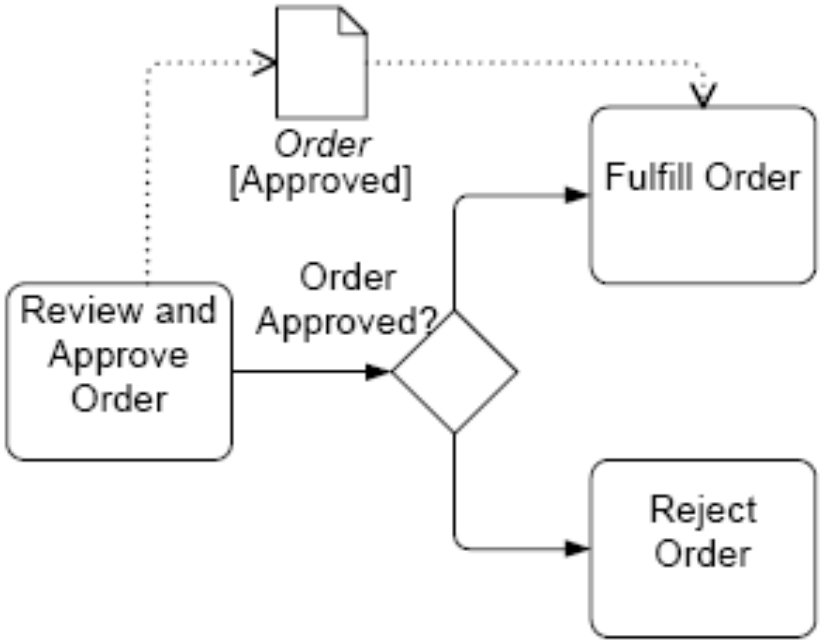
Conditional Sequence Flow

- A Sequence Flow MAY have a defined condition if it exits an activity
  - Such an activity must have at least two Sequence Flows
- The condition has to be True to allow the flow to continue down the Sequence Flow
  - A mini-diamond shows that the Sequence Flow has a condition
- At least one of the outgoing Sequence Flow must be chosen during Process performance



