BPMN
Business Process Model and Notation

Modeling Language Components

- A modeling language consists of three parts:
  - ***Syntax***
    - Set of modeling elements and rules to combine them
      - BPMN syntax includes activities, events, gateways, sequence flows
  - ***Semantics***
    - Bind syntactical elements with textual descriptions to a precise meaning
      - Behavior of BPMN elements
  - ***Notation***
    - Defines graphical symbols for elements

Business Process Model and Notation (BPMN)

- Object Management Group (OMG) standard
  - http://www.omg.org
  - http://www.omg.org/specs/bpmn
  - version 1.0 (2006)
  - version 1.1 (2007)
  - version 1.2 (2009)
  - version 2.0 (2011)

- Provide a notation to describe Business Process understandable by:
  - BP analysts
  - IT developers
  - BP workers and managers

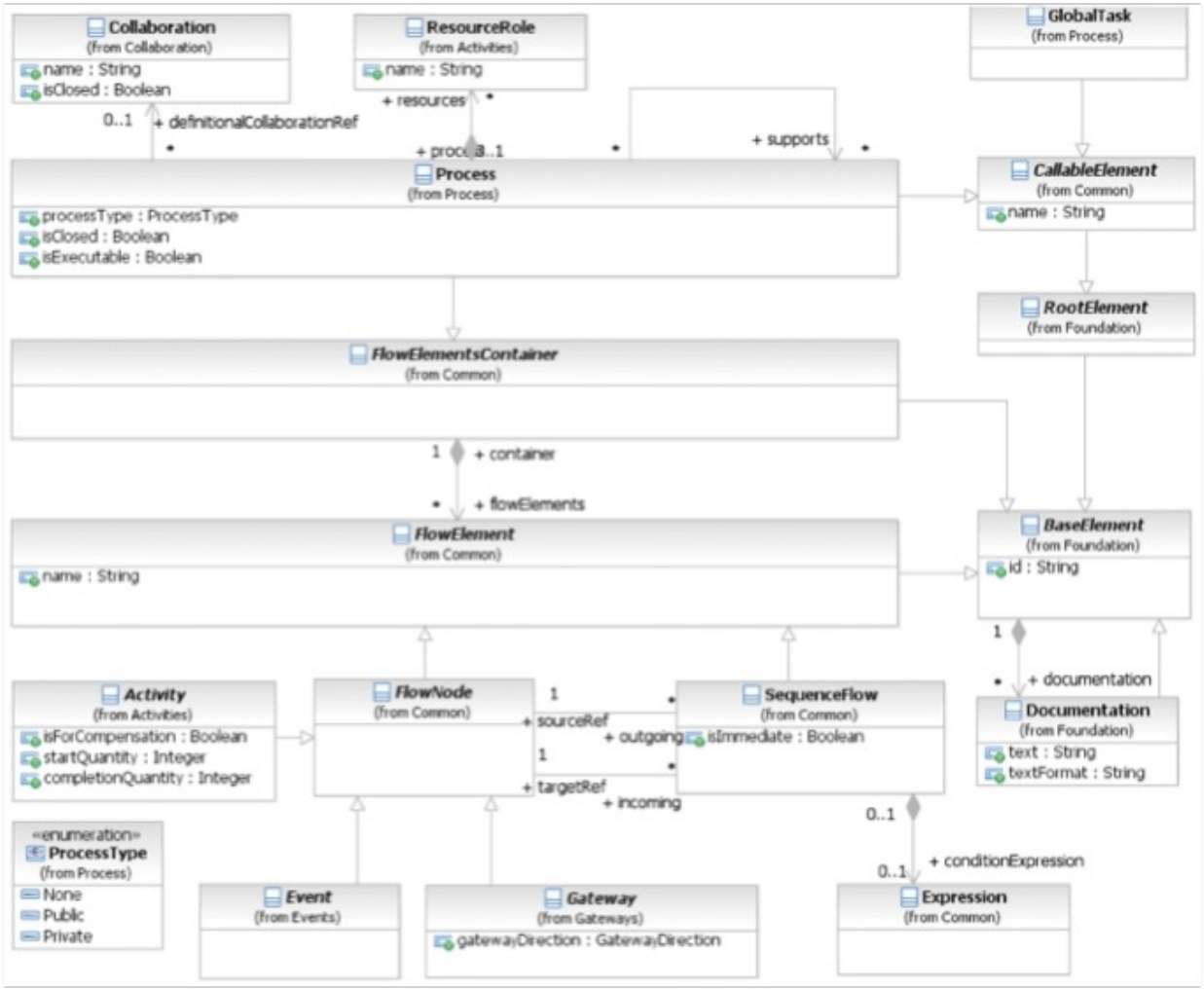
Why BPMN

- Business people are very comfortable with visualizing Business Processes in a flow-chart format
- BPMN ***execution semantics*** is fully formalized
- BPMN 2.0 has a ***formal definition*** (metamodel)
  - Precise definition of the constructs and rules for creating models

BPMN Metamodel

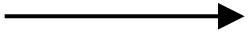
- Metamodeling has the following benefits
  - Formalization of models and entities
  - Formalization of relationship between elements
  - Interoperability
- It is not necessary for the modeler to handle the metamodel
- Is the modeling tool that ensures the model is compliant with metamodel

BPMN Metamodel (partial)



BPMN Semantic

- To describe how the BPMN elements behave, the theoretical concept of ***token*** is used
  - The "***simulated***" behavior of elements can be defined by describing how they interact with a token
  - The token is not part of the BPMN specification
  - BPMN modeling tools are not required to implement any form of token
- The token will traverse the sequence flows and pass through the elements in the process
  - Token traverse sequence flows instantaneously, so there is no time associated with them
  - When arrives at an element, the token may continue instantaneously or can be delayed depending on the element
- We use this notation for tokens 



Sequence Flow

- Sequence flow connects model elements showing their ***order of execution***

- Each sequence flow has only one source and only one target
  - Model elements can have one or more incoming sequence flow and one or more outgoing sequence flow but usually they have one

- Source and target can be activities, events, gateways

Start Event

- Beginning of the process
  - It has no incoming flows
- A process may have zero, one or more start events
  - If not present, all activities without incoming flows start together
  - If an end event is present, at least one start event is mandatory

End Event

- Where the flow of the process ends
  - It has no outgoing flows
- A process may have zero, one or more end events
  - If not present, all activities that do not have any outgoing flows mark the end of a process path
    - In this case, the Process ends when all parallel paths have completed
- If a start event is present, at least one end event is mandatory

