CSCI427/927 Service-Oriented Software Engineering

Business Process Modelling and Management

Acknowledgement: Materials in these slides are adapted from "BPMN Modeling and Reference Guide: Understanding and Using BPMN" by Stephen A. White and Derek Miers

Business Processes

- A business process consists of **a set of activities**, performed by their relevant roles or collaborators, to intentionally **achieve a set of common business goals.**
- Business processes are the **core assets** of any enterprise, covering many aspects in industry such as design, engineering, manufacturing, purchasing, physical distribution, production management and supply chain management
  - E.g. production line of a car manufacturer, enrolment process at a university, insurance claim handling process, etc.

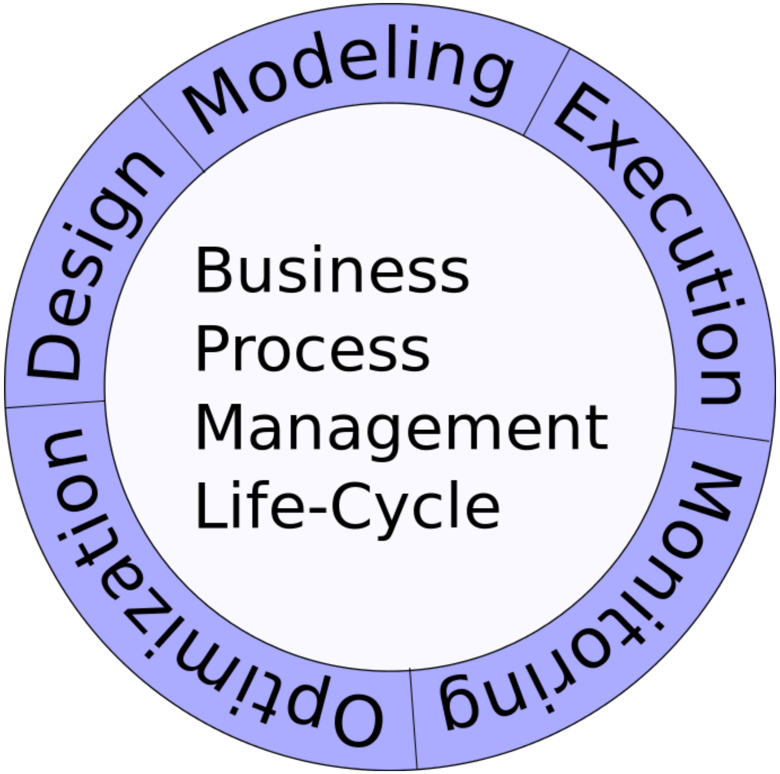
The importance for process modelling

- A number of questions arises when describing a process:
  - Which steps are really necessary?
  - Who should do them?
  - Should they be kept in house or outsourced?
  - How they should be done?
  - What capabilities are needed?
  - What results do we expect and how will they be monitored?

⇒ **Need a commonly agreed description** of the business process in question

The importance for process modelling

- Business process models are essential knowledge assets for an organization manage its business processes in terms of:
  - documenting and implementing procedures,
  - control their execution,
  - analyse their performance,
  - and improve them (basis for process improvement, understanding, communication, execution)



Business process models

- Organizations committed to long-term business process management (BPM) may have repositories of hundreds or even thousands of business process models.
  - IBM BIT Process Library has 735 process models
  - The SAP Reference Model contains 604 process models,
  - 6,000+ process models in Suncorp's process model repository for insurance.

Business process modelling

- Business processes involve:
  - Multiple actors (people, business units,…)
  - Concurrent activities
  - Explicit synchronization points
    - E.g. some task cannot start until several other concurrent tasks are complete
  - End-to-end flow of activities
- Choice of modelling language:
  - **Business Process Modelling Notation (BPMN)**
    - New (emerging) standard
  - UML Activity diagrams
  - Petri Nets, etc.

Business Process Modelling Notation BPMN

- *"The primary goal of BPMN is to provide a notation that is **readily understandable by all business users**, from the **business analysts** that create the initial draft of the processes, to the **technical developers** responsible for implementing the technology that will perform these processes, and finally to the **business people** who will manage and monitor those processes."*
- *"The idea is to create a **standardized bridge** for the gap between the business process **design** and process **implementation**"* [BPMN 2.0 spec.]

BPMN elements

- The four basic categories of elements are:
    - Flow Objects
    - Connecting Objects
    - Swimlanes
    - Artifacts

Flow objects

Source: Stephen A. White, Introduction to BPMN, May 2004

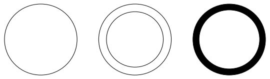
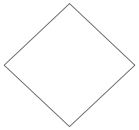
Event	An *Event* is represented by a circle and is something that "happens" during the course of a business process. These Events affect the flow of the process and usually have a cause (trigger) or an impact (result). Events are circles with open centers to allow internal markers to differentiate different triggers or results. There are three types of Events, based on when they affect the flow: *Start*, *Intermediate*, and *End* (see the figures to the right, respectively).	
Activity	An *Activity* is represented by a rounded-corner rectangle (see the figure to the right) and is a generic term for work that company performs. An Activity can be atomic or non-atomic (compound). The types of Activities are: *Task* and *Sub-Process*. The Sub-Process is distinguished by a small plus sign in the bottom center of the shape.	
Gateway	A *Gateway* is represented by the familiar diamond shape (see the figure to the right) and is used to control the divergence and convergence of Sequence Flow. Thus, it will determine traditional decisions, as well as the forking, merging, and joining of paths. Internal Markers will indicate the type of behavior control.	

Table 1: Core BPD Flow Objects

Connecting objects

Source: Stephen A. White, Introduction to BPMN, May 2004

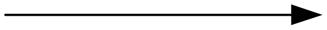
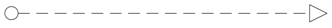
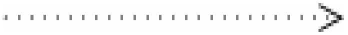
Sequence Flow	A *Sequence Flow* is represented by a solid line with a solid arrowhead (see the figure to the right) and is used to show the order (the sequence) that activities will be performed in a Process. Note that the term "control flow" is generally not used in BPMN.	
Message Flow	A *Message Flow* is represented by a dashed line with an open arrowhead (see the figure to the right) and is used to show the flow of messages between two separate Process Participants (business entities or business roles) that send and receive them. In BPMN, two separate Pools in the Diagram will represent the two Participants.	
Association	An *Association* is represented by a dotted line with a line arrowhead (see the figure to the right) and is used to associate data, text, and other Artifacts with flow objects. Associations are used to show the inputs and outputs of activities.	

Table 2: BPD Connecting Elements

Swimlanes

Source: Stephen A. White, Introduction to BPMN, May 2004

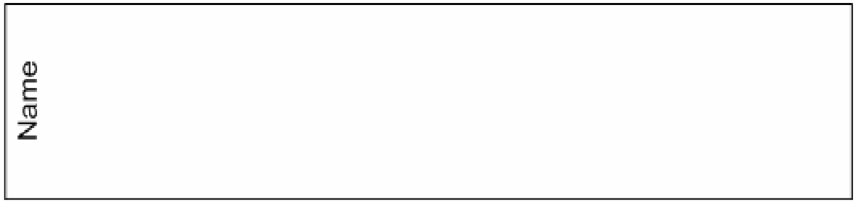
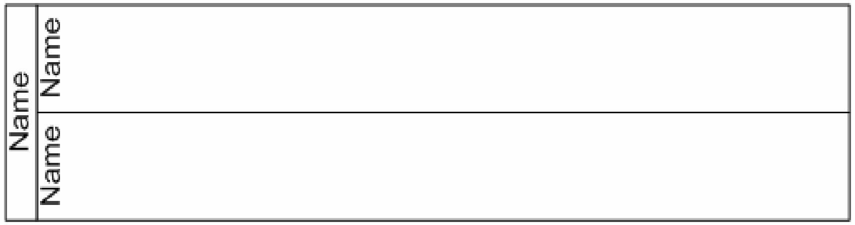
Pool	A *Pool* represents a Participant in a Process. It is also acts as a graphical container for partitioning a set of activities from other Pools (see the figure to the right), usually in the context of B2B situations.	
Lane	A *Lane* is a sub-partition within a Pool and will extend the entire length of the Pool, either vertically or horizontally (see the figure to the right). Lanes are used to organize and categorize activities.	

Table 3: BPD Swimlane Objects

Artifacts

Source: Stephen A. White, Introduction to BPMN, May 2004

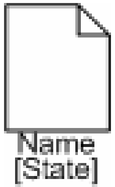
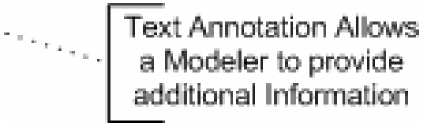
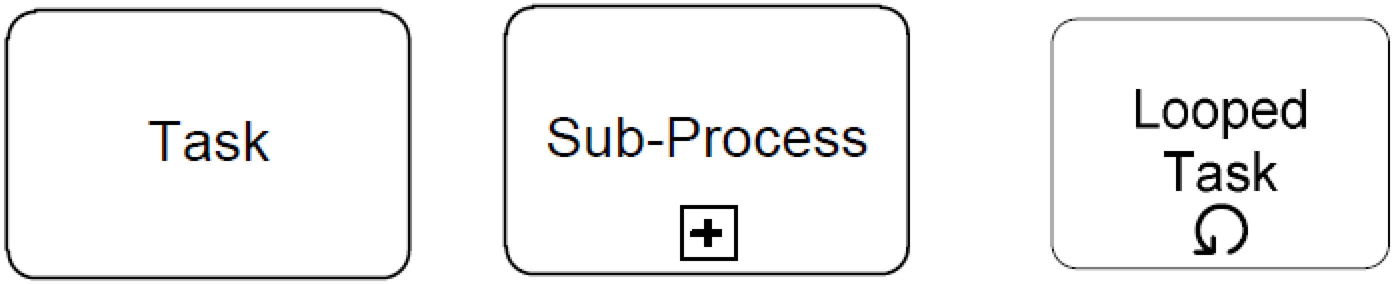
Data Object	*Data Objects* are a mechanism to show how data is required or produced by activities. They are connected to activities through Associations.	
Group	A *Group* is represented by a rounded corner rectangle drawn with a dashed line (see the figure to the right). The grouping can be used for documentation or analysis purposes, but does not affect the Sequence Flow.	
Annotation	*Annotations* are a mechanism for a modeler to provide additional text information for the reader of a BPMN Diagram (see the figure to the right).	 Text Annotation Allows a Modeler to provide additional Information

Table 4: BPD Artifact Elements

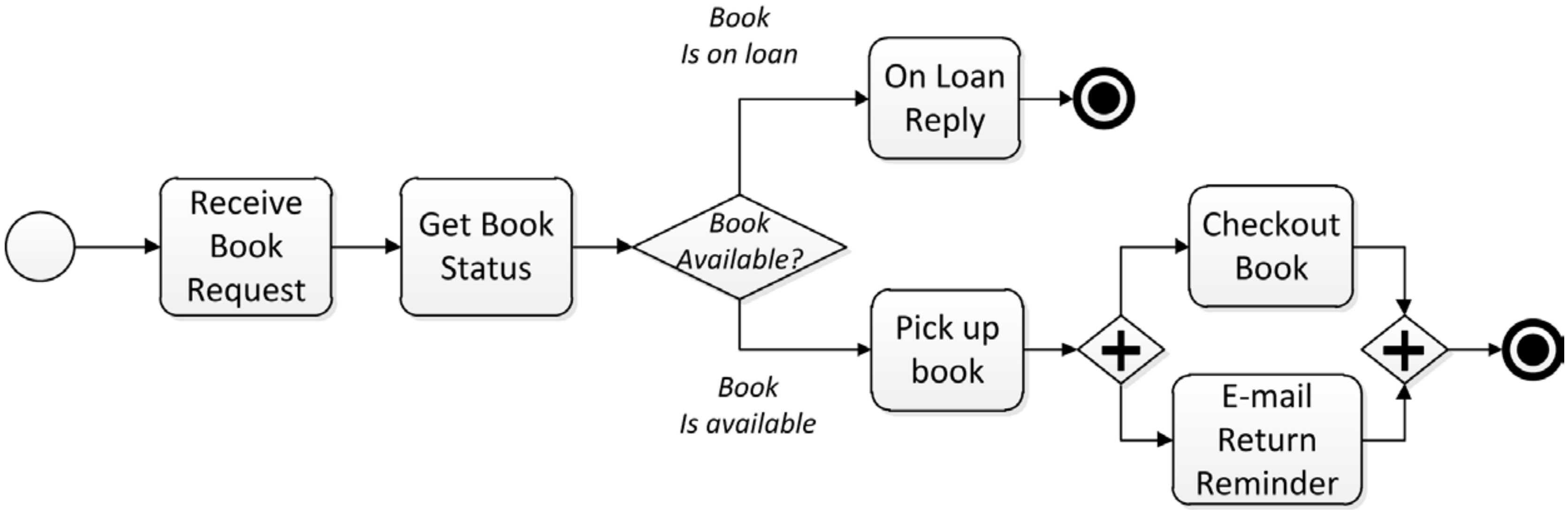
Activities

- An activity is a work that is performed within a business process
- An activity can take some time to perform and involves one or more resources
- An activity can be **atomic (i.e. Task)** or **compound (i.e. Sub-Process)**.
- An activity can performed once or can have internally defined loops