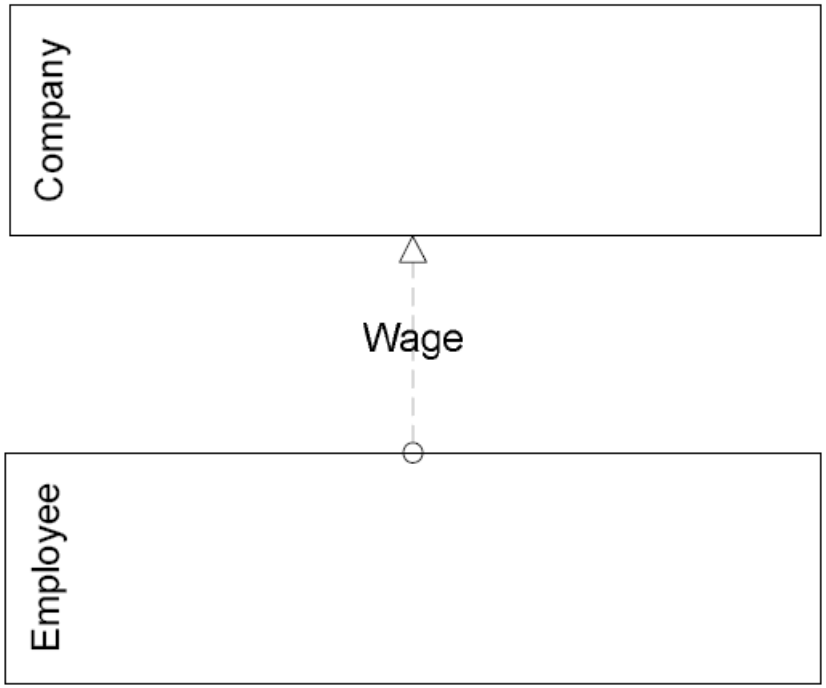
Message Flow und Associations





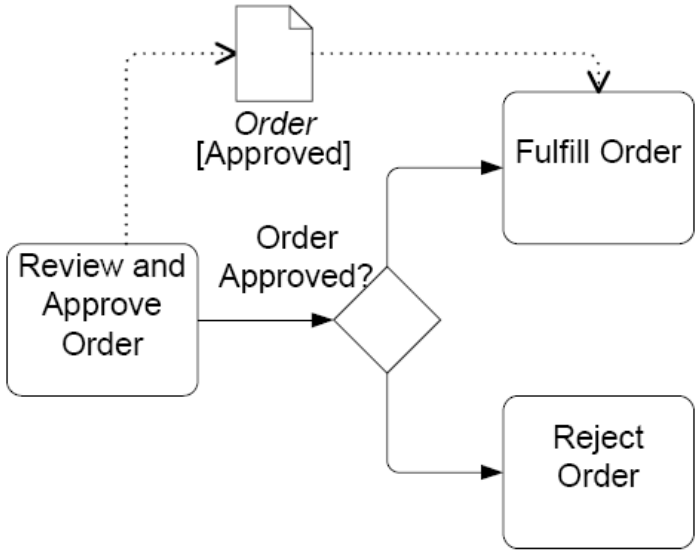
Message Flow

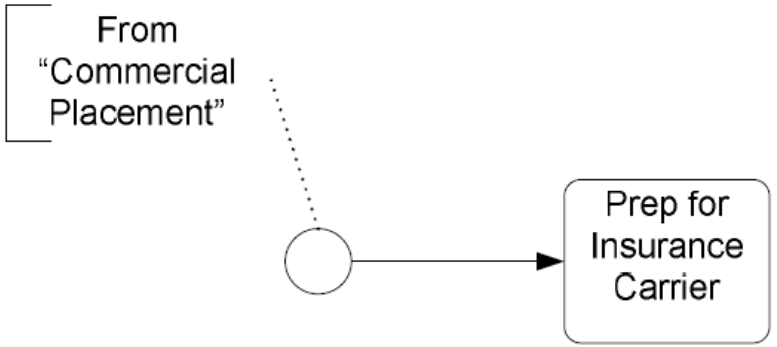
- A Message Flow is used to show the flow of messages between two Participants of Process
  - In BPMN, separate Pools are used to represent the Participants
- A Message Flow can connect to the boundary of the Pool or to an object within the Pool
- Message Flow are not allowed between objects within a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



Associations

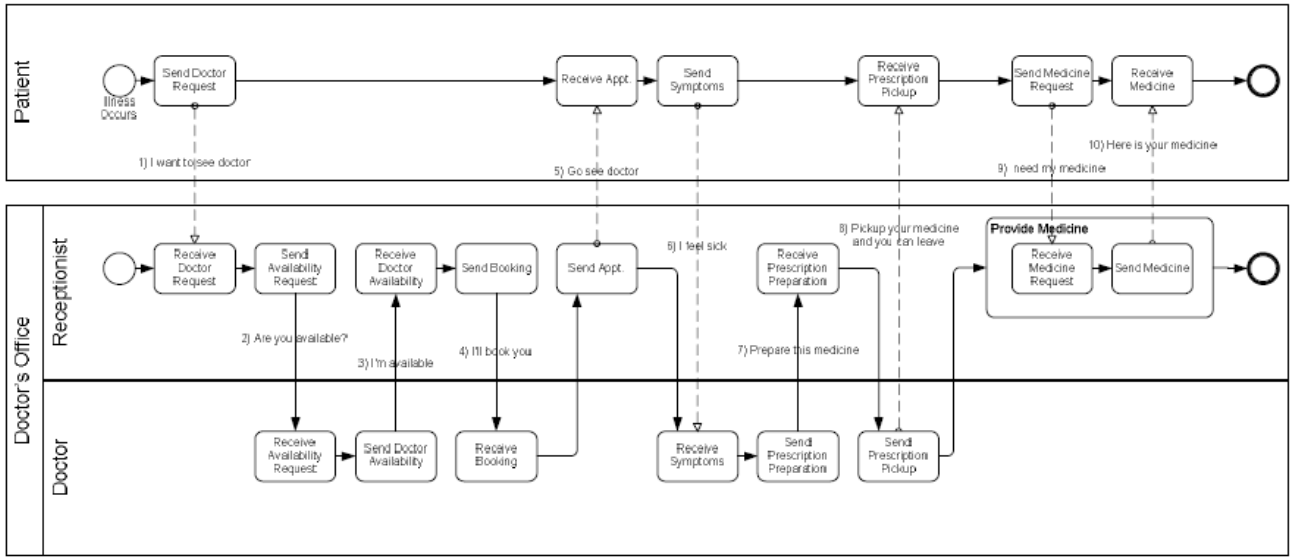
- An Association is used to associate objects to one another (such as Artifacts and Activities)
- Associations are used to show how data is input to and output from Activities
- Text Annotations can be Associated with objects