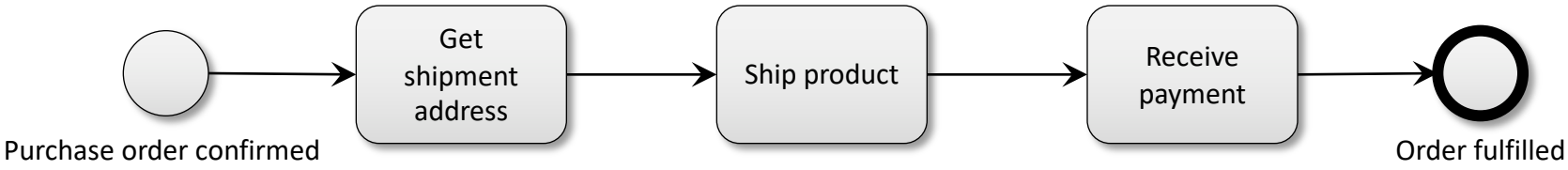
Task

- Unit of work (atomic activity), the job to be performed

- A task is used when the work in the process cannot be broken down to a finer level of detail

Our First BPMN Model

- A simple BPMN model generally has
  1. A start point
  2. An end point
  3. Some activities
  4. Connections between activities



Gateway

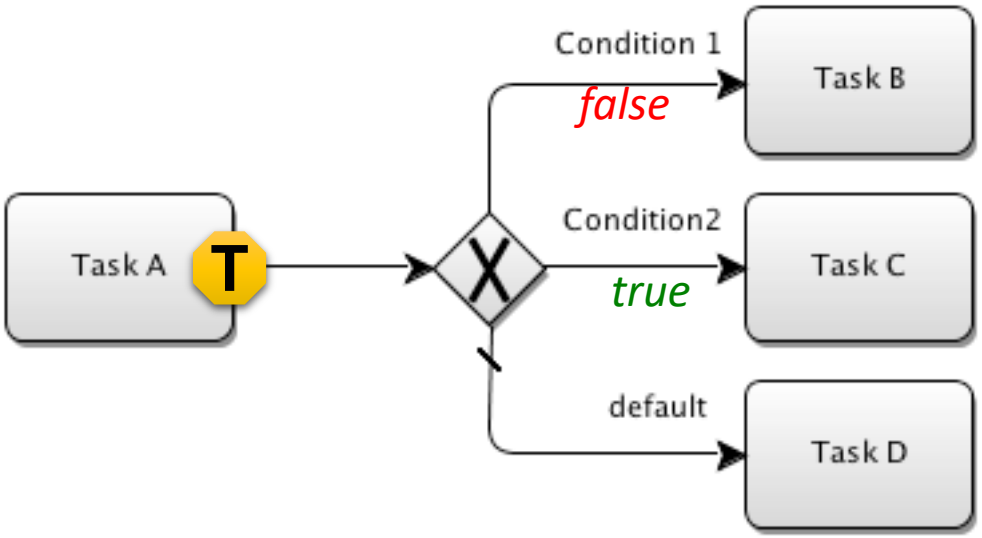
- Used to control how sequence flows interact as they converge and diverge within a process

- ***Split gateway***
  - Point where the process flow diverges
  - Has one incoming sequence flow and multiple outgoing sequence flows

- ***Merging gateway***
  - Point where the process flow converges
  - Has multiple incoming sequence flows and one outgoing sequence flow

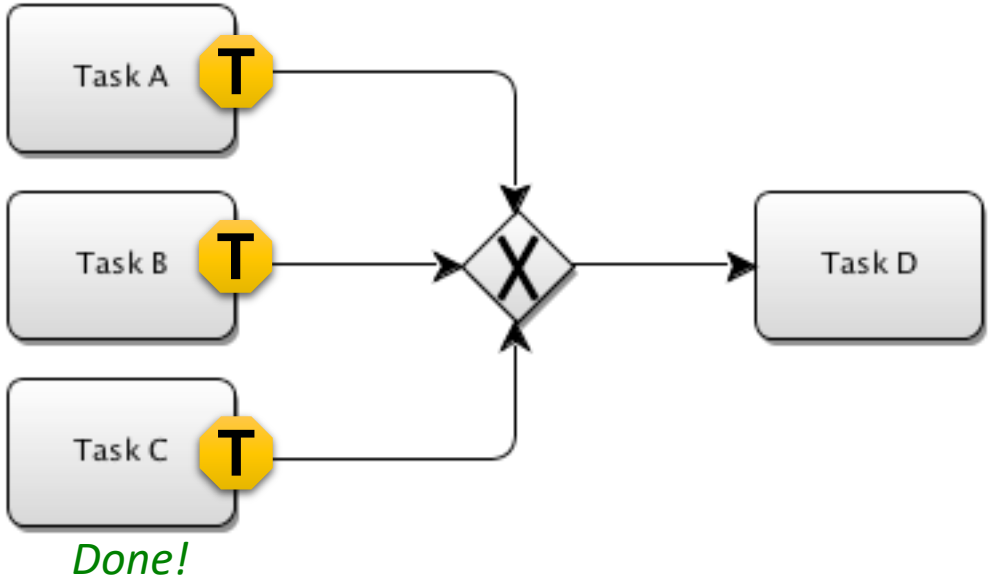
Exclusive Gateway

- ***Diverging Exclusive Gateway***
  - Creates alternative paths within a process flow
  - The decision is considered as a question with a defined set of alternative answers
  - Only one of the paths can be taken (XOR)
  - The default path (if defined) is taken if none of the conditional expressions evaluate to true



Exclusive Gateway

- ***Converging Exclusive Gateway***
  - Used to merge alternative paths
  - Each incoming token is routed to the outgoing sequence flow without synchronization

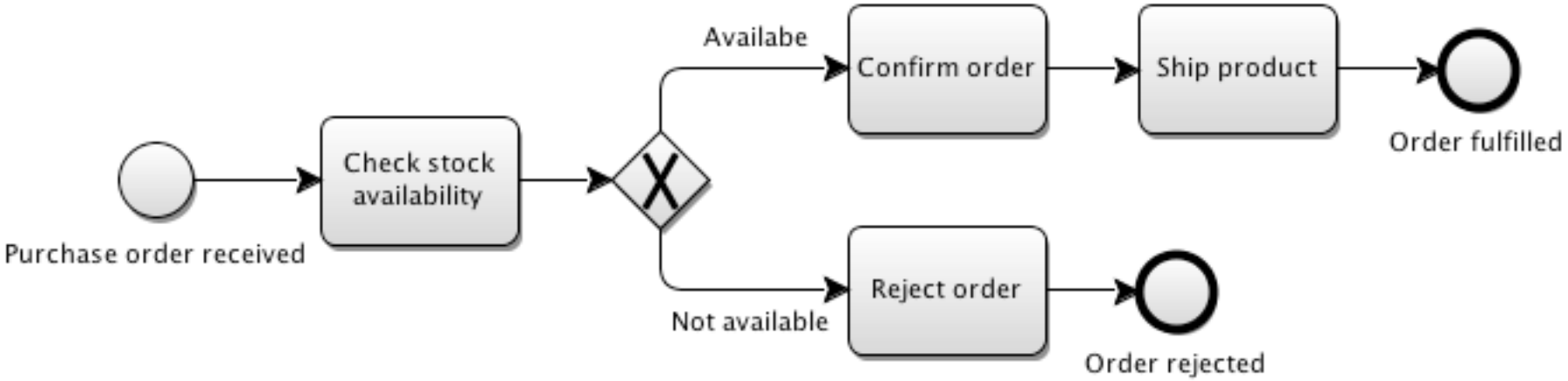


Exclusive Gateway Example

- Order shipment
  - *When a purchase order is received, check stock availability. If the item is available, confirm order and ship product, otherwise reject order*.