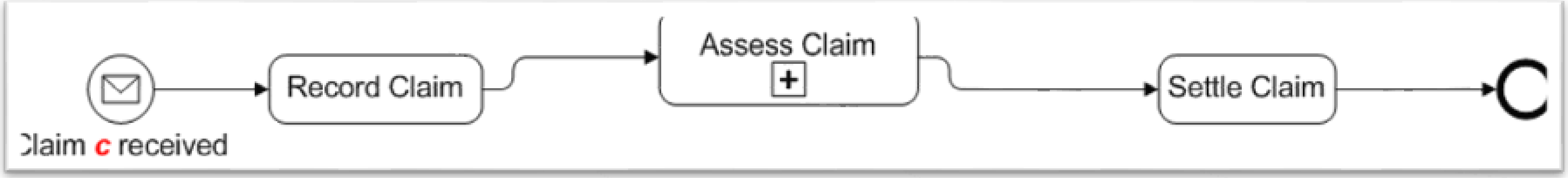
Tasks

- A Task is an atomic activity that is included within a process.
- A Task is used when work in the Process is not broken down to a finer level of detail



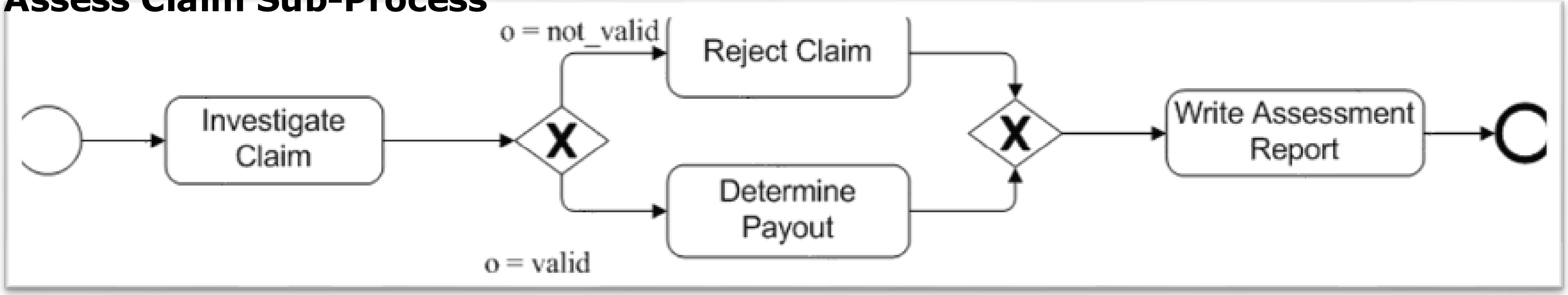
Sub-Process

- 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

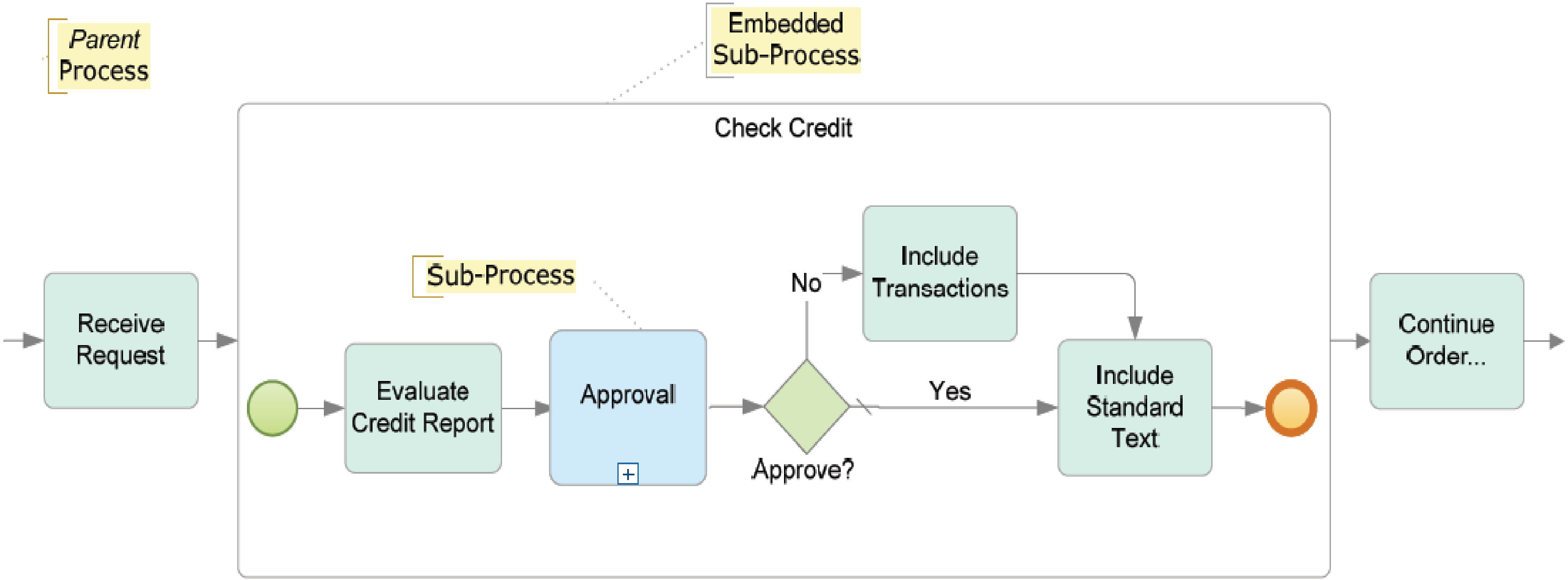


Expanded sub-process

Assess Claim Sub-Process

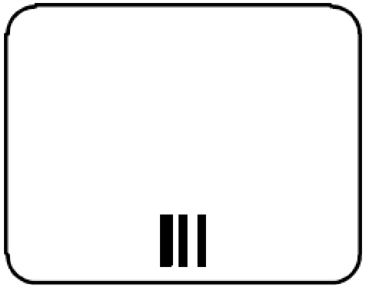


Sub-Process (cont.)

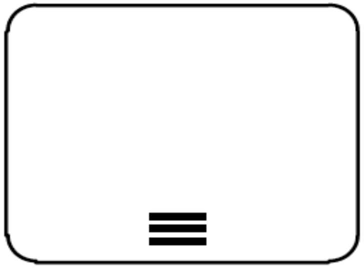


Multi-instance Activities

- An activity may be performed many times with different data sets.
  - E.g. When a major corporation checks financial results of **all** subsidiaries, it needs to carry out this many times (each time for each subsidiary with different data).
- Multi-instance Activity (or For Each) supports this behaviour



Parallel

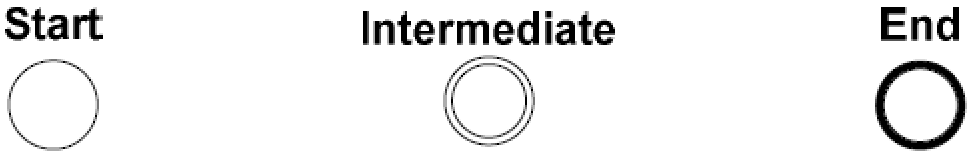


Sequential

loop can be also seq

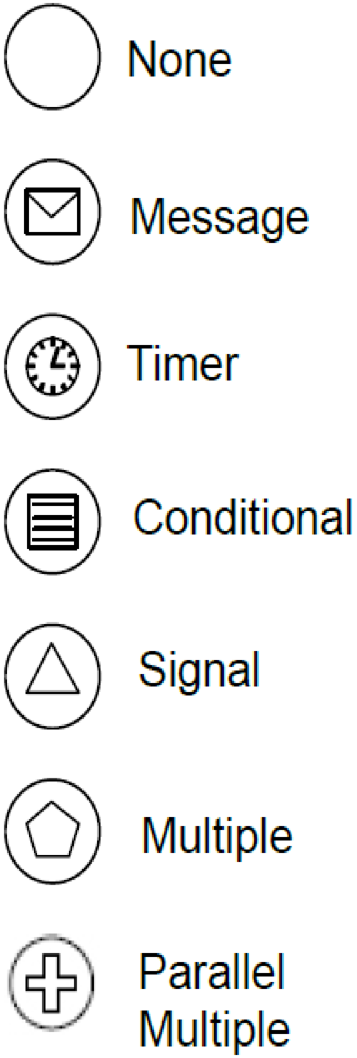
Events

- An Event is something that “happens” during the course of a business process
- An Event may affect the flow of the Process and usually have a **trigger** or a **result**.
- They can **start**, **delay**, **interrupt**, or **end** the flow of the process.



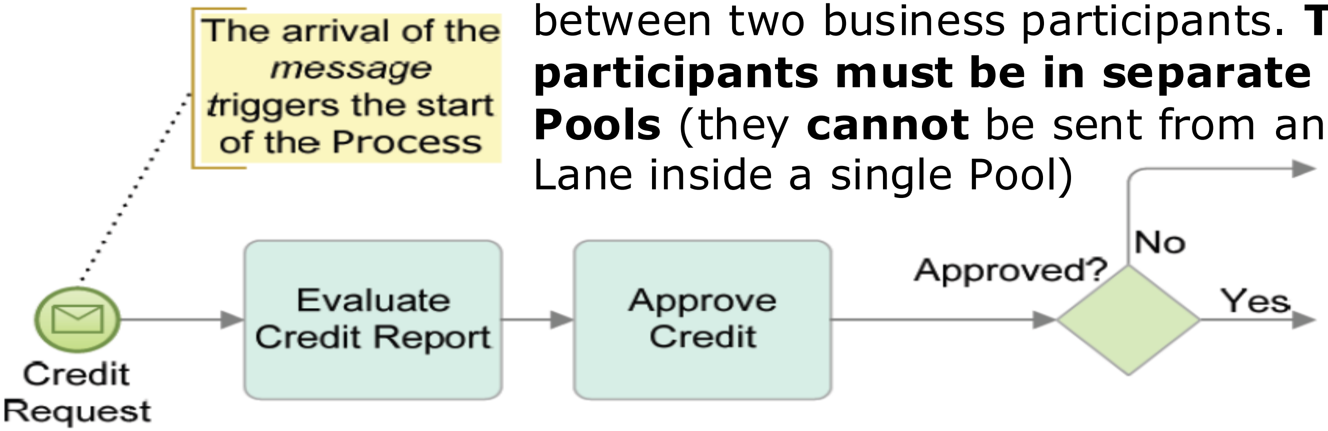
Start Events

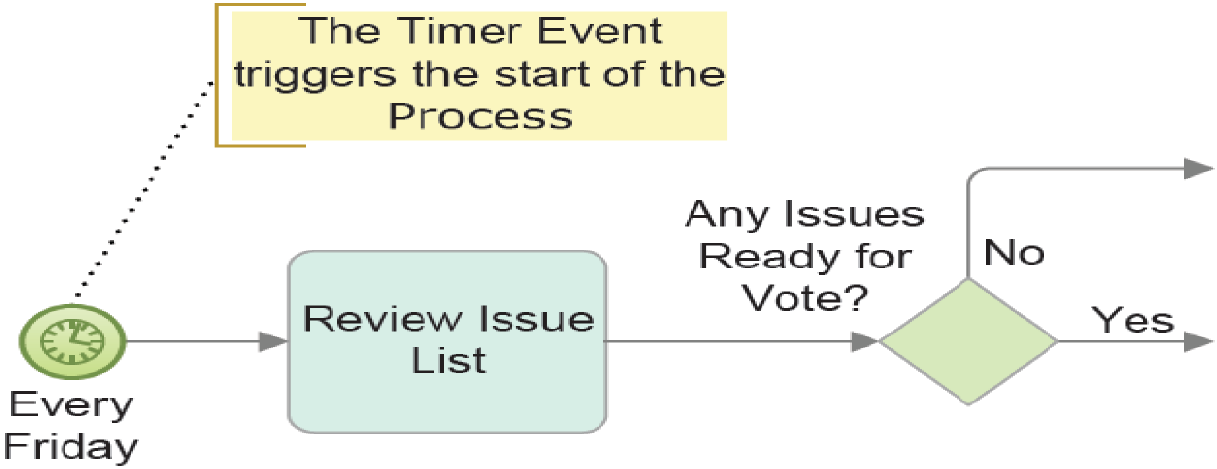
- A Start Event shows where a Process will begin
- Different types of Start Events indicate different situations that can **trigger** the start of a Process.
- A Start Event **can only have outgoing Sequence Flows**.
- Trigger-based Start Events can only exist in top-level Processes, i.e. Not used in Sub-Processes.



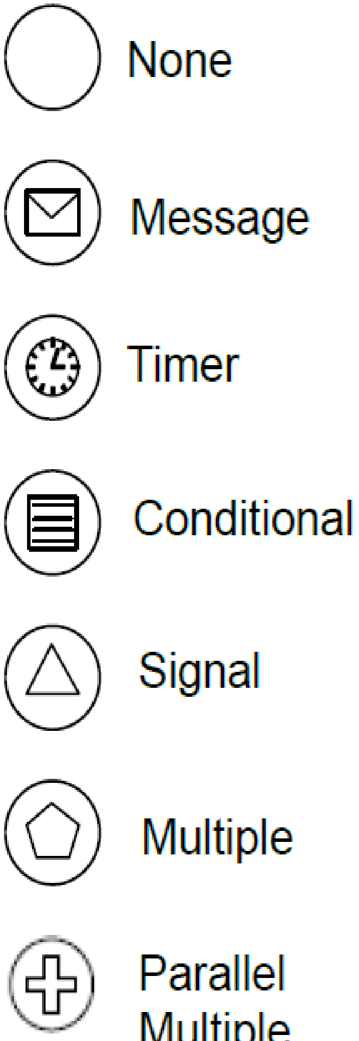
Start Events (cont.)