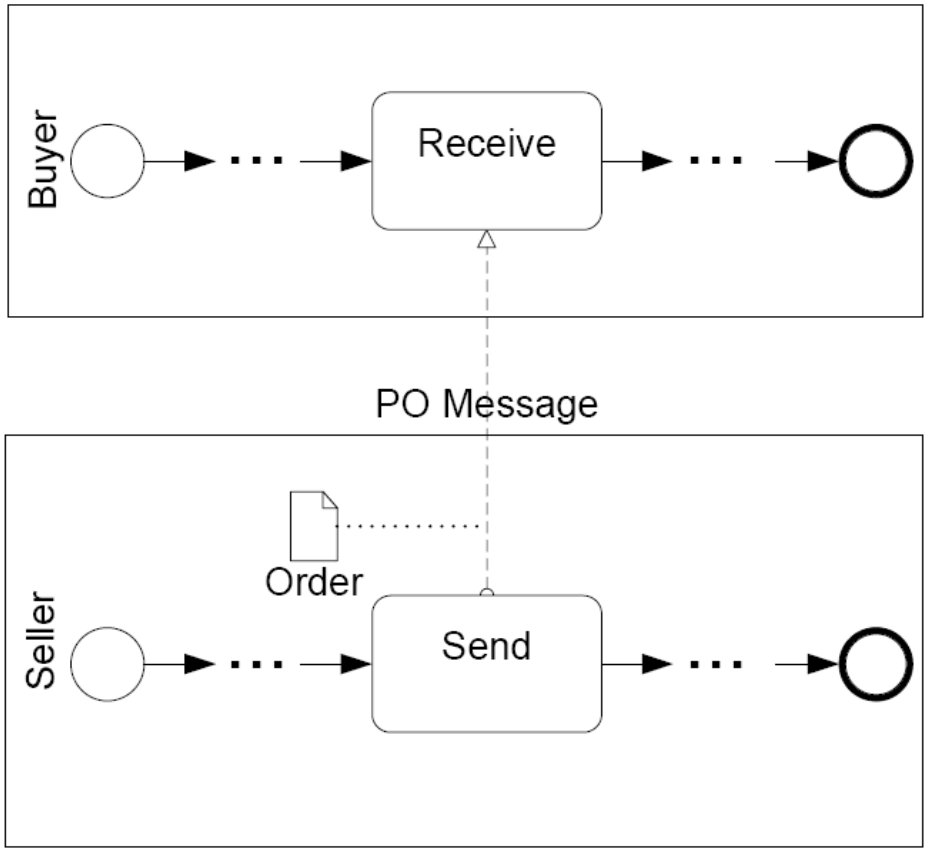
Swimlanes

- BPMN uses the concept known as “swimlanes” to help partition and/organize 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