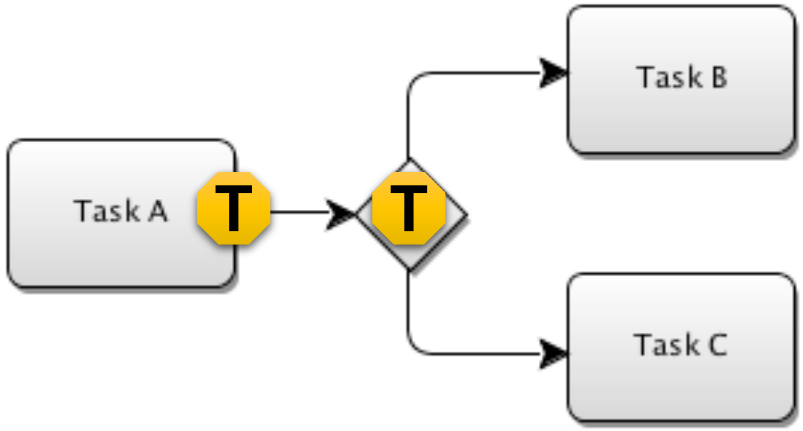
Exclusive Gateway Example

- Order shipment
  - *When a purchase order is received, check stock availability. If the item is available, confirm order and ship product, otherwise reject order*



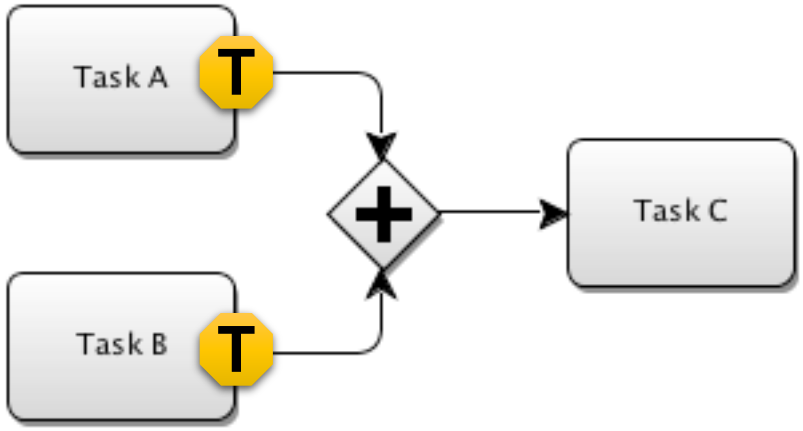
Parallel Gateway

- ***Diverging Parallel Gateway***
  - Creates parallel paths without checking any conditions
  - Each outgoing sequence flow receives a token



Parallel Gateway

- ***Converging Parallel Gateway***
  - Used to synchronize (combine) parallel flows and to create parallel flows
  - Waits for all incoming flows before routing the token to the outgoing flows (AND)

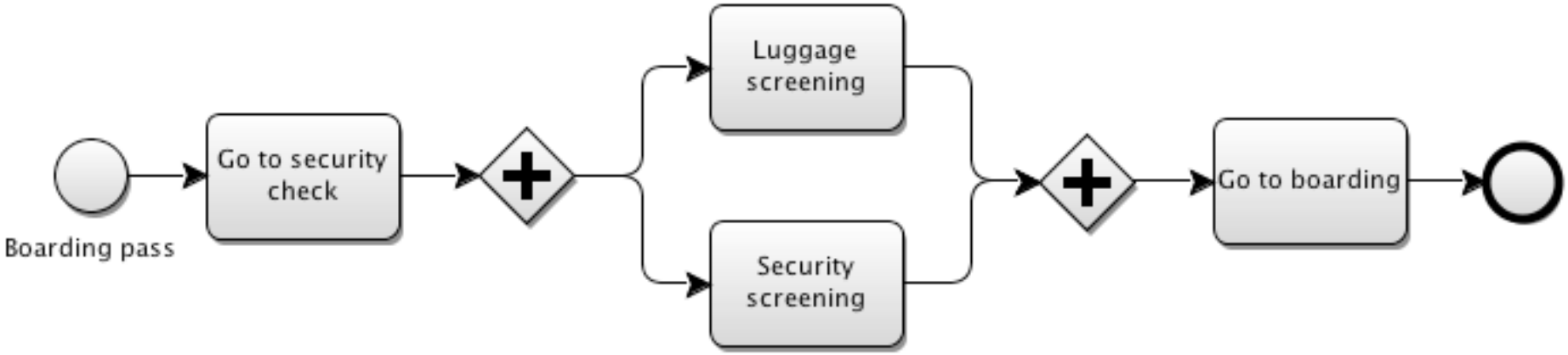


Parallel Gateway Example

- Boarding security check
  - *Having the boarding pass, go to security check for luggage and security screening. When passed both go to boarding*

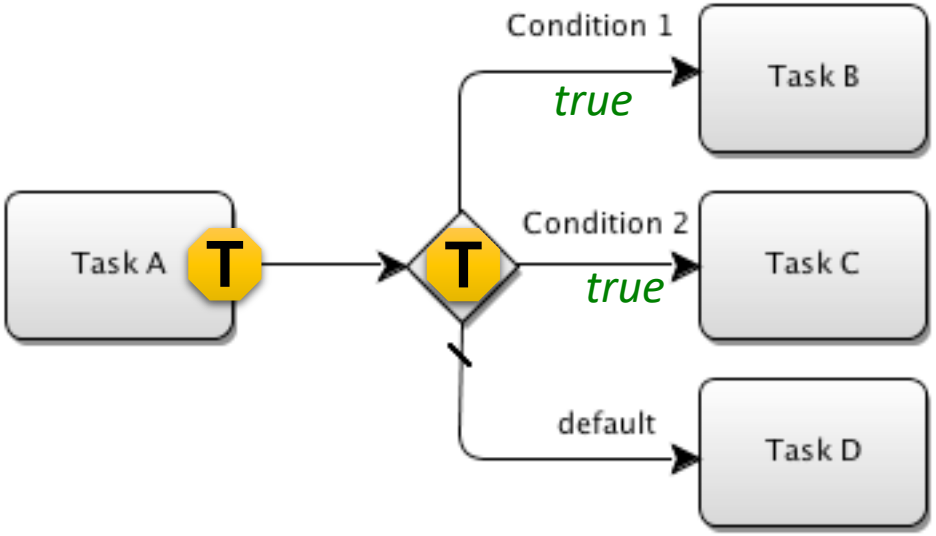
Parallel Gateway Example

- Boarding security check
  - *Having the boarding pass, go to security check for luggage and security screening. When passed both go to boarding*