A message is a direct communication between two business participants. **These participants must be in separate Pools** (they **cannot** be sent from another Lane inside a single Pool)

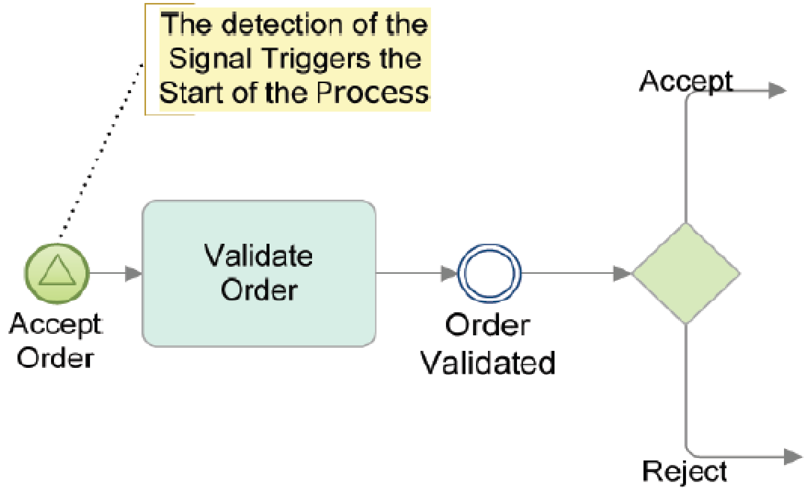



The Timer Event triggers the start of the Process
Every Friday
Review Issue List
Any Issues Ready for Vote?
No
Yes

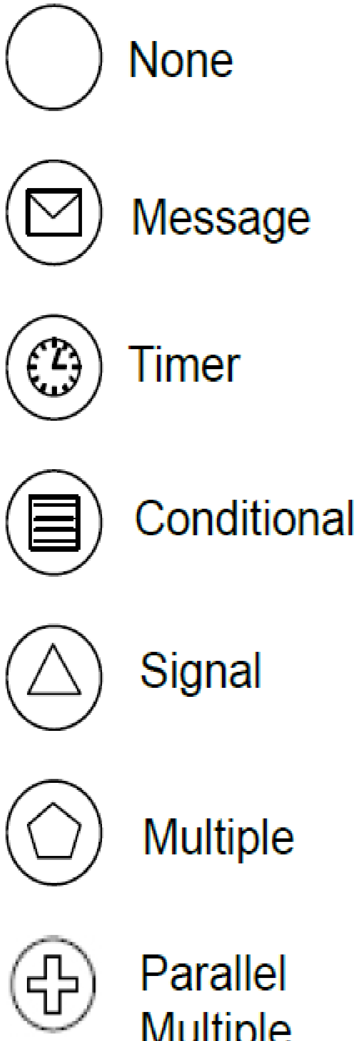
This could be a specific date and time (e.g., January 1, 2009 at 8am) or a recurring time (e.g., every Monday at 8am).

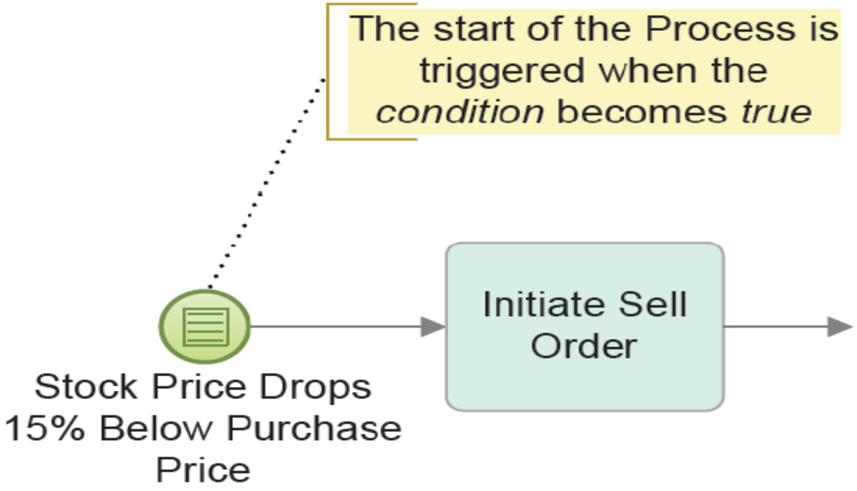


Start Events (cont.)


The detection of the Signal Triggers the Start of the Process
Accept Order
Validate Order
Order Validated
Accept
Reject

Signals have no specific target or recipient - i.e. all Processes and participants can see the signal and it is up to each of them to decide whether or not to react



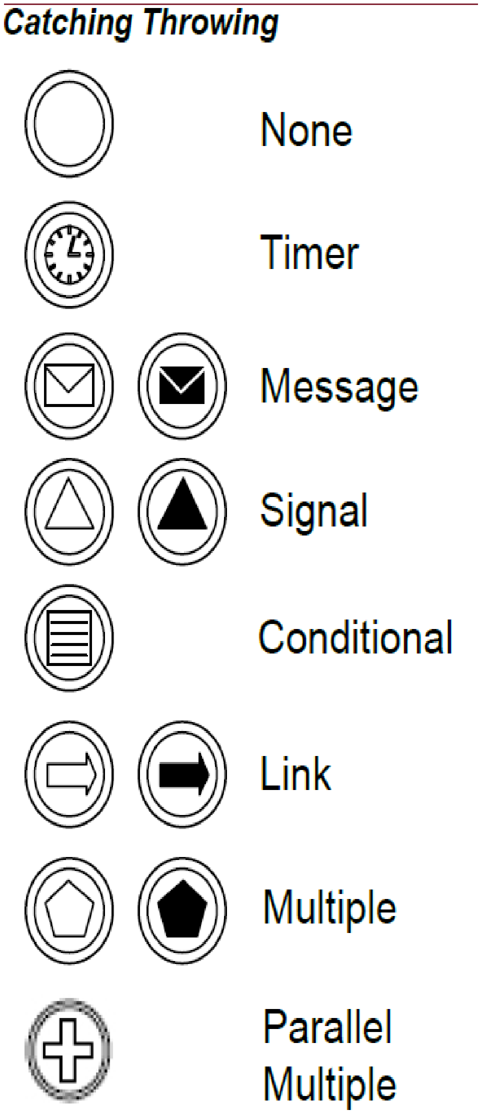

The start of the Process is triggered when the *condition* becomes *true*
Stock Price Drops 15% Below Purchase Price
Initiate Sell Order

A condition is used to define the details of the change in data that is expected.

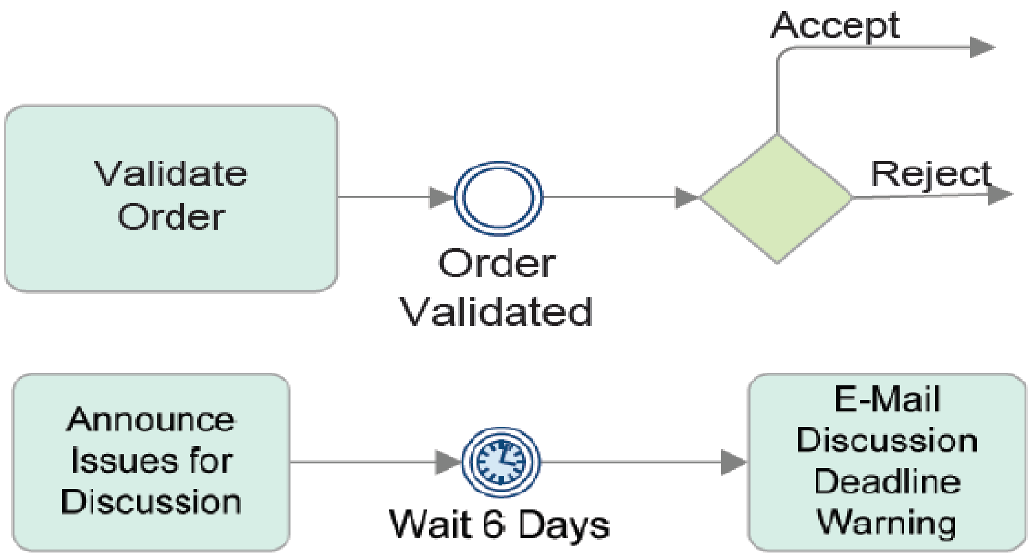
The condition for the Event must become false and then true again before the Event can be triggered again.

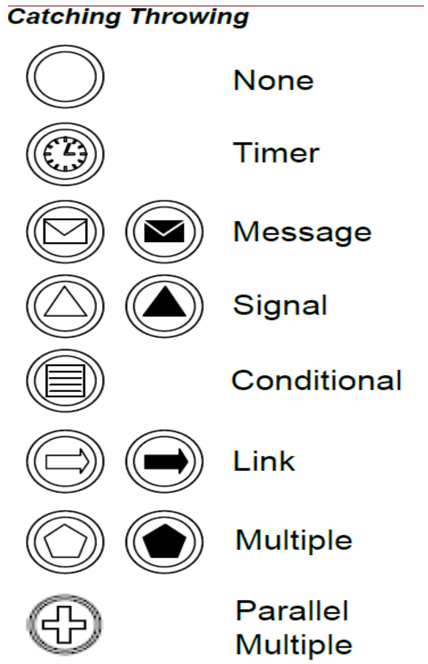
Intermediate Events

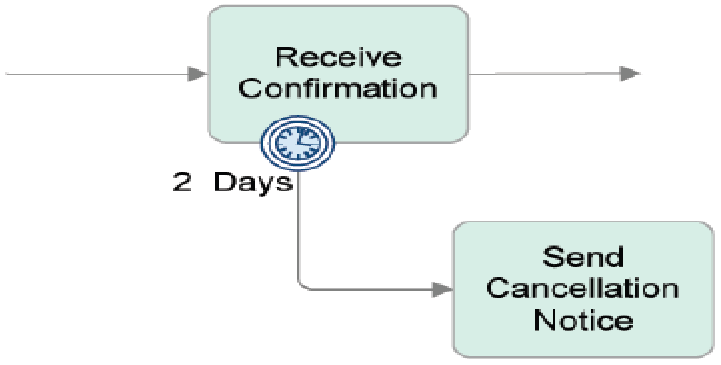
- An Intermediate Event indicates where something happens/occurs after a Process has **started and before** it has ended.
- They may also **interrupt** the normal process of an Activity.
- Each type of Intermediate Events can either **throw** or **catch** the event.
- A catching Intermediate Event **waits for something to happen** (i.e. wait for the circumstance defined on the trigger).
- A throwing Intermediate Event **immediately fires** (effectively creating the circumstance defined on the trigger).



Intermediate Events (cont.) Catching

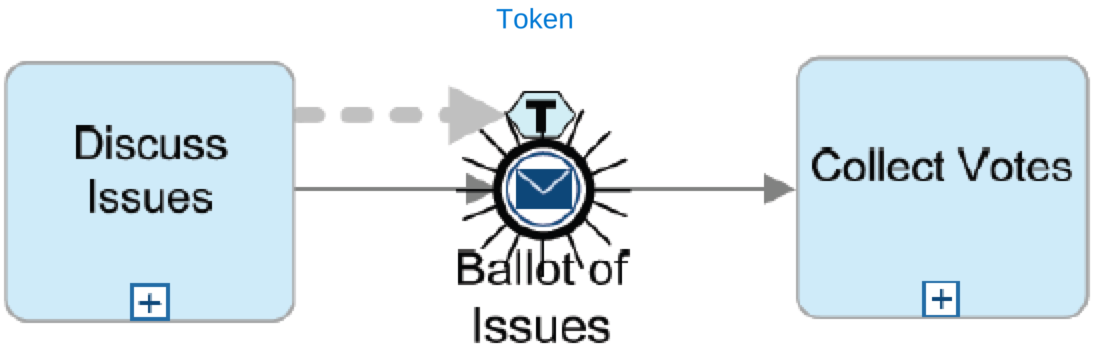

Accept
Validate Order
Order Validated
Reject
Announce Issues for Discussion
Wait 6 Days
E-Mail Discussion Deadline Warning





When the Activity starts, so does the timer. If the Activity finishes first, then it completes normally and the Process continues normally. If the timer goes off before the Activity is completed, the Activity is **immediately interrupted** and the Process continues down the Sequence Flow from the Timer Intermediate Event.