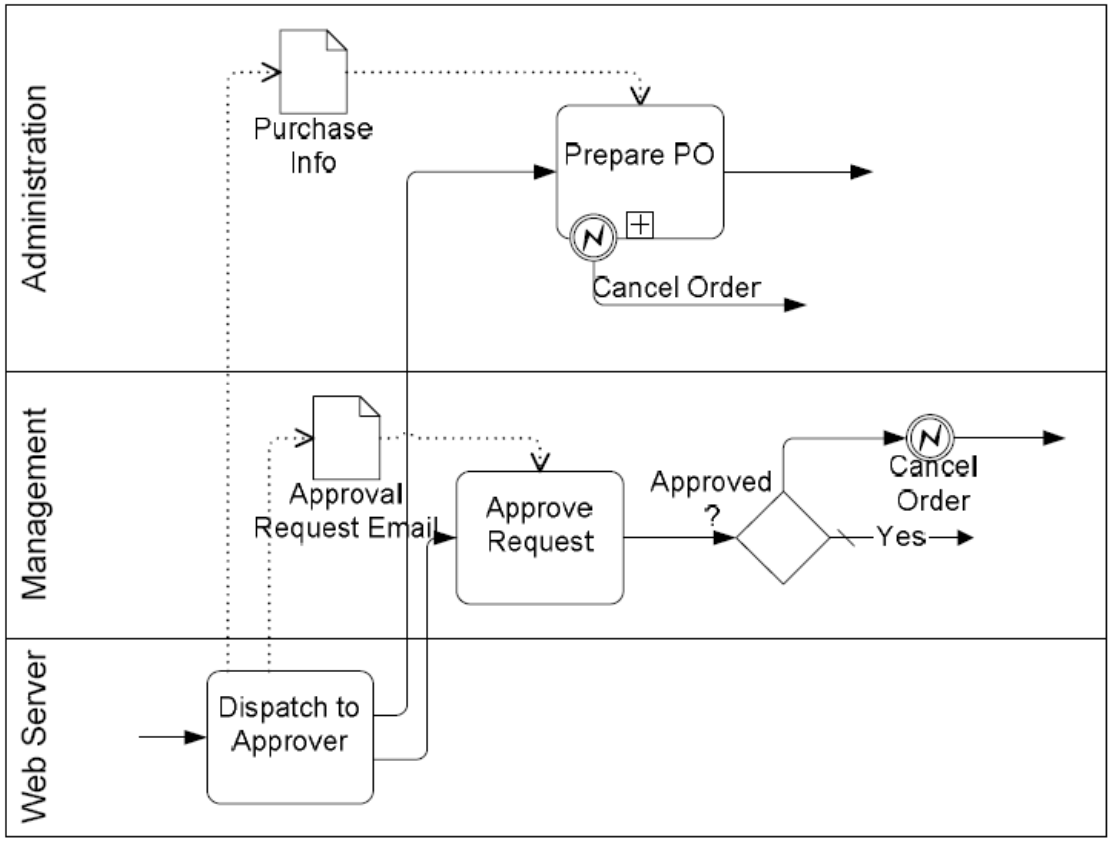
Pools

- Pools represent Participants in an interactive (B2B) Business Process Diagram
  - A Participant may be a business role (e.g., “buyer” or “seller”) or may 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

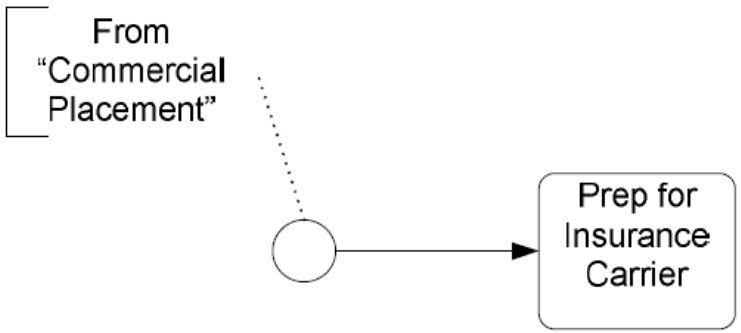
- Lanes represent sub-partitions for the objects within a Pool
- They often represent organization roles (e.g., Manager, Associate), but can represent any desired Process characteristic
- Sequence Flow can cross Lane boundaries