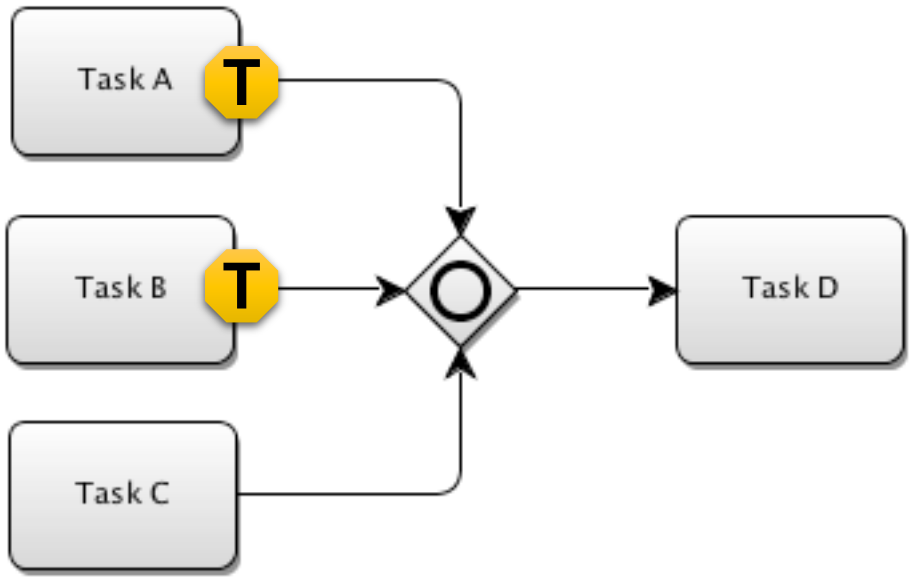
Inclusive Gateway

- ***Diverging Inclusive Gateway***
  - Used to create alternative but also parallel paths within a process flow
  - Unlike the exclusive gateway, all condition expressions are evaluated
    - All Sequence Flows with a true evaluation will be traversed by a token (OR)
    - The default path (if defined) is taken if none of the conditional expressions evaluate to true



Inclusive Gateway

- ***Converging Inclusive Gateway***
  - Used to merge a combination of alternative and parallel paths
  - Token arriving at an inclusive gateway may be synchronized with some other tokens that arrive later at the gateway
    - When all the expected tokens have arrived the token moves to the outgoing sequence flow

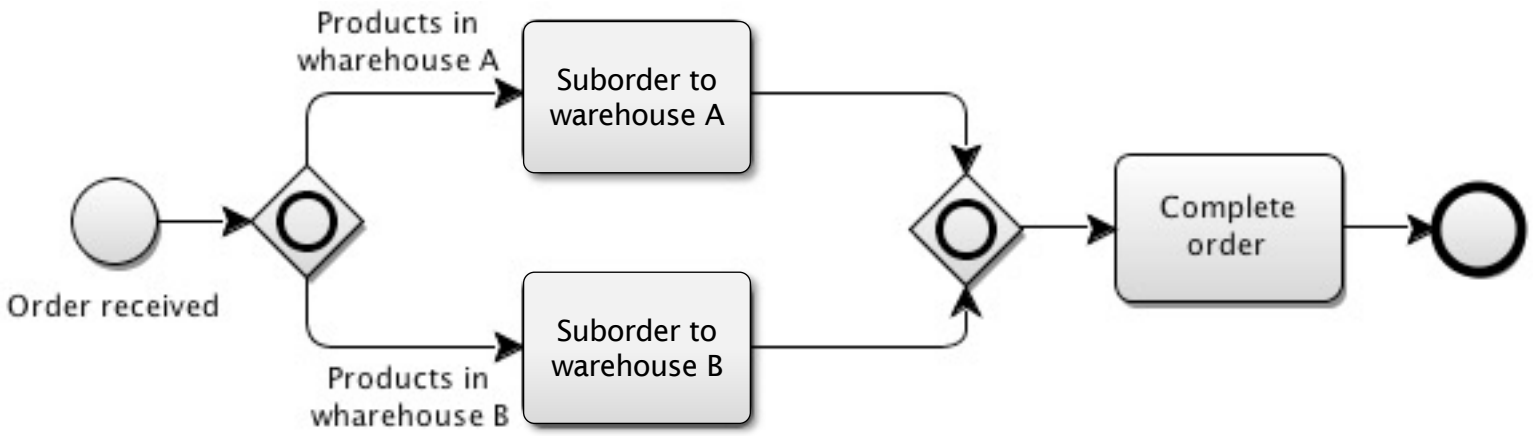


Inclusive Gateway Example

- Order decomposition
  - *When an order is received, get product of type A from warehouse A and products of type B from warehouse B*.

Inclusive Gateway Example

- Order decomposition
  - *When an order is received, get product of type A from warehouse A and products of type B from warehouse B.*



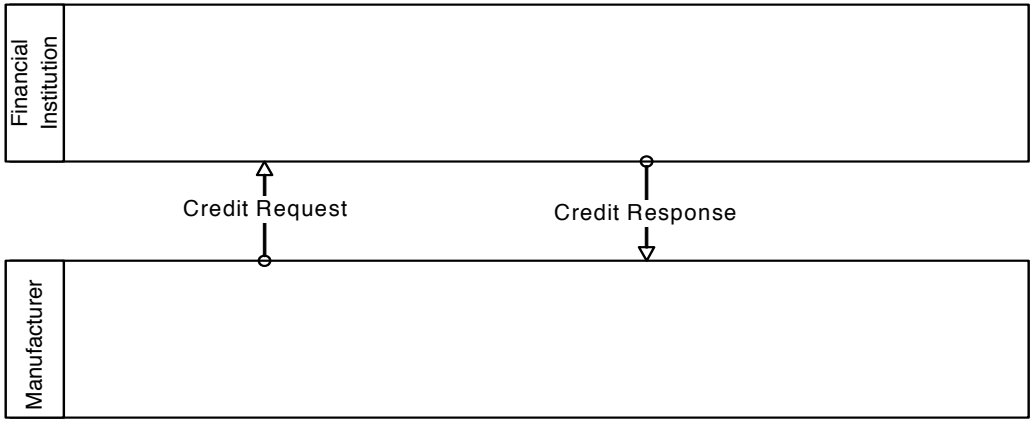
Resource Modeling

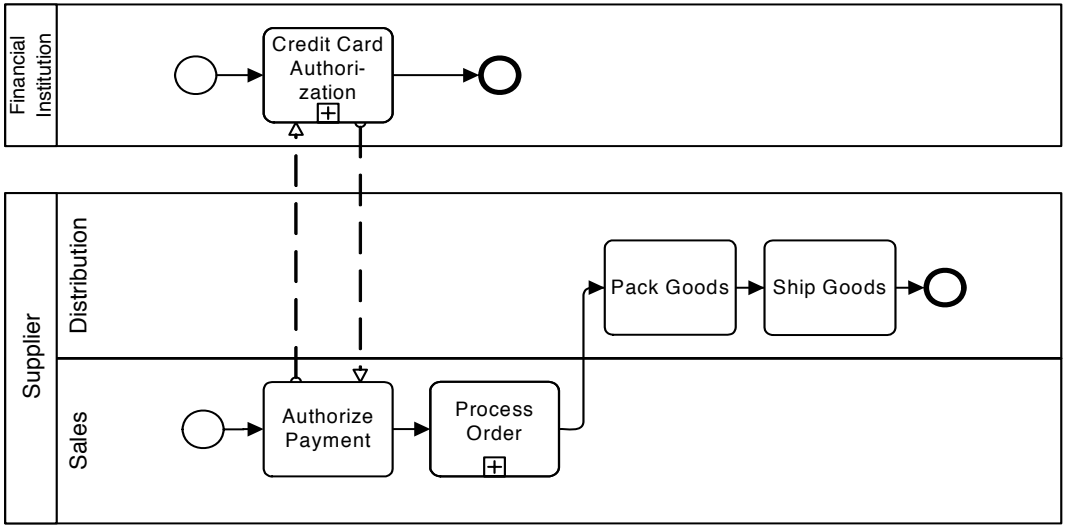
- Organizational resource are mapped in pools and lanes