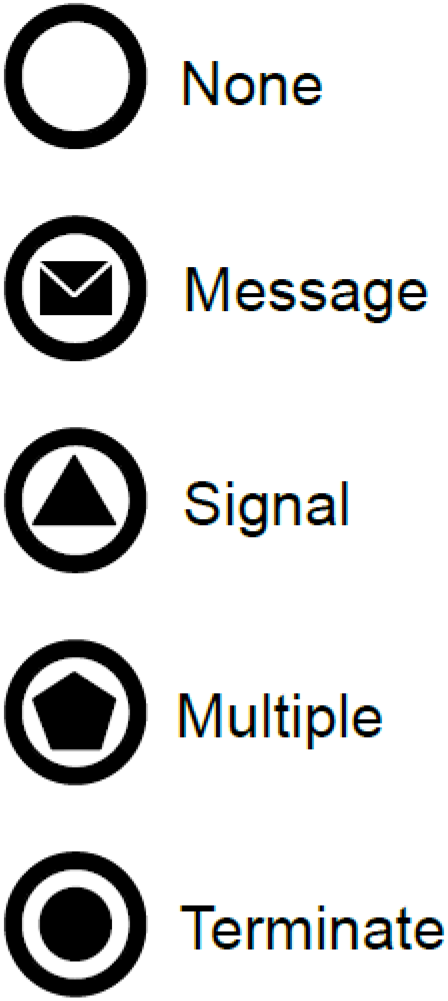
Intermediate Events (cont.) Throwing



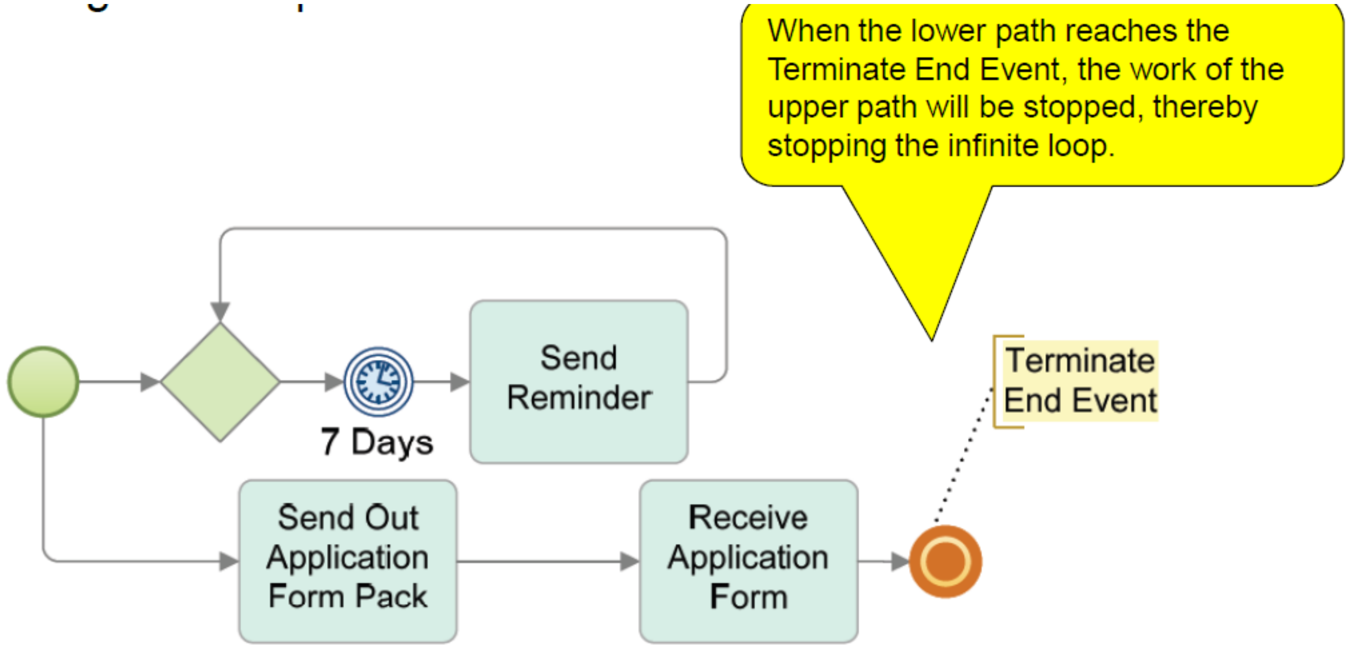
A Throwing Intermediate Event **can not be attached** to the boundary of an Activity.

End Events

- Different types of End Events indicate different categories of **results** for the Process
  - A result is something that occurs at the end of a particular path of the Process (e.g. message sent, signal broadcasted).
  - All End Events are *throw results*.
- Only incoming Sequence Flow is permitted, (i.e. Sequence Flow cannot leave from an End Event).
- A **None** End Event is always used to mark the end of Sub-Processes (moving from one level up to the next).



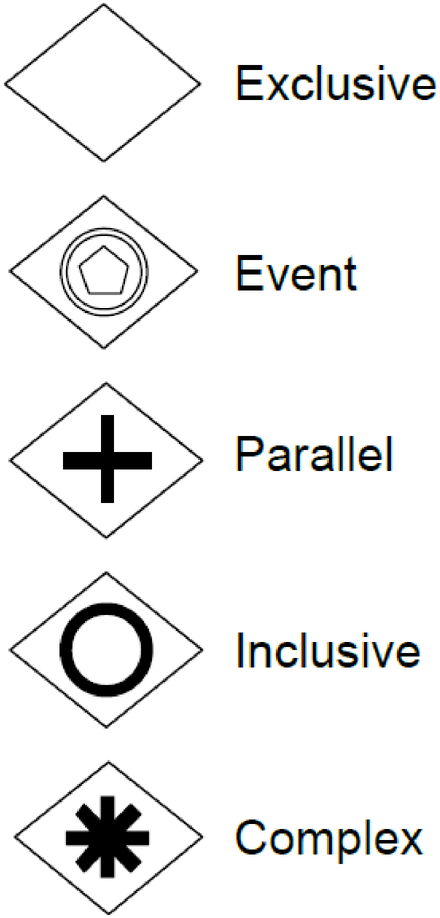
End Events



When the lower path reaches the Terminate End Event, the work of the upper path will be stopped, thereby stopping the infinite loop.

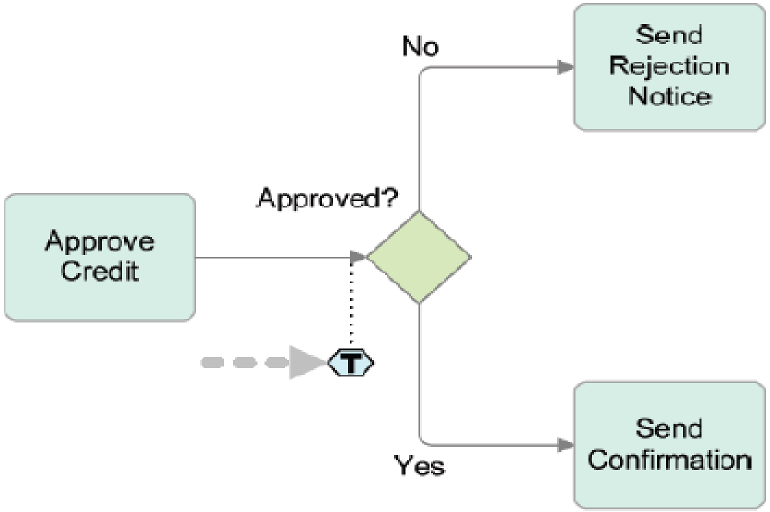
Gateways

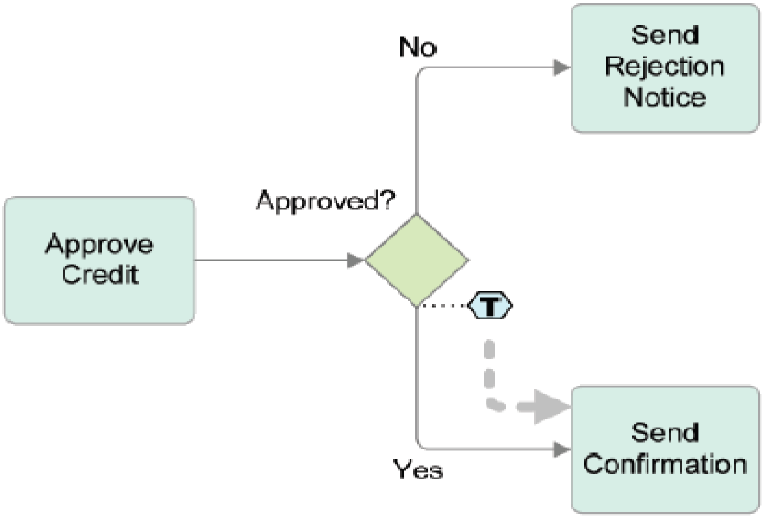
- Gateways control how the Process diverges or converges.
  - Split and merge the flow of a Process
- There are different types of Gateways
  - The type (splitting and merging) for a single Gateway must be matched, e.g. a Gateway cannot be Parallel on the input side, and Exclusive on the output side.



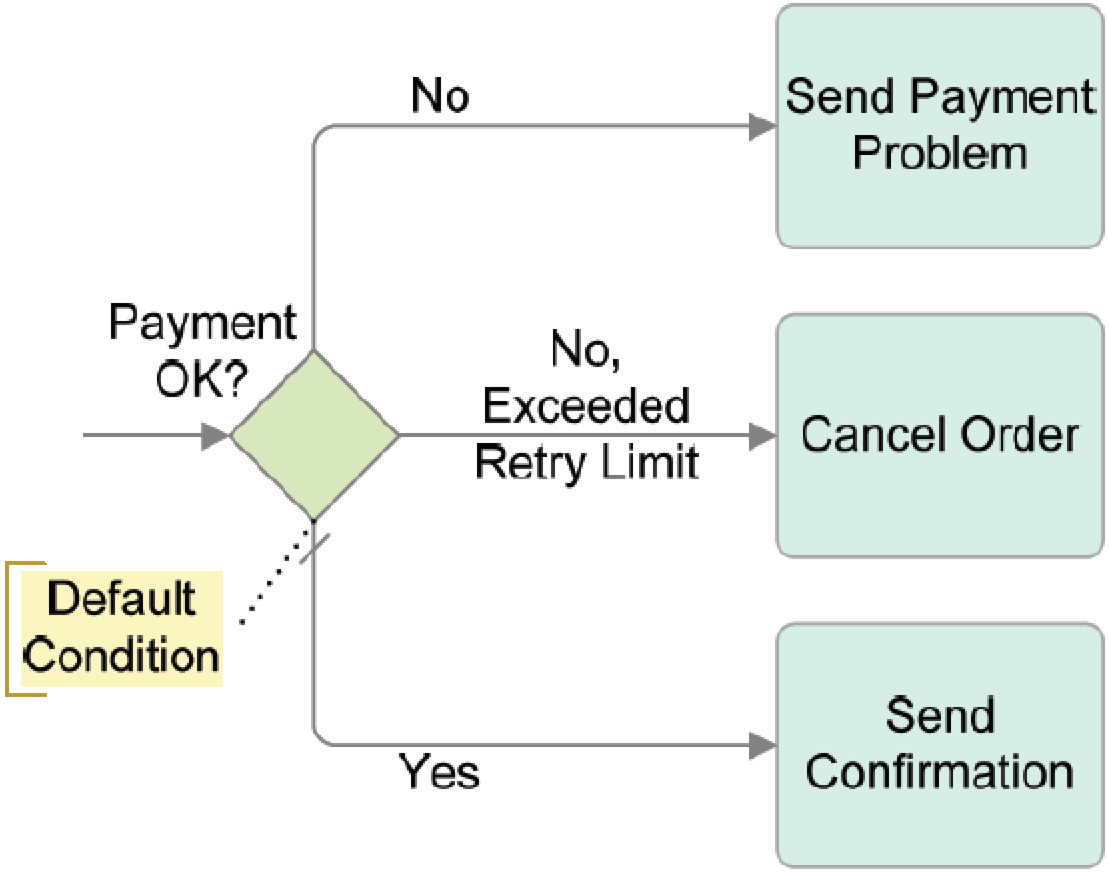
Exclusive Gateways Splitting Behaviour

- An Exclusive Gateway splits the flow when it has 2 or more outgoing paths.
- There is a condition on each of the outgoing paths. One of those conditions must evaluate to true.