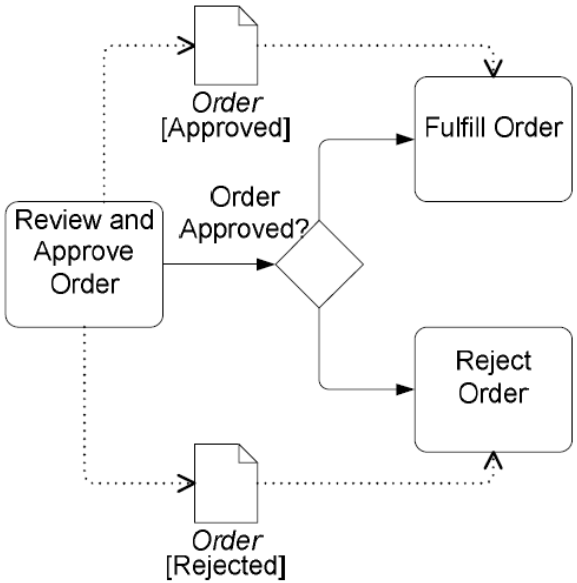


Artifacts

- 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

Text Annotations and Data Objects



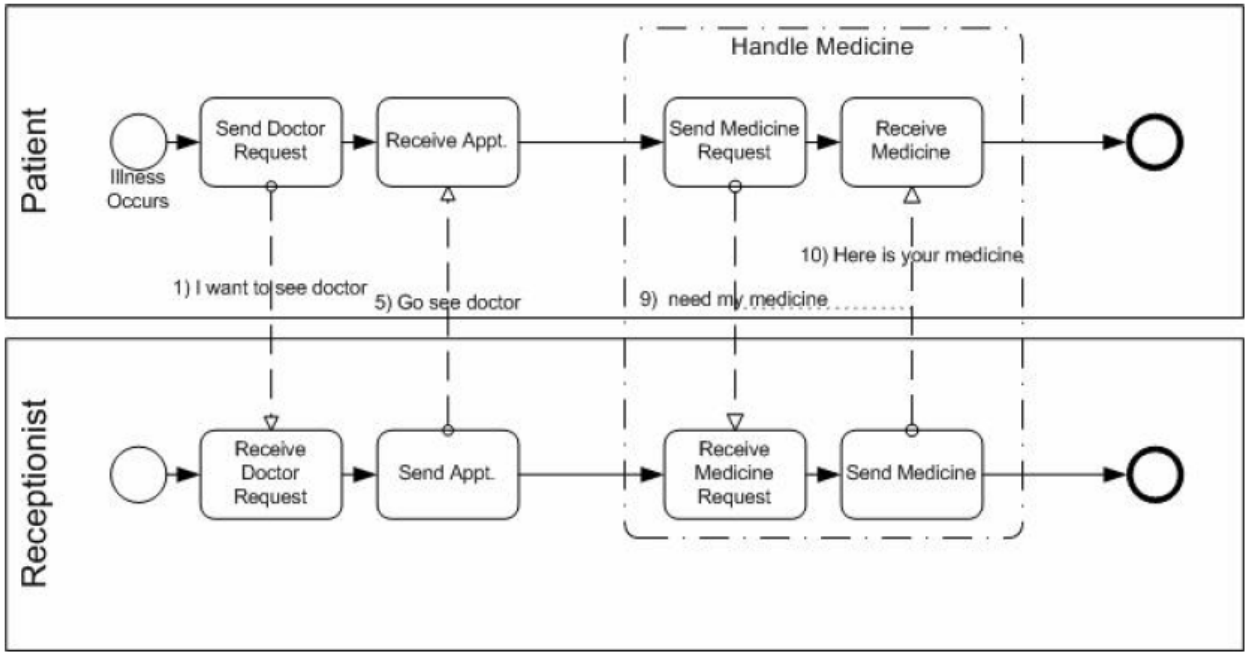


- Text Annotations are a mechanism for a modeler to provide additional information about a Process
- Text Annotations can be connected to a specific object on the Diagram with an Association

- 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

Groups

- Groups are Artifacts that are used to highlight certain sections of a Diagram without adding additional constraints for performance – as a Sub-Process would
  - Groups can be used to categorize elements for reporting purposes
- Groups are not constrained by restrictions of Pools and Lanes



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