- ***Pools***
  - Independent organizational entities (***do not share*** any common system that allows them to communicate implicitly)
    - Customer is independent from the Supplier

- ***Lanes***
  - Multiple resource classes in the same organization (***share*** common systems)
    - Sales department and marketing Department

Pools and Lanes





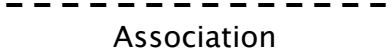
Message Flow

- Used to show the flow of messages between two participants
  - Participants are prepared to send and receive the messages
- Message flow only connects two separate pools
  - Message flow can be attached to pools, activities, or message events
  - Message flow cannot connect two objects within the same pool

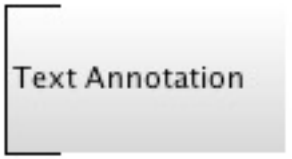
Artifacts

- Additional information that is not directly related to the sequence flows or message flows
  - Associations, groups, text annotations

- Association is used to associate artifacts with process elements

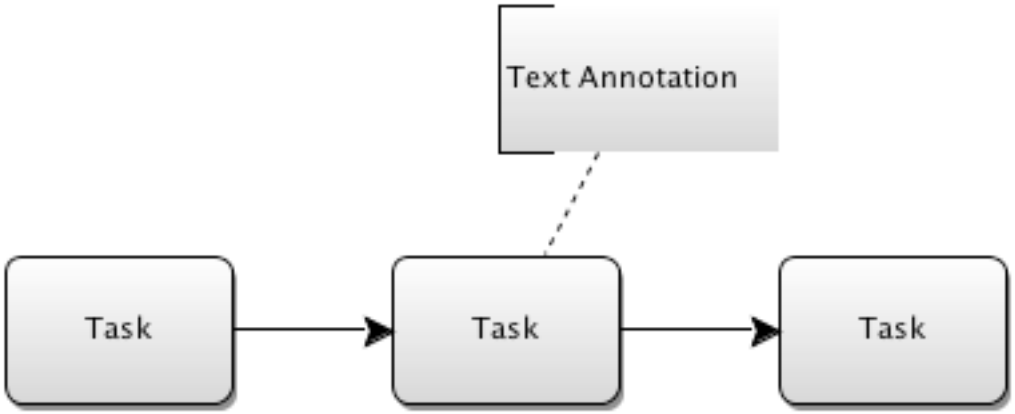


Association



Text Annotation

- Provides additional information for the reader of a BPMN diagram
  - Does not affect the flow of the process



Data Objects

- Represent data and document exchanged in the process
- Can be used to show input and output of activities
- BPMN defines 5 kind of data objects

Data Objects

- ***Data object***
  - Information traversing the process (email, letters, documents)

- ***Collection data object***
  - Collection of information

- ***Data input***
  - External input necessary to start the activity

- ***Data output***
  - Outcome of the activity or the process

- ***Data store***
  - Place where the process can read and write data (database)

BPMN Process Type

- ***Orchestration***
- ***Collaborations***
- ***Choreographies***

- BPMN uses the terms collaboration and choreography when modeling the interaction between processes

Orchestration Example

- BP ***internal*** to a specific organization
- Generally called workflow or BPM process

Image from "Business Process Model and Notation (BPMN)", http://www.omg.org/spec/BPMN/2.0

Collaboration

- Collaboration shows ***interactions*** between two or more participants
  - A pool represents one of the participants in the collaboration
- Collaboration can be shown as two or more processes communicating with each other
  - The message exchanged between the participants is shown by a message flow