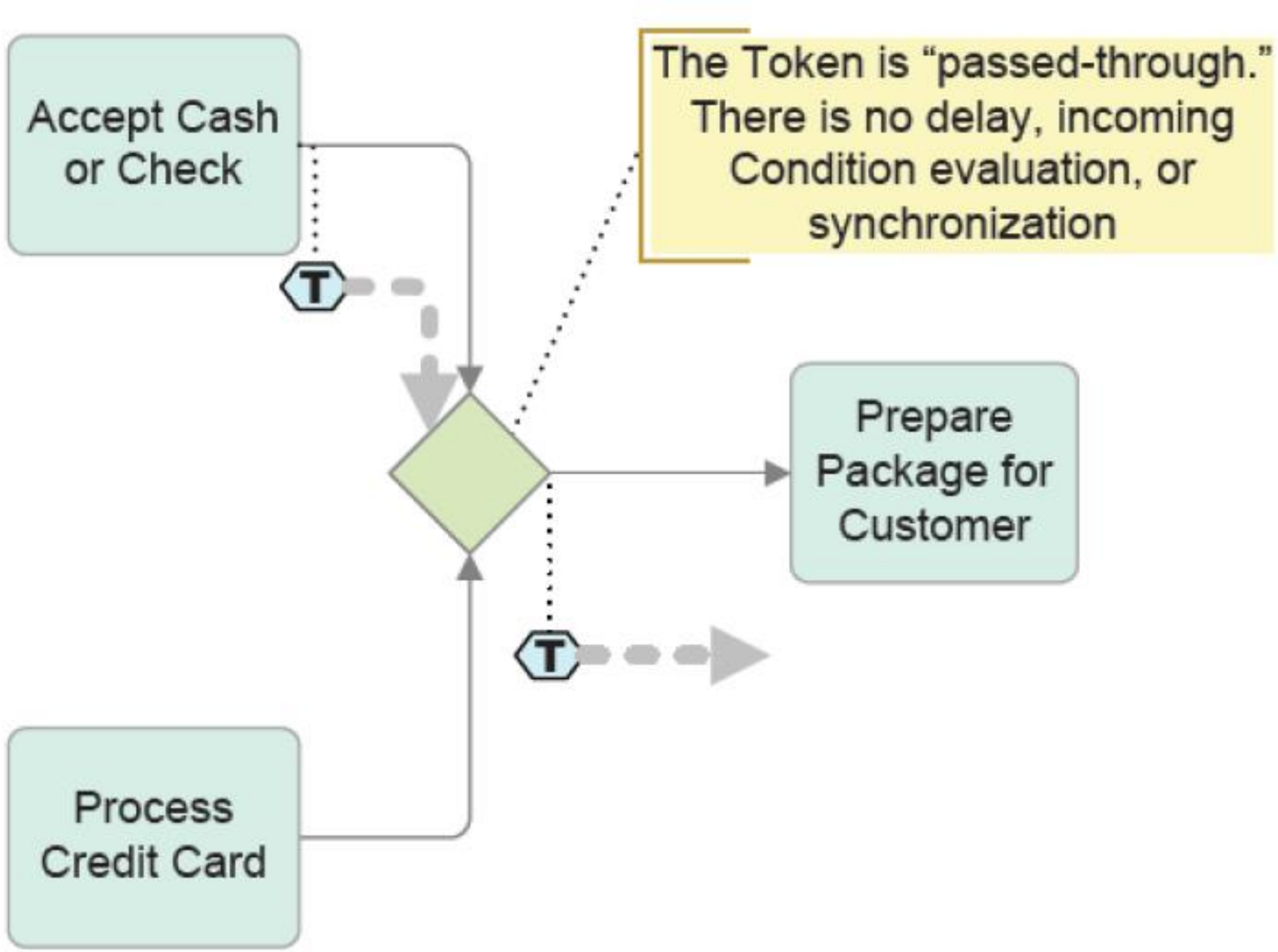
Exclusive Gateways
Default Condition



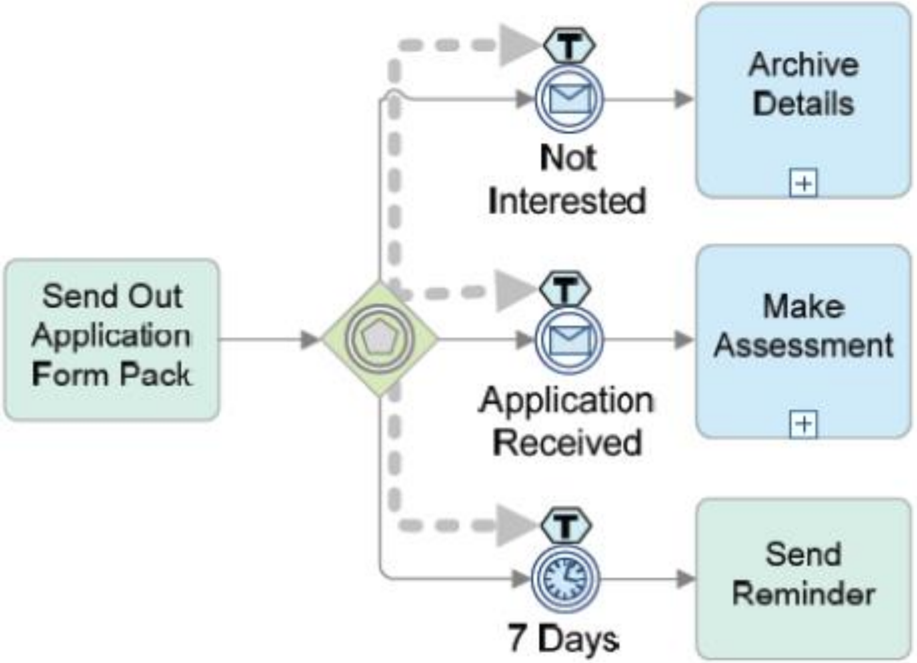
One way for the modeler to ensure that the Process does not get stuck at an Exclusive Gateway is to use a *Default Condition* for one of the outgoing Sequence Flow.

The Default is chosen if all the other Sequence Flow conditions turn out to be false.

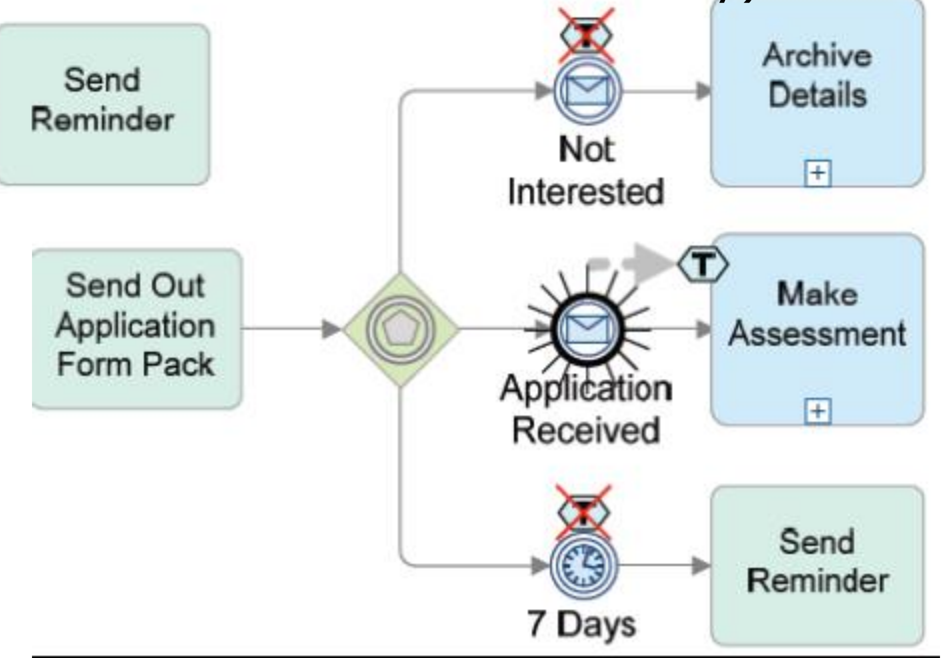
Exclusive Gateways Merging behaviour


Accept Cash or Check
T
The Token is "passed-through." There is no delay, incoming Condition evaluation, or synchronization
Prepare Package for Customer
T
Process Credit Card

Event-based Exclusive Gateways

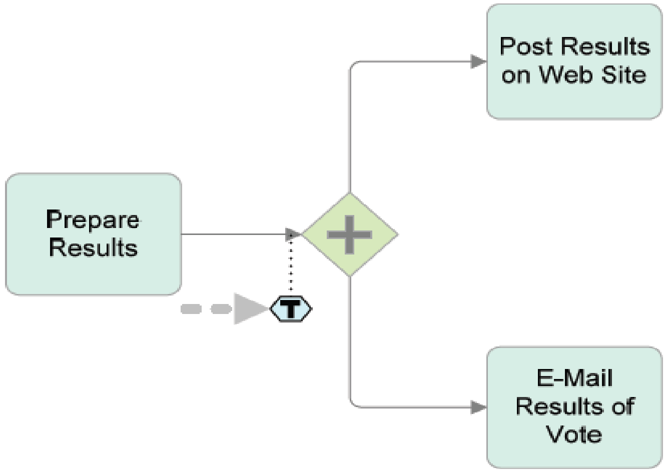


Event-Based Exclusive Gateways represent an alternative branching point where the decision is based on **two or more Events that might occur,** rather than data-oriented conditions (as in an Exclusive Gateway).

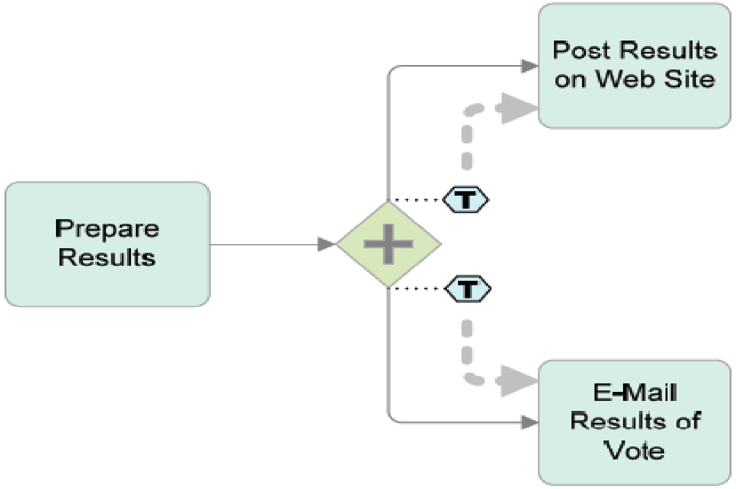


Parallel Gateways Splitting

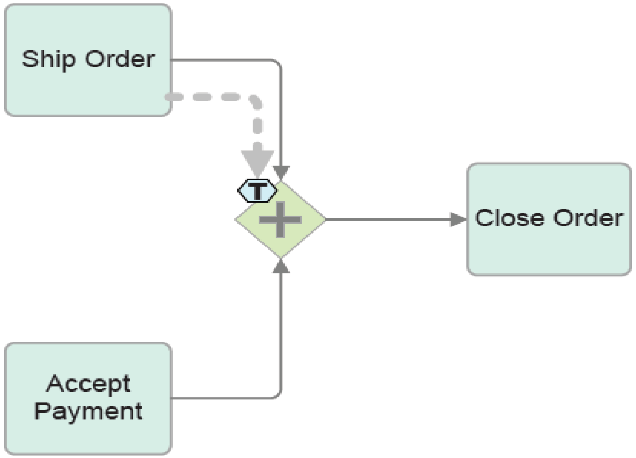
- There is no evaluation to any conditions at the Parallel Gateways

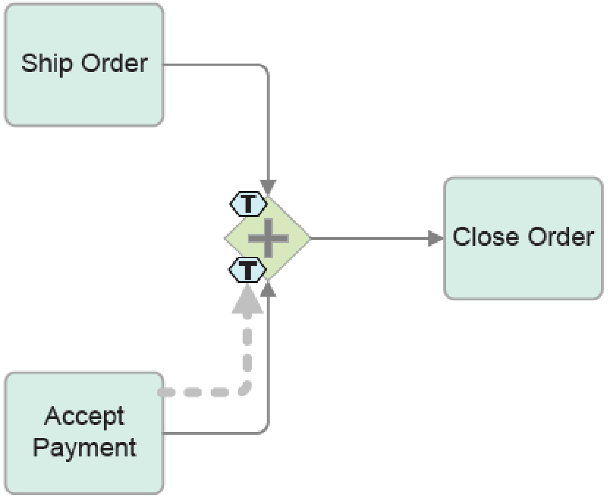


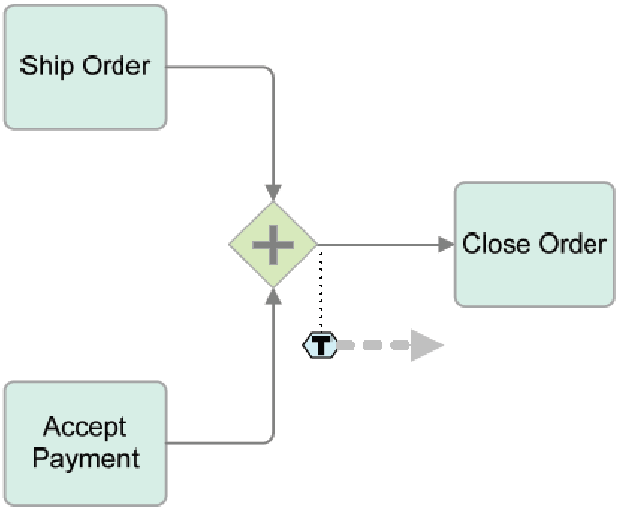
- The Parallel Gateway will create parallel paths.



Parallel Gateways Merging



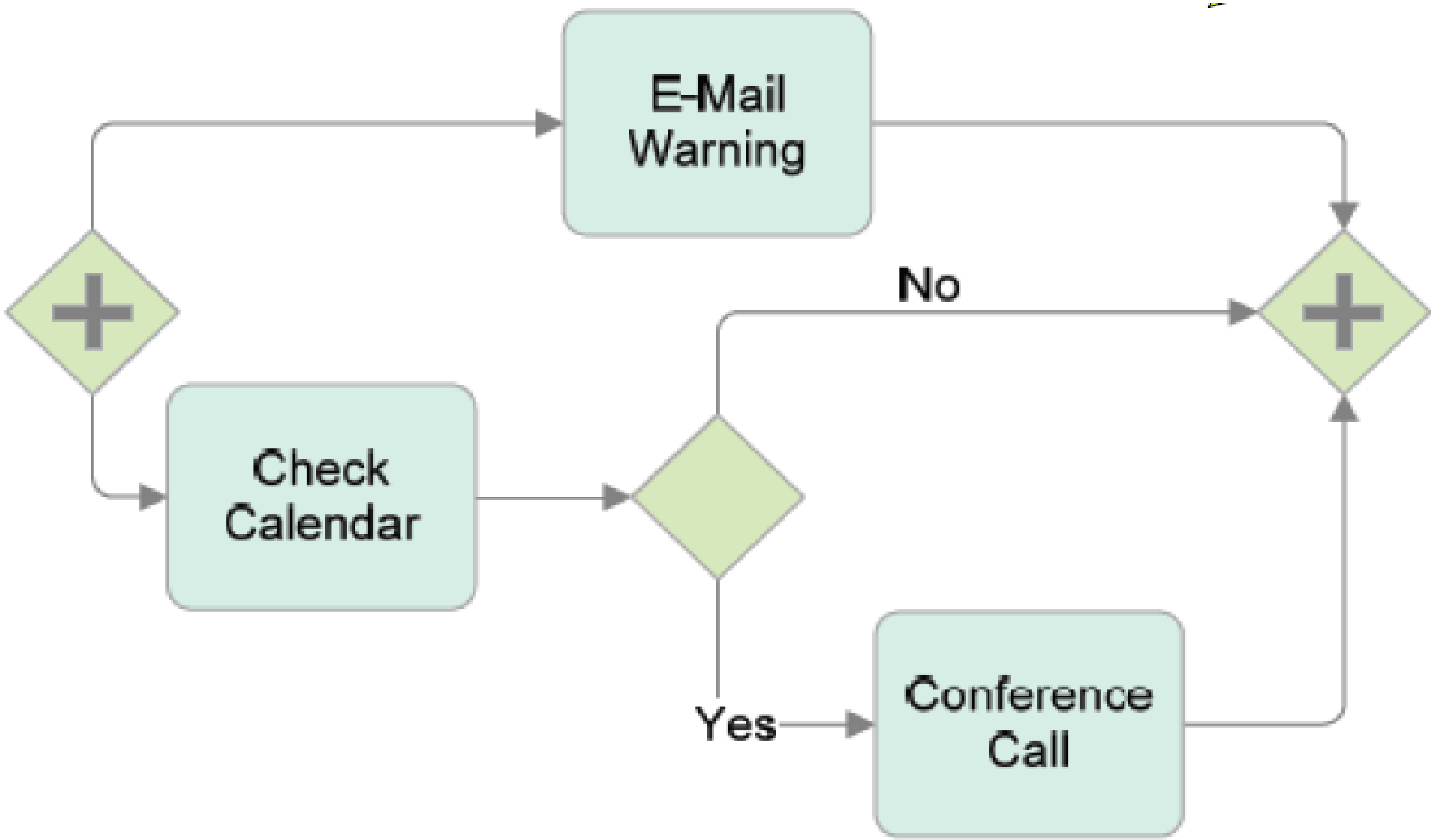




- The Parallel Gateway will wait for a token to arrive from each incoming Sequence Flow.
- When the first token arrives, there is no evaluation of a condition for the incoming Sequence Flow, but the token **is "held" at the Gateway and does not continue.**
- When all the tokens have arrived, then they are merged and one token moves down the outgoing Sequence Flow

