

## Activities

- Task**: A Task is a unit of work, the job to be performed. When marked with a symbol it indicates a Subprocess, an activity that can be refined.
- Transaction**: A Transaction is a set of activities that logically belong together; it might follow a specified transaction protocol.
- Event-Subprocess**: An Event-Subprocess is placed into a Process or Subprocess. It is activated when its start event gets triggered and can interrupt the higher level process context or run in parallel (non-interrupting) depending on the start event.
- Call Activity**: A Call Activity is a wrapper for a globally defined Subprocess or Task that is reused in the current process.

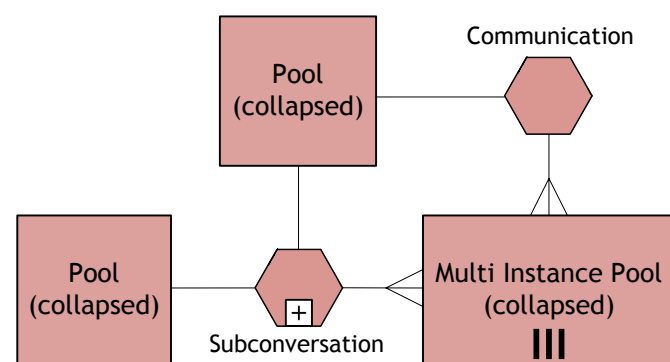
- Activity Markers**  
Markers indicate execution behavior of activities:
- Subprocess Marker
  - Loop Marker
  - Parallel MI Marker
  - Sequential MI Marker
  - Ad Hoc Marker
  - Compensation Marker
- Task Types**  
Types specify the nature of the action to be performed:
- Send Task
  - Receive Task
  - User Task
  - Manual Task
  - Business Rule Task
  - Service Task
  - Script Task

- Sequence Flow**: defines the execution order of activities.
- Default Flow**: is the default branch to be chosen if all other conditions evaluate to false.
- Conditional Flow**: has a condition assigned that defines whether or not the flow is used.

## Conversations

- A **Communication** defines a set of logically related message exchanges. When marked with a symbol it indicates a Sub-conversation, a compound conversation element.
- A **Conversation Link** connects Communications and Participants.
- A **Forked Conversation Link** connects Communications and multiple Participants.