Collaboration Example

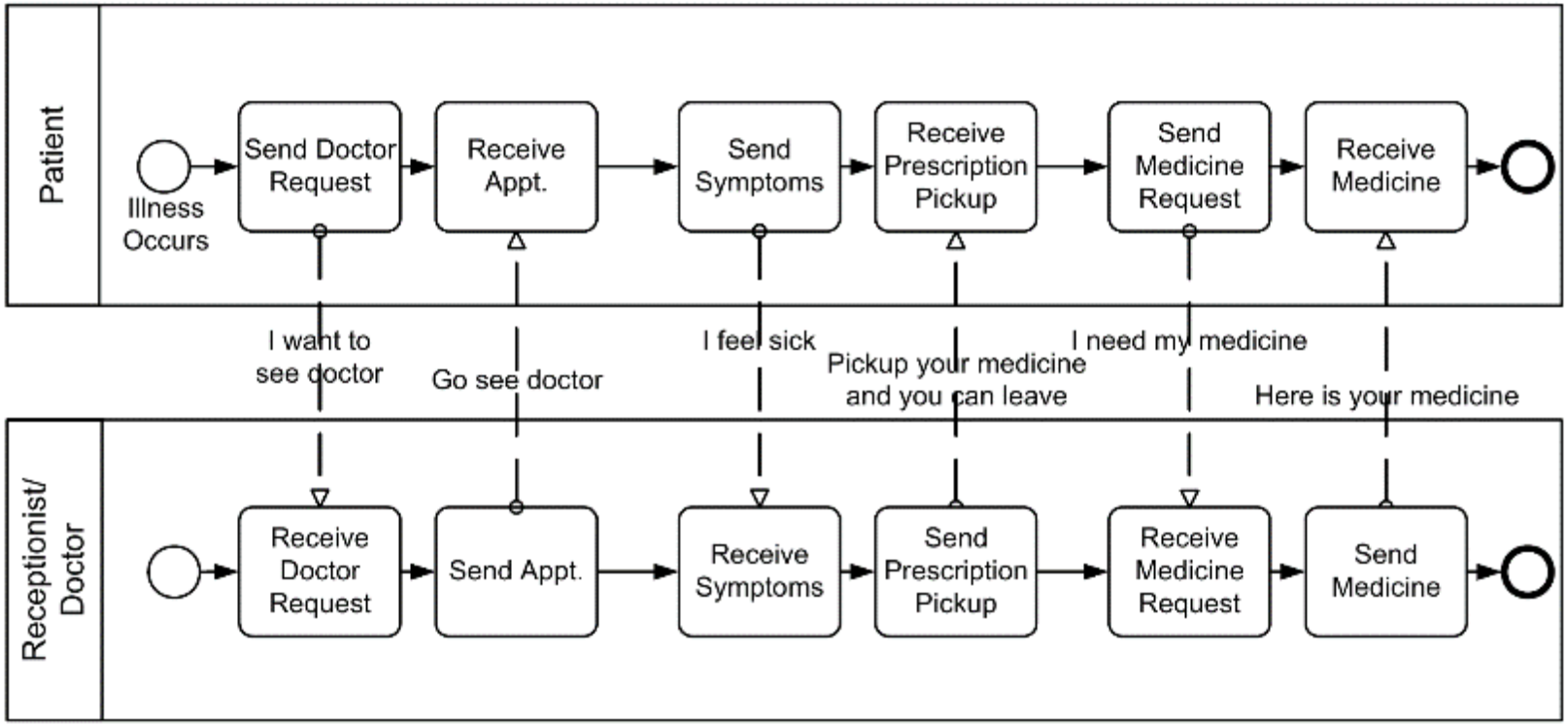


Image from "Business Process Model and Notation (BPMN)", http://www.omg.org/spec/BPMN/2.0

Specialized Task

- ***Send/Receive message***
  - Task that send/receive a message to/from an external participant (within the process)
- ***User activity***
  - Task performed by a human with the assistance of a software application
- ***Manual activity***
  - Task performed without the aid of any application
- ***Service calling***
  - Task that uses some sort of external service identified with an URI
- ***Script***
  - Task performed by a business process engine

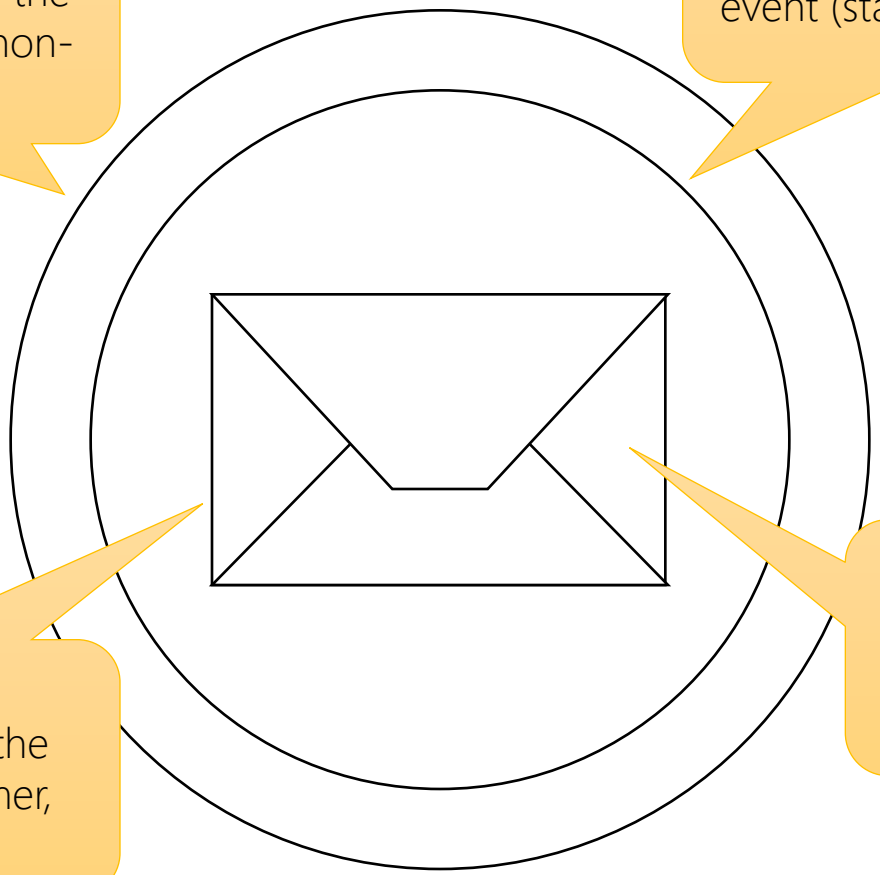
Event Definition

- The “Event” is the composition of
  - ***Position***
    - Start, Intermediate, End
  - ***Nature***
    - Catch, Throw
  - ***Impact***
    - Interrupting, Non-Interrupting
  - ***Type***
    - Message, Timer, etc.

Event Symbol

Dashed or solid circle
Define the impact of the event (interrupting, non-interrupting)

Single, double or filled outer circle
Define the position of the event (start, intermediate, end)



Internal marker
Define the type of the event (message, timer, etc.)

Filled or unfilled marker
Define the nature of event (throw, catch)

Event Position

- Start event
  - initial point of execution flow

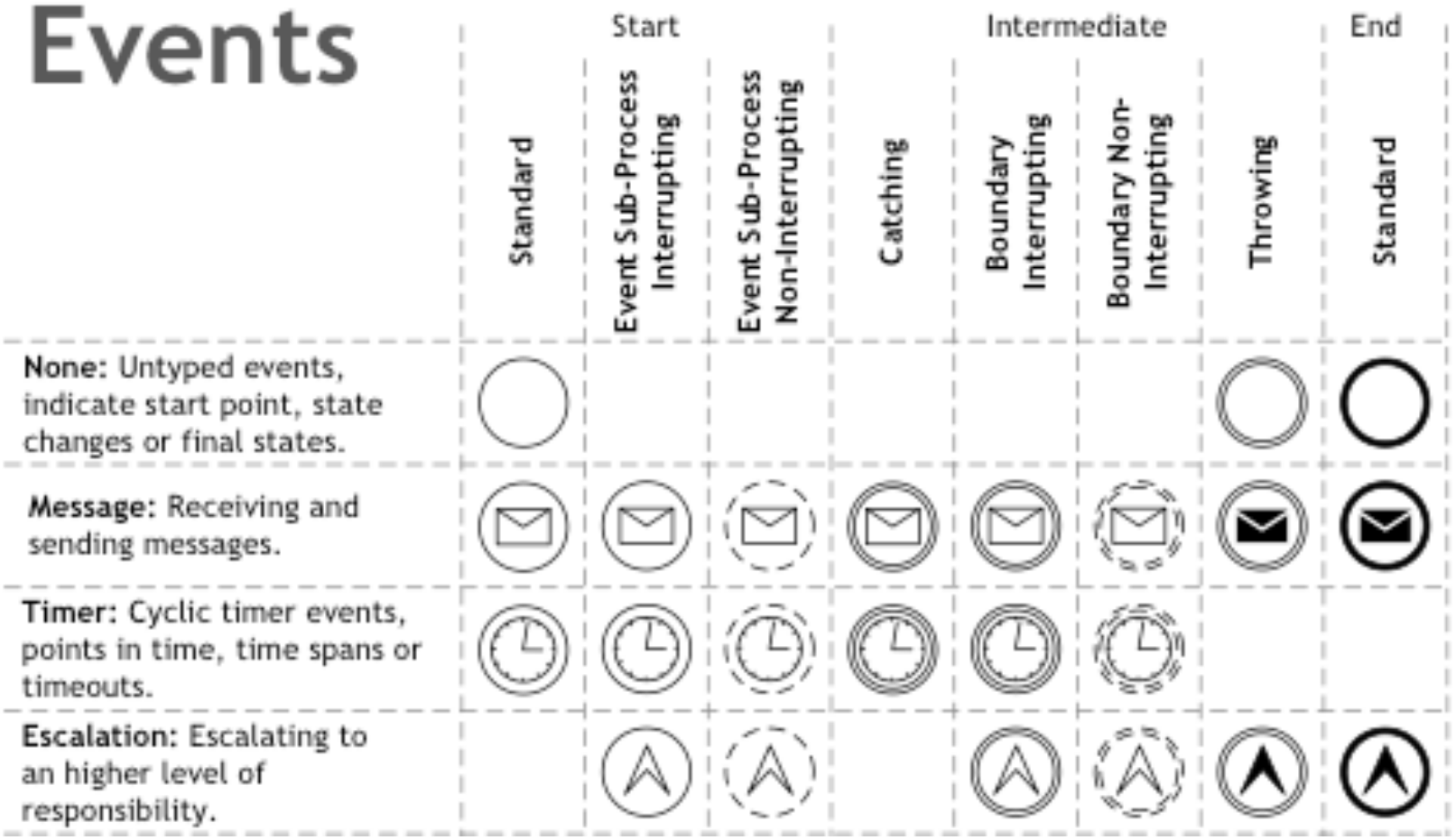
- Intermediate event
  - may occur between start and end of the execution flow