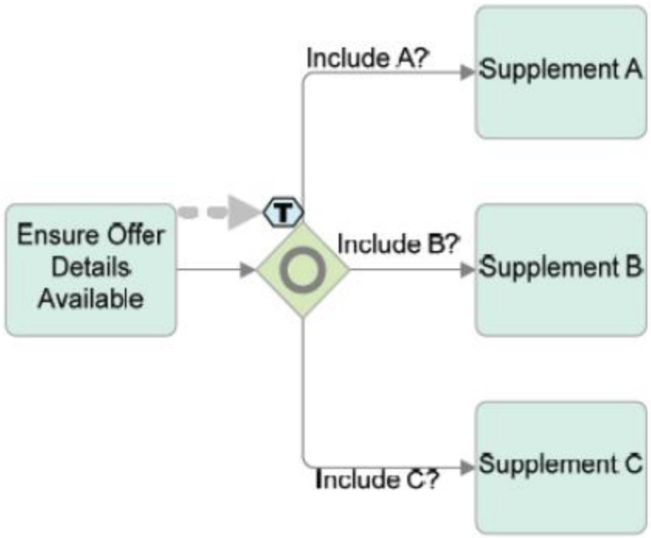
Pen and Paper Exercise

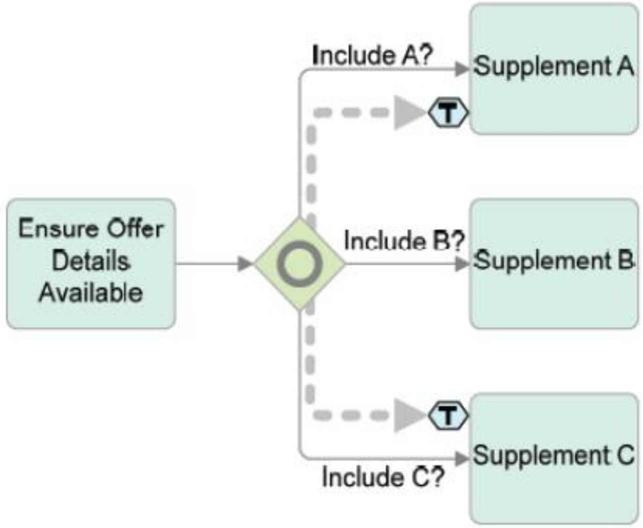
- Describe the behaviour of this process



Inclusive Gateways Splitting

- Inclusive Gateway creates alternative paths based on the conditions on those paths.
- Inclusive Gateway activates **1 or more** paths.
- Every condition that evaluates to true will result in a token moving down that Sequence Flow.
- At least one of those conditions must evaluate to true.

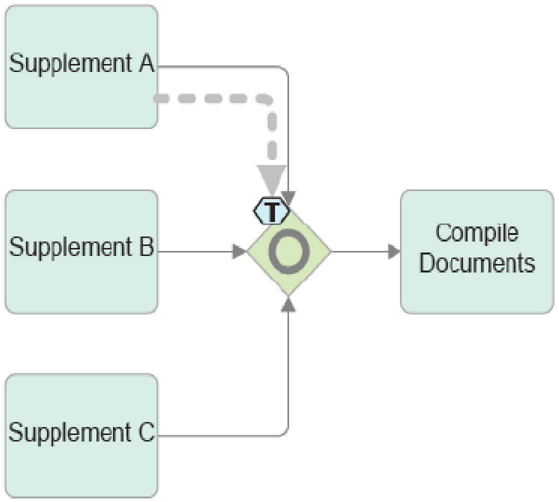


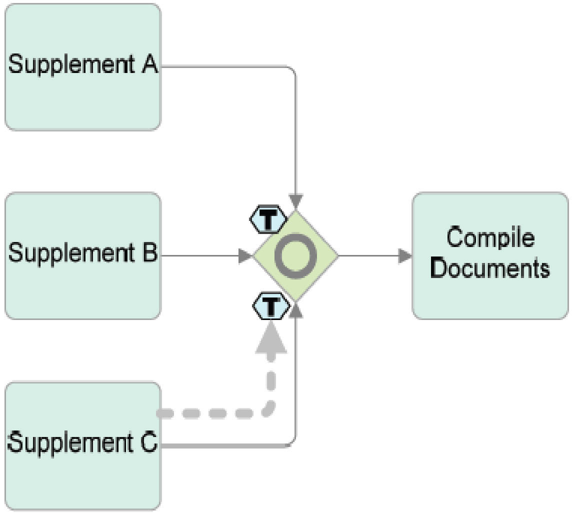


Inclusive Gateways Splitting

- When the first token arrives, the Gateway will look upstream for each of the other incoming Sequence Flow to see whether there is a token that might arrive at later time.

- When the expected tokens have arrived at the Gateway, the Process flow is synchronized (i.e. the incoming tokens are merged).



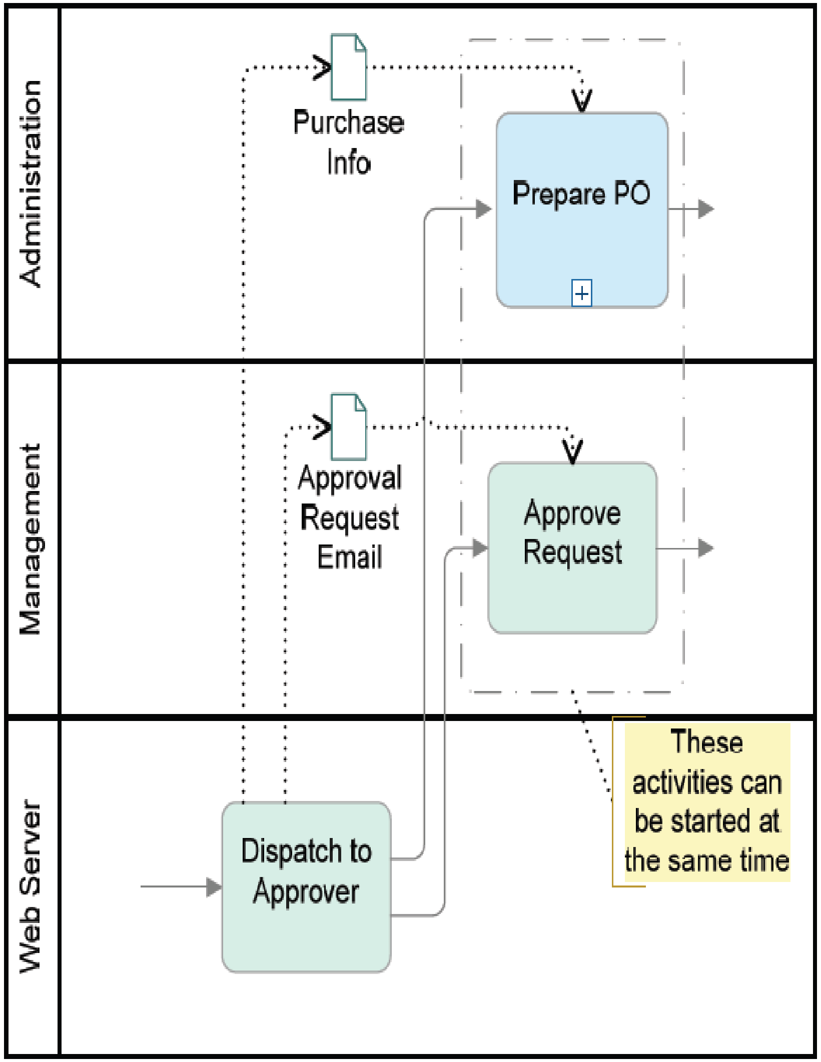


Swimlanes

- BPMN uses “swimlanes” to help partition and/organize activities in a diagram. There are two main types:
  - ***Pools*** - act as containers for a Process, each one representing a **participant** in a collaborative Business Process Diagram.
  - ***Lanes*** - often assumed to represent **internal business roles** within a Process, Lanes actually provide a generic mechanism for partitioning the objects within a Pool based on the characteristics of the Process or elements.

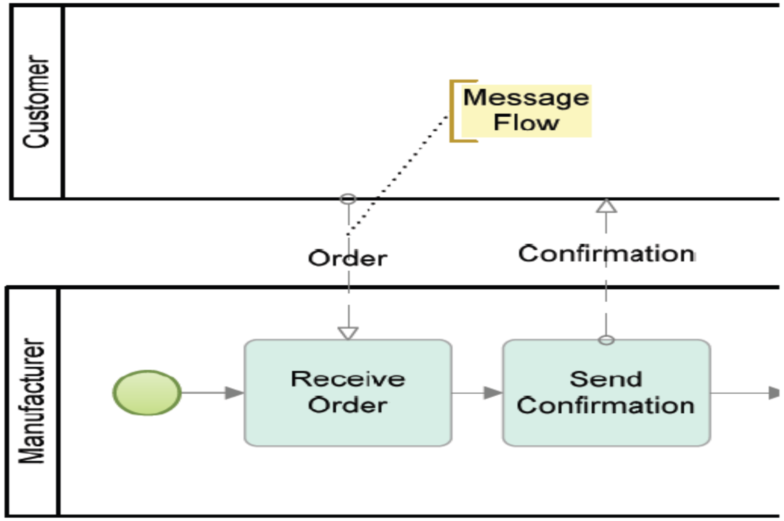
Lanes

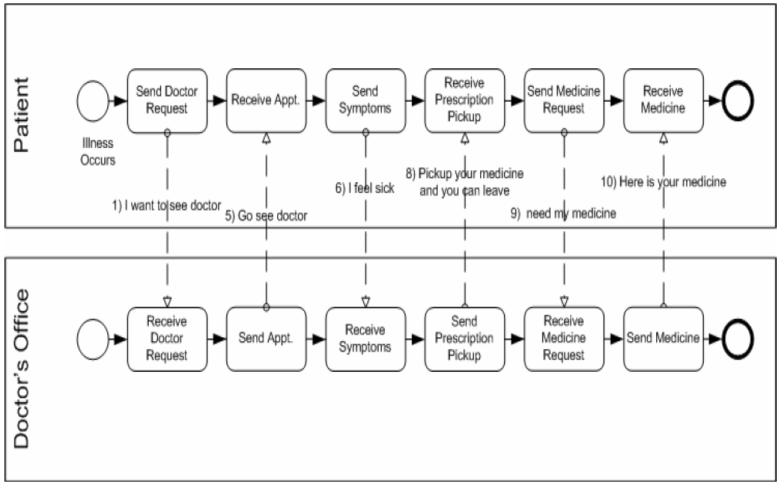
- Lanes often represent **organization roles** (e.g., Manager, Administration, Associate, etc), but can represent any **desired classification** (e.g., underlying technology, organizational departments, company products, etc).
- Sequence Flow can cross Lane boundaries.
- Message Flow is **not** used within or across Lanes of a Pool.
- Lanes can be nested.