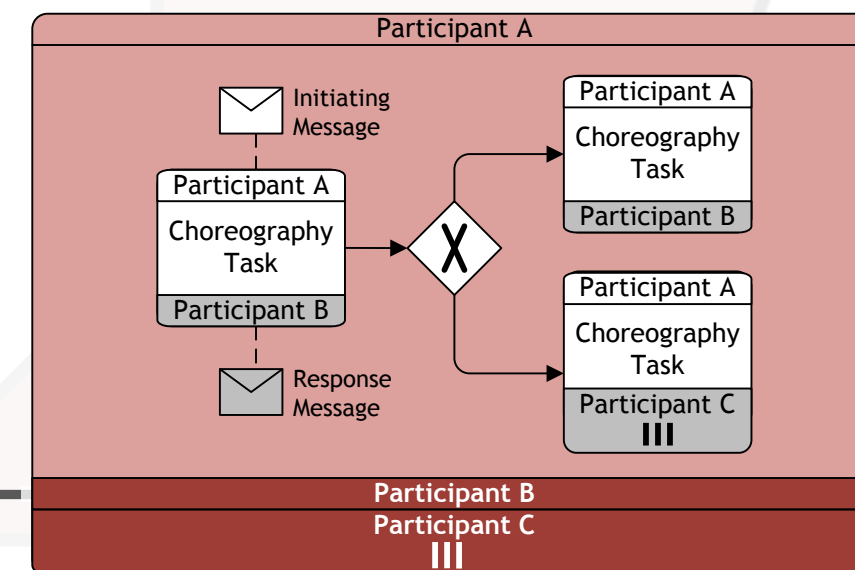
### Conversation Diagram



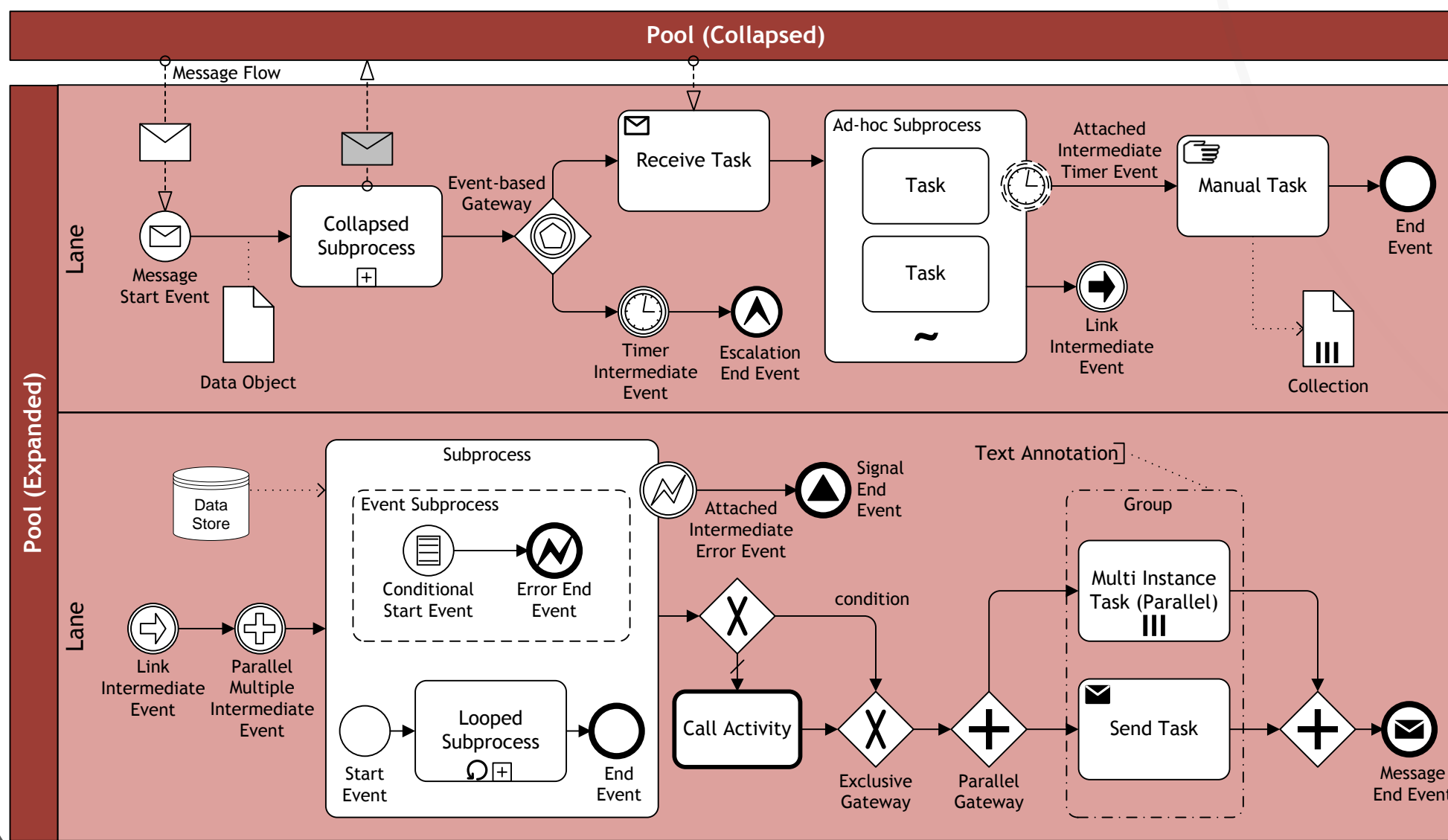
## Choreographies

- A **Choreography Task** represents an Interaction (Message Exchange) between two Participants.
- Multiple Participants Marker** denotes a set of Participants of the same kind.
- A **Choreography Sub-process** contains a refined choreography with several Interactions.