- End event
  - stop of the execution flow

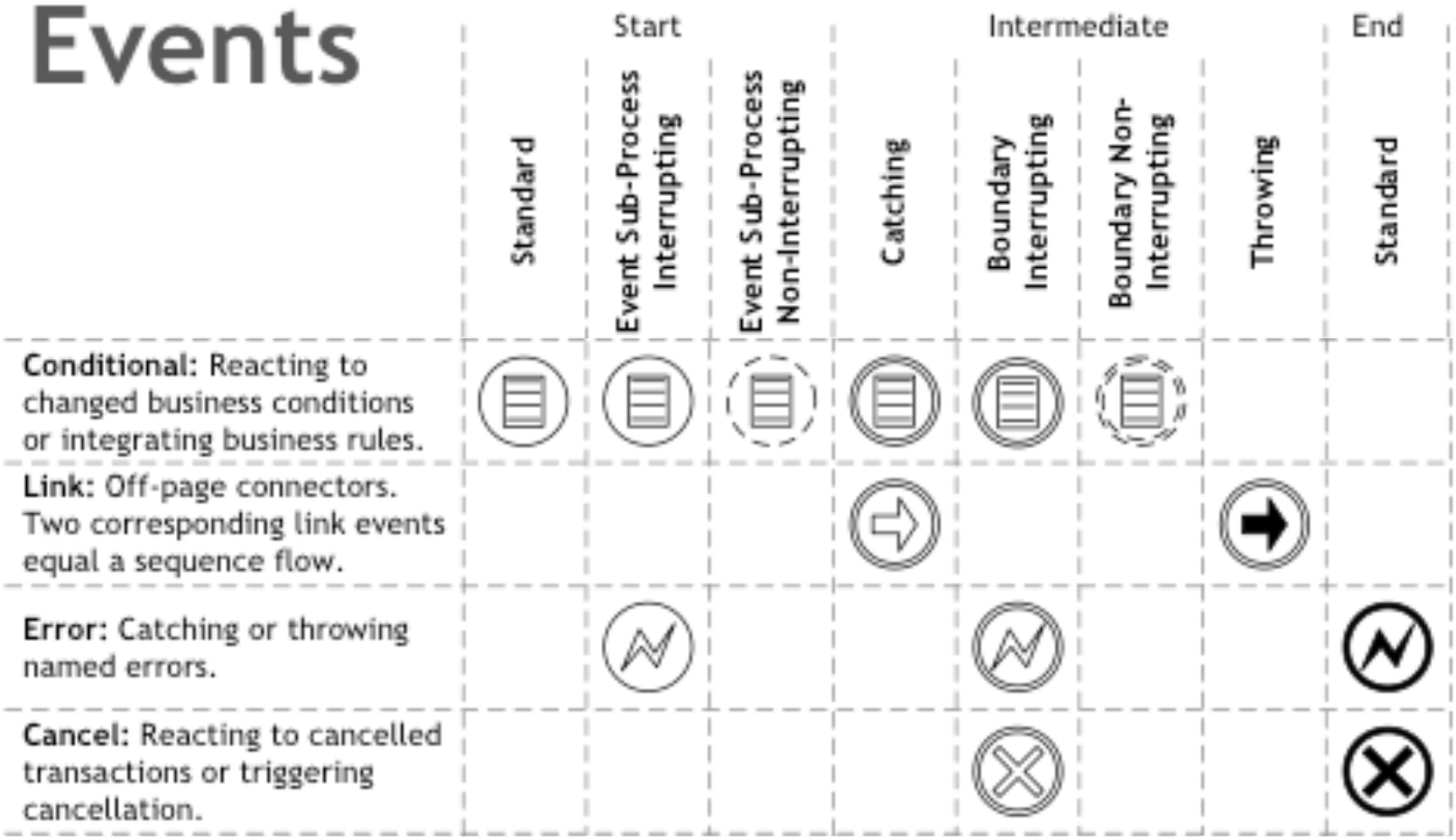
Event Nature

- ***Catch event***
  - All start events and some intermediate events
  - When the event is triggered, the token is generated
- ***Throw event***
  - All end events and some intermediate events are throwing events that may eventually be caught by another event
  - Typically, the trigger carries information from the scope where the throw event occurred into the scope of the catching events