Message Flows

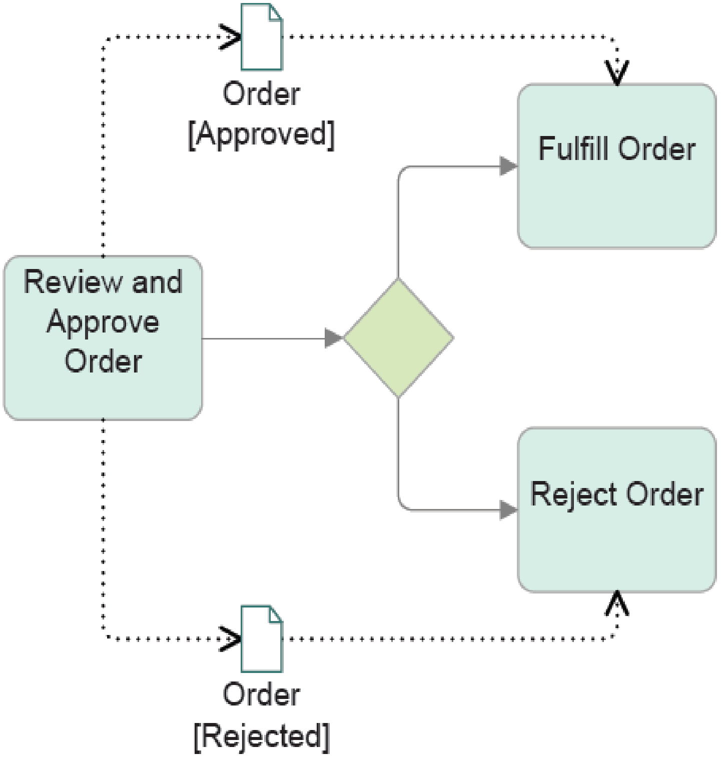
- Message Flow defines the messages/communications between two separate participants (shown as Pools) of the diagram.
- Message Flow **must always occur between two separate Pools** and **cannot** connect two objects within a single Pool.
- Where a Pool has Process elements, the Message Flow connects to those elements
  - When a Pool is black box, Message Flow connects to its boundary
- Sequence Flow cannot cross a Pool boundary - i.e., a Process is fully contained within a Pool





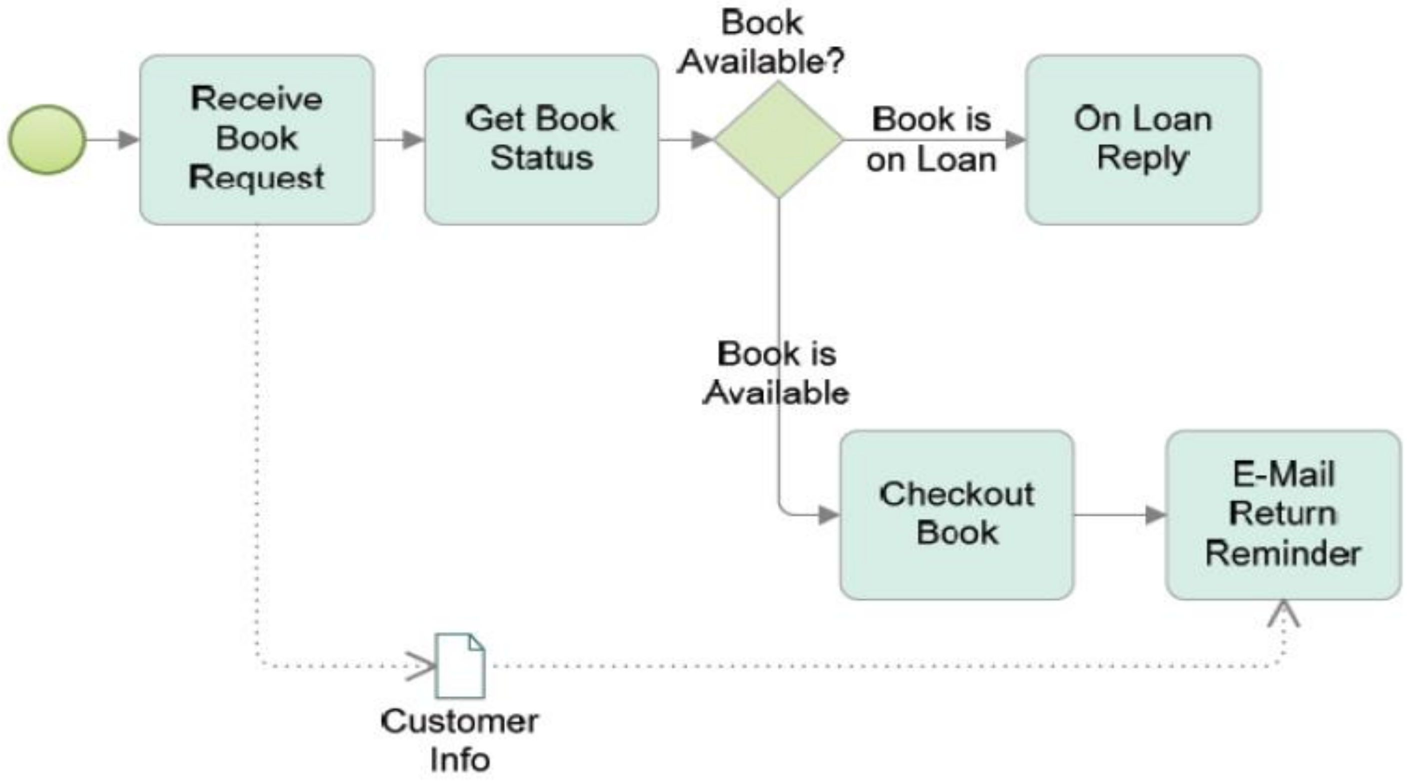
Data Object

- Data Objects represent data and documents in a process.
- Data Objects usually define the **inputs** and **outputs** of Activities.
- Data Objects also have "states" that depict how the object is updated within the process.
  - Can be used to describe the changes that a Data Object will go through during the Process.



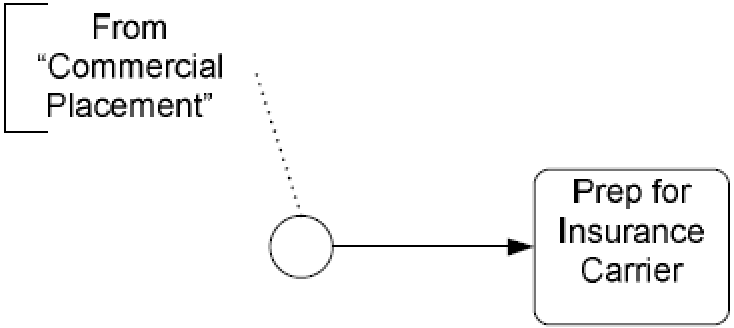
Data Flow

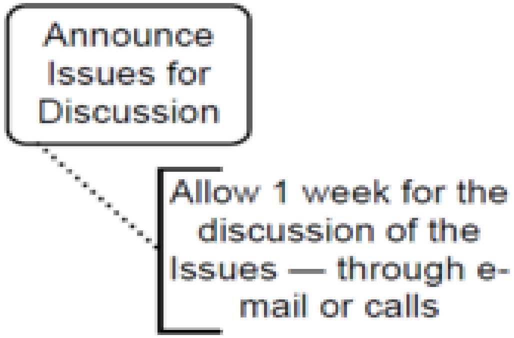
- Data flow represents the movement of Data Objects from into and out of Activities.
- In BPMN, data flow **is decoupled** from the Sequence Flow



Text Annotations

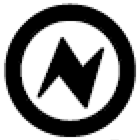
- Text Annotations provide the ability to add further descriptive information or notes about a Process and its elements.
- Text Annotations can connect to any object on the diagram or they can float freely anywhere on a diagram



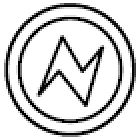

Announce Issues for Discussion
Allow 1 week for the discussion of the Issues — through e-mail or calls

Error Events

- Error Intermediate Event describes the occurrence of an error that needs the interrupting of an Activity (to which it is attached).
- An error is generated by applications or systems involved in the work or by End Events
- Error End Event is used to throw an error
- Error Intermediate Event can only be used when attached to the boundary of an Activity
  - It can only be used to catch an error.

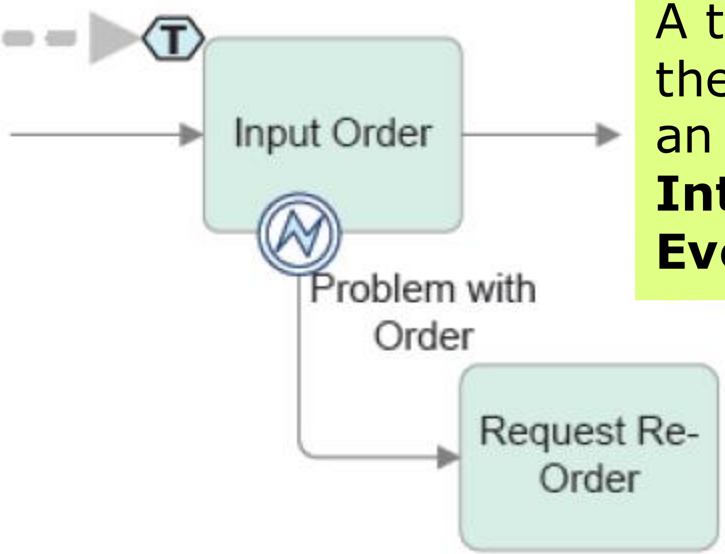


Error End Event Throwing

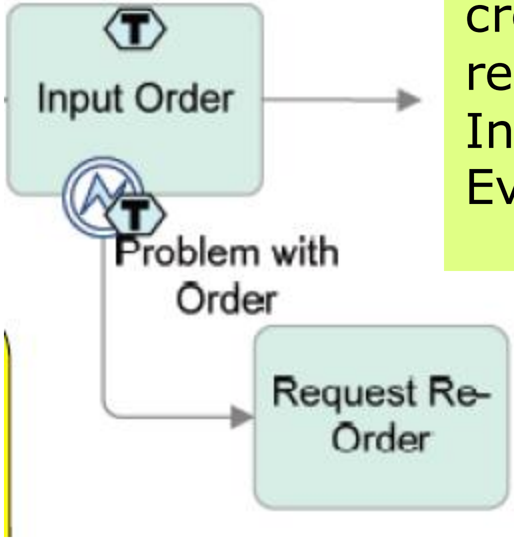


Error Intermediate Event Catching

Exception Handling

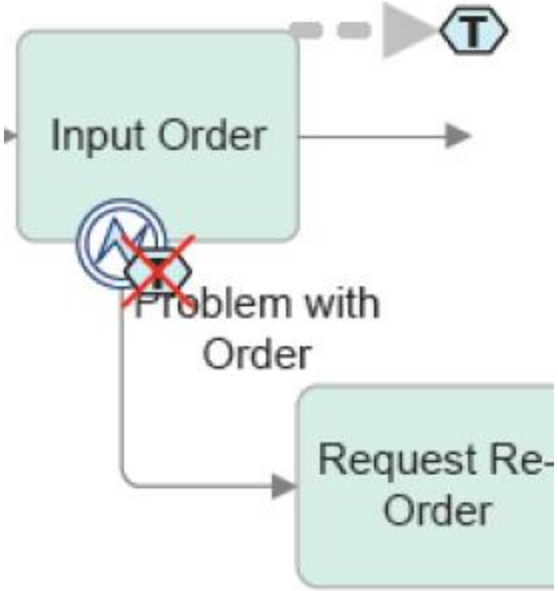


A token arrives at the Activity with an attached **Error Intermediate Event**

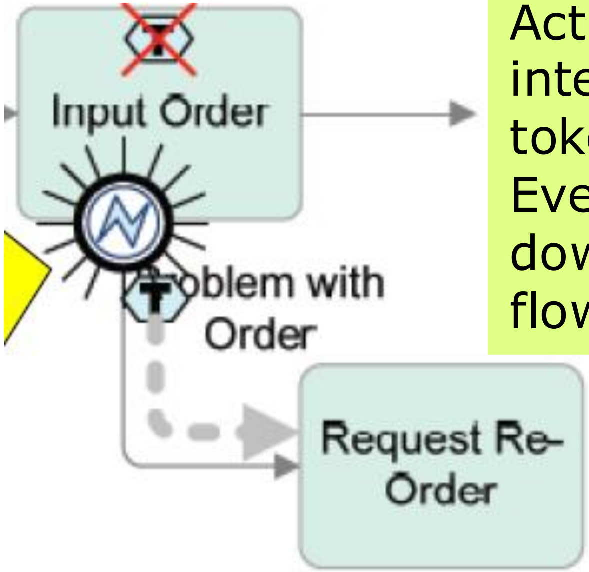


Start performing the Activity, and at the same time another token is created and resides in the Intermediate Event

Exception Handling (cont.)



