

Activities

- Task**: A Task is a unit of work, the job to be performed. When marked with a symbol it indicates a Sub-Process, 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 Sub-Process**: An Event Sub-Process is placed into a Process or Sub-Process. 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 Task or Process reused in the current Process. A call to a Process is marked with a symbol.

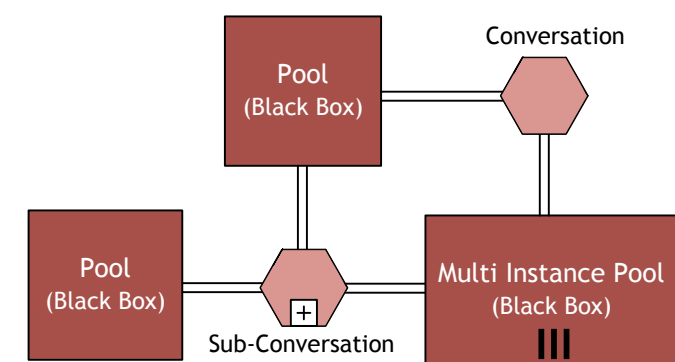
- Activity Markers**
Markers indicate execution behavior of activities:
- Sub-Process Marker
 - Loop Marker
 - Parallel MI Marker
 - Sequential MI Marker
 - Ad Hoc Marker
 - Compensation Marker
- Task Types**
Types specify the nature of the action to be performed:
- Send Task
 - Receive Task
 - User 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 Conversation defines a set of logically related message exchanges. When marked with a symbol it indicates a Sub-Conversation, a compound conversation element.
- A Call Conversation is a wrapper for a globally defined Conversation or Sub-Conversation. A call to a Sub-conversation is marked with a symbol.
- A Conversation Link connects Conversations and Participants.