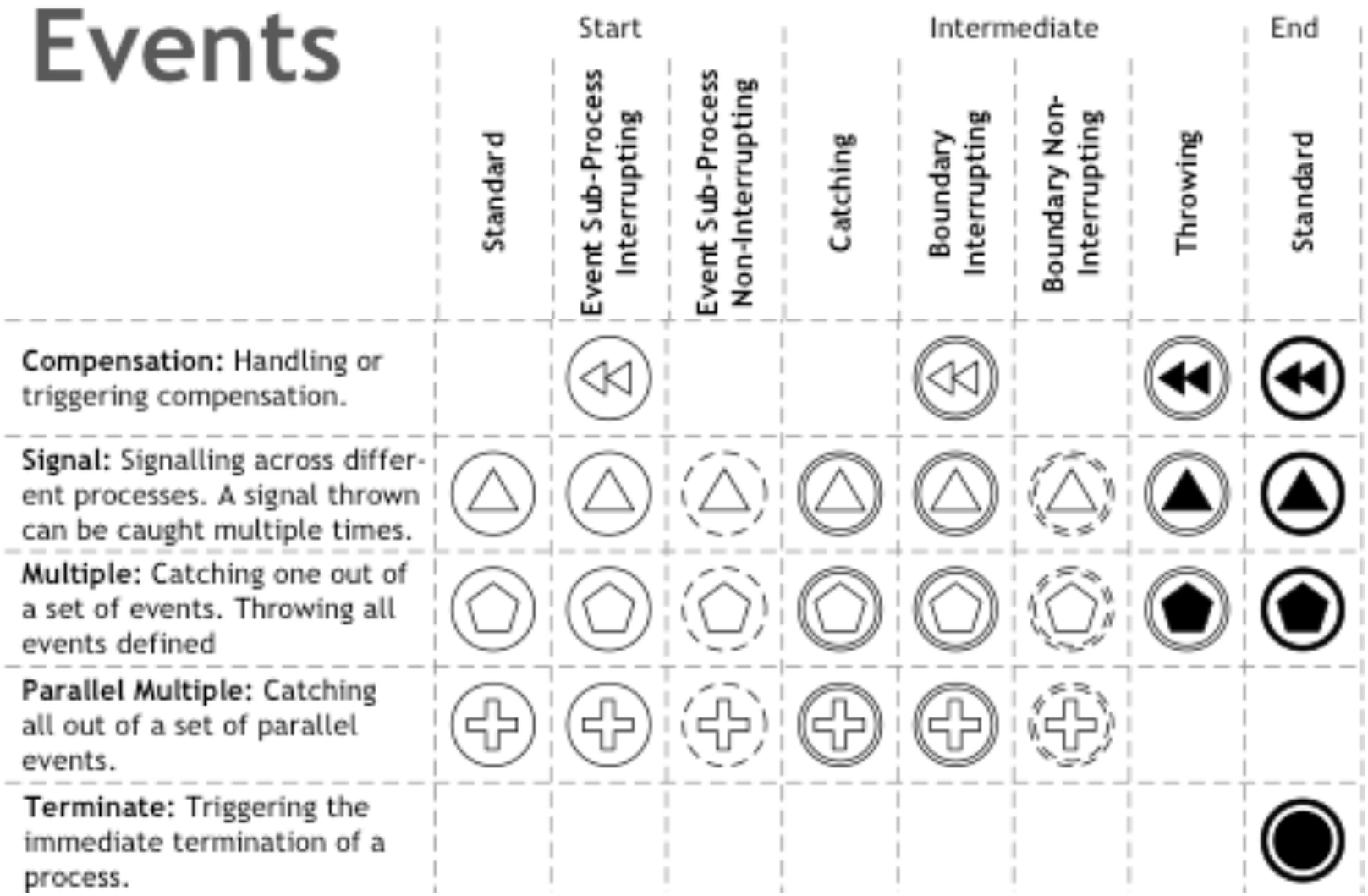
Events Definition



Events Definition



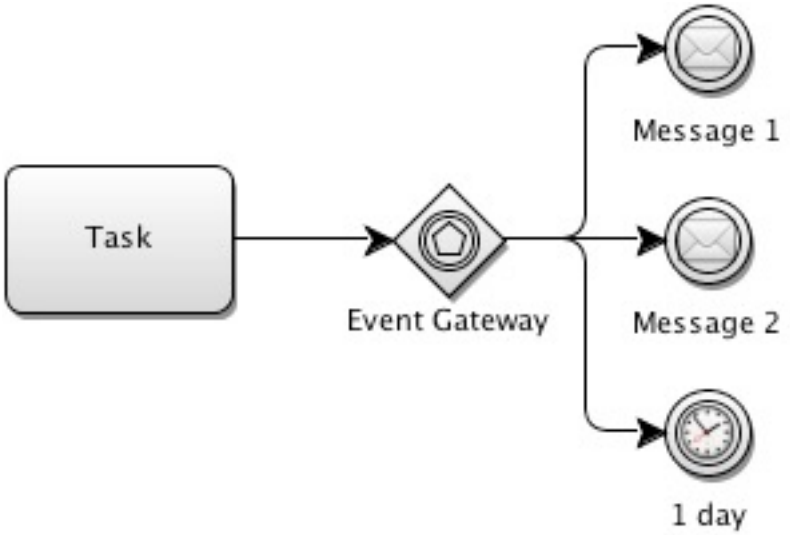
Events Definition



Event-Based Gateway

- Alternative paths are based on events that occur, rather than the evaluation of expressions
  - Usually the receipt of a message determines the path that will be taken
- When the first event is triggered, then the corresponding path is activated
  - All remaining paths will no longer be valid
  - The event gateway is thus a race condition where the first event that is triggered wins
- Event gateways can be used to start the process according to the event occurred

Event-Based Gateway

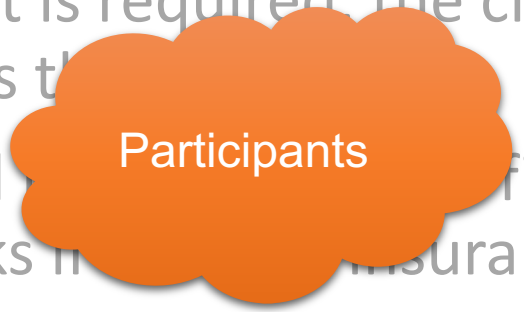


Demo - Shipment Process of a Hardware Retailer

- When goods are ready to be sent, the warehouse worker starts packaging the goods and a clerk decides if this is a normal postal or a special shipment.
- If a special shipment is required, the clerk selects a carrier and prepares the paperwork.
- Otherwise a normal post shipment is fine and in this case the clerk checks if an extra insurance is necessary.
- If the extra insurance is required, the logistics manager prepares that insurance. In any case, the clerk has to fill in a postal label for the shipment.

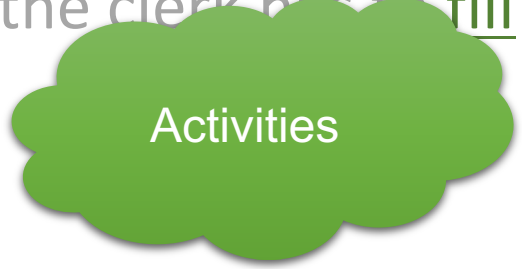
Demo - Shipment Process of a Hardware Retailer

- When goods are ready to be sent, the warehouse worker starts packaging the goods and a clerk decides if this is a normal postal or a special shipment.
- If a special shipment is required, the clerk selects a carrier and prepares t
- Otherwise a normal fine and in this case the clerk checks i surance is necessary.
- If the extra insurance is required, the logistics manager prepares that insurance. In any case, the clerk has to fill in a postal label for the shipment.



Demo - Shipment Process of a Hardware Retailer

- When goods are ready to be sent, the warehouse worker starts packaging the goods and a clerk decides if this is a normal postal or a special shipment.
- If a special shipment is required, the clerk selects a carrier and prepares the paperwork.
- Otherwise a normal post shipment is fine and in this case the clerk checks if an extra insurance is necessary.
- If the extra insurance is required, the logistics manager prepares that insurance. In any case, the clerk has to fill in a postal label for the shipment.



Demo - Shipment Process of a Hardware Retailer