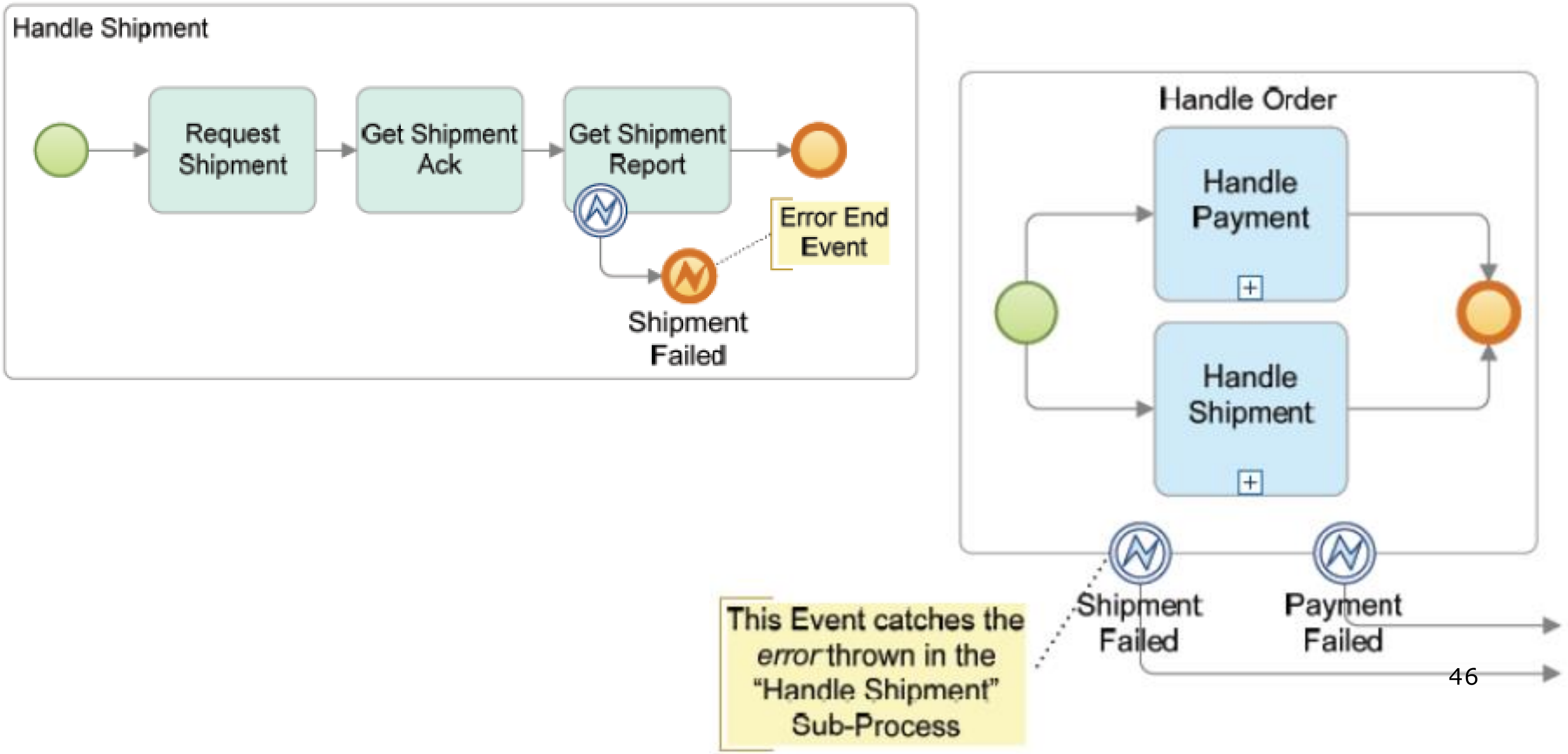
If the Activity finishes without any error event fired, the token leaves the Activity through the normal outgoing flow.



If the attached Intermediate Error event triggers before the Activity finishes, then the Activity is interrupted, the token from the Event moves down its outgoing flow.

Exception Handling (cont.)

- The error thrown by the (End) Event will be caught by an Intermediate Event a higher level.
- Errors only move upward in the Process hierarchy.

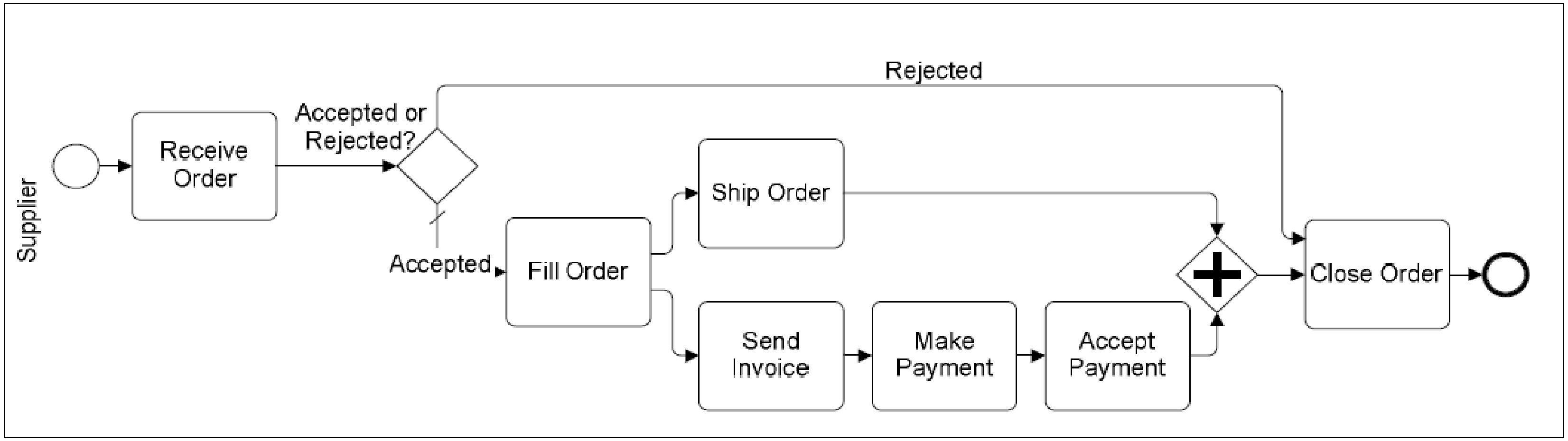


Error End Event

This Event catches the *error* thrown in the “Handle Shipment” Sub-Process

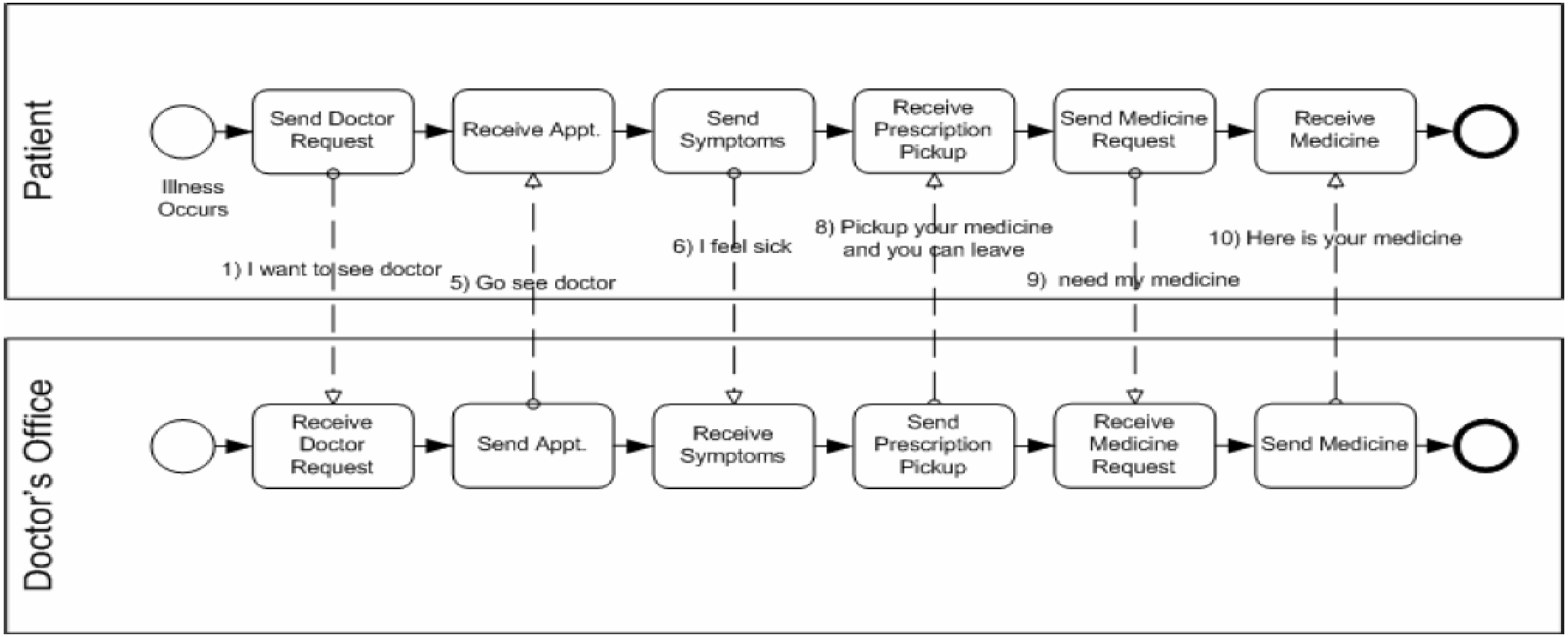
Orchestration

- Orchestration describes how a single business entity (i.e. a process participant such as a buyer, shipper, seller, or supplier) does things.
- Each orchestration appears within its own container (called a Pool). Each Pool can only represent one participant.



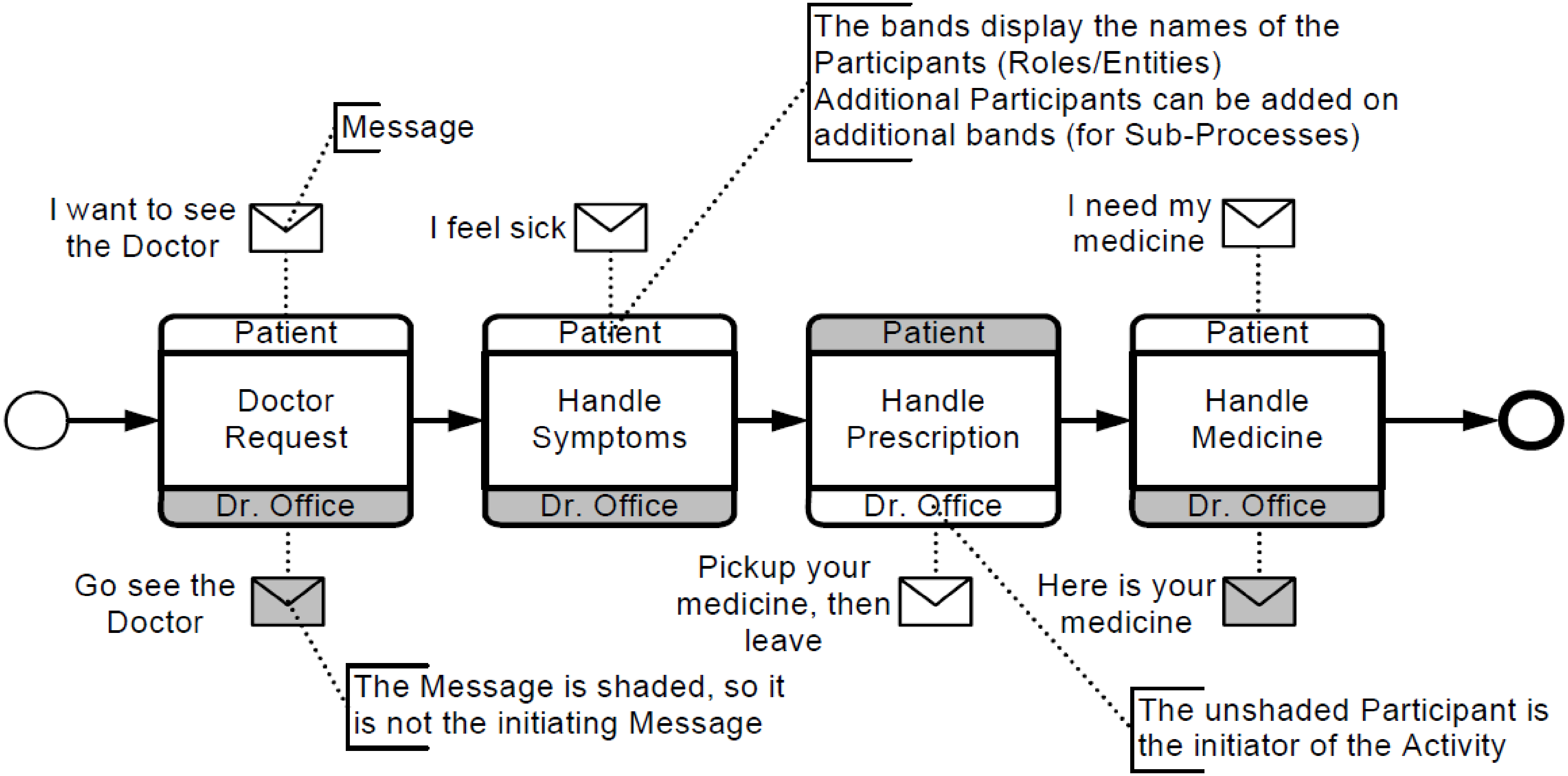
Choreography

- A Choreography process depicts the interactions between two or more business entities (as modelled with Pools)
  - Shown by the Message Flow between the Pools



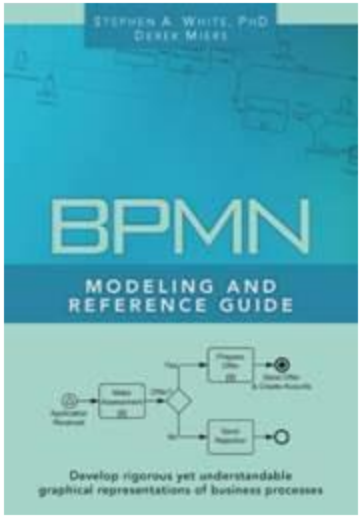
Choreography (cont.)

- In BPMN 2.0, Choreography provides a flow chart view to sequence the interactions between participants.


The bands display the names of the Participants (Roles/Entities)
Additional Participants can be added on additional bands (for Sub-Processes)
Message
I want to see the Doctor
I feel sick
I need my medicine
Patient
Doctor Request
Dr. Office
Patient
Handle Symptoms
Dr. Office
Patient
Handle Prescription
Dr. Office
Patient
Handle Medicine
Dr. Office
Go see the Doctor
Pickup your medicine, then leave
Here is your medicine
The Message is shaded, so it is not the initiating Message
The unshaded Participant is the initiator of the Activity

References

- Stephen A. White and Derek Miers, BPMN Modeling and Reference Guide, 2008.
- BPMN http://www.bpmn.org


STEPHEN A. WHITE, PhD
DEREK MIERS
BPMN
MODELING AND REFERENCE GUIDE
Develop rigorous yet understandable graphical representations of business processes