### Choreography Diagram



## Collaboration Diagram



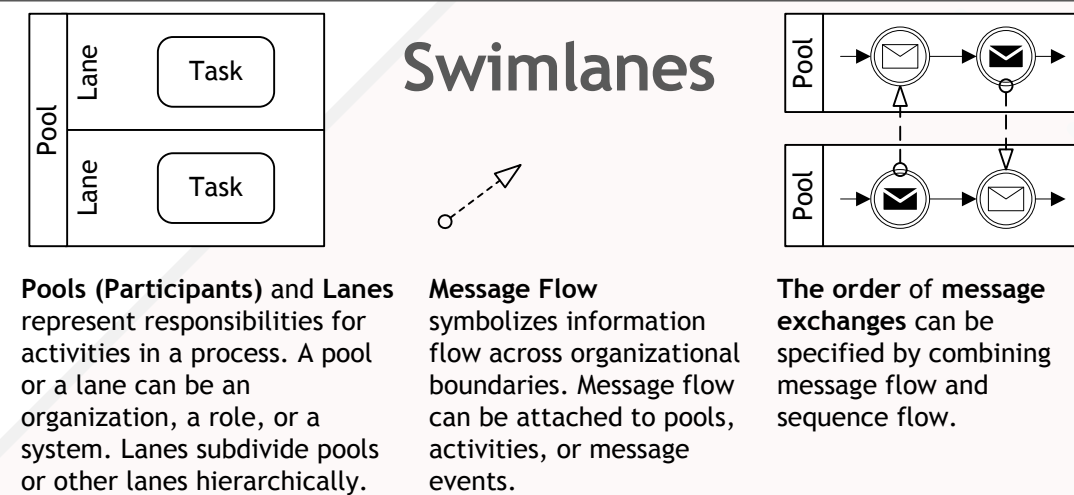
## Events

	Top-Level	Start	Intermediate	End
	Event Sub-Process Interrupting	Event Sub-Process Non-interrupting	Catching	Throwing
<b>None</b> : Untyped events, indicate start point, state changes or final states.				
<b>Message</b> : Receiving and sending messages.				
<b>Timer</b> : Cyclic timer events, points in time, time spans or timeouts.				
<b>Escalation</b> : Escalating to an higher level of responsibility.				
<b>Conditional</b> : Reacting to changed business conditions or integrating business rules.				
<b>Link</b> : Off-page connectors. Two corresponding link events equal a sequence flow.				
<b>Error</b> : Catching or throwing named errors.				
<b>Cancel</b> : Reacting to cancelled transactions or triggering cancellation.				
<b>Compensation</b> : Handling or triggering compensation.				
<b>Signal</b> : Signalling across different processes. A signal thrown can be caught multiple times.				
<b>Multiple</b> : Catching one out of a set of events. Throwing all events defined.				
<b>Parallel Multiple</b> : Catching all out of a set of parallel events.				
<b>Terminate</b> : Triggering the immediate termination of a process.				

## Gateways

- Exclusive Gateway**: When splitting, it routes the sequence flow to exactly one of the outgoing branches. When merging, it awaits one incoming branch to complete before triggering the outgoing flow.
- Event-based Gateway**: Is always followed by catching events or receive tasks. Sequence flow is routed to the subsequent event/task which happens first.
- Parallel Gateway**: When used to split the sequence flow, all outgoing branches are activated simultaneously. When merging parallel branches it waits for all incoming branches to complete before triggering the outgoing flow.
- Inclusive Gateway**: When splitting, one or more branches are activated. All active incoming branches must complete before merging.
- Exclusive Event-based Gateway (instantiate)**: Each occurrence of a subsequent event starts a new process instance.
- Complex Gateway**: Complex merging and branching behavior that is not captured by other gateways.
- Parallel Event-based Gateway (instantiate)**: The occurrence of all subsequent events starts a new process instance.

## Swimlanes



## Data

- Data Input**: A Data Input is an external input for the entire process. It can be read by an activity.
- Data Output**: A Data Output is a variable available as result of the entire process.
- Data Object**: A Data Object represents information flowing through the process, such as business documents, e-mails, or letters.
- Collection Data Object**: A Collection Data Object represents a collection of information, e.g., a list of order items.
- Data Store**: A Data Store is a place where the process can read or write data, e.g., a database or a filing cabinet. It persists beyond the lifetime of the process instance.
- Message**: A Message is used to depict the contents of a communication between two Participants.