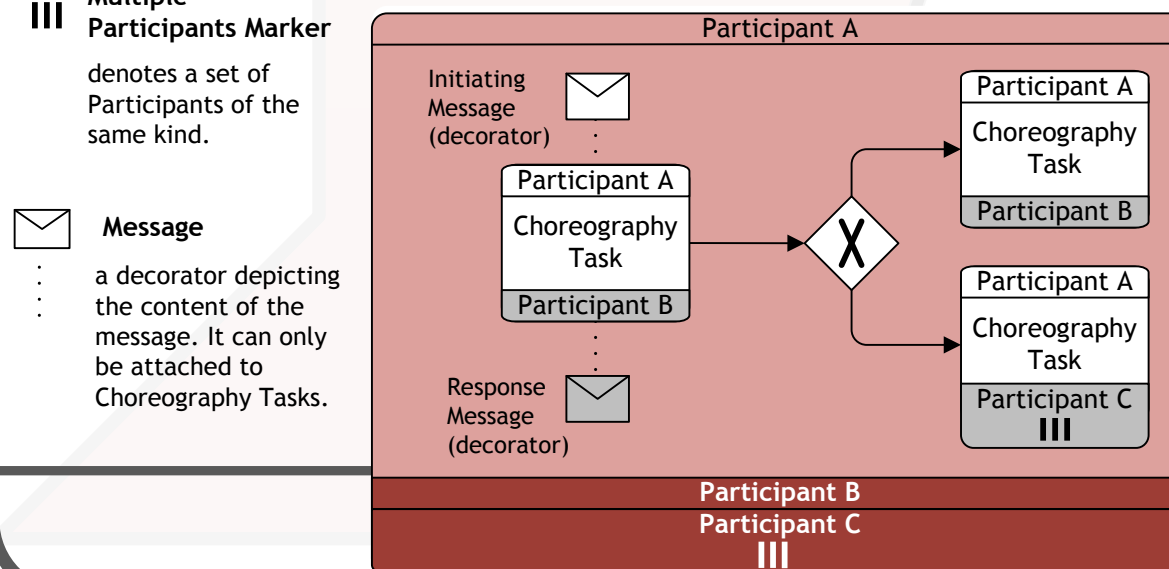
Conversation Diagram



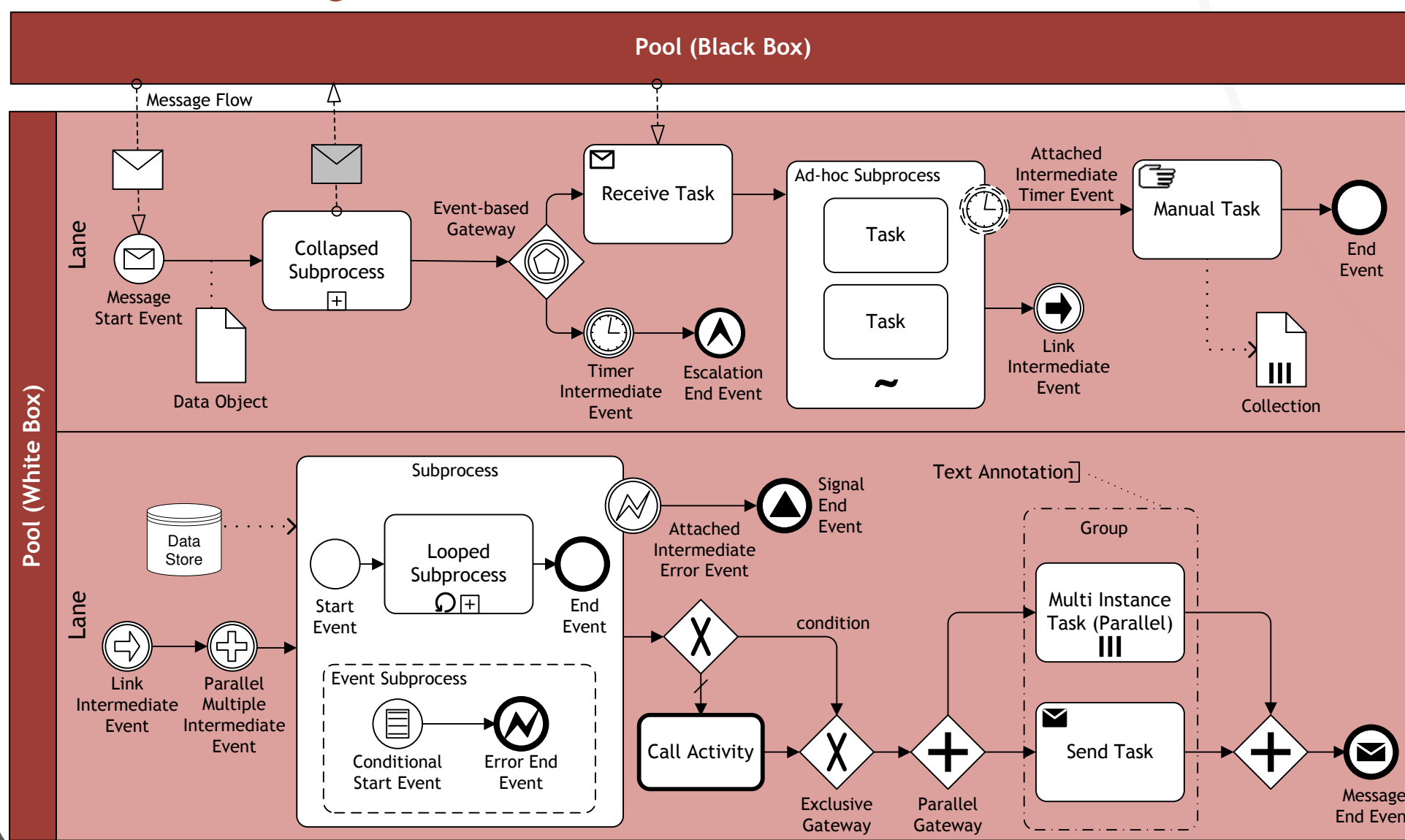
Choreographies

- Participant A**
Choreography Task
Participant B
 - Participant A**
Sub-Choreography
Participant B
Participant C
 - Participant A**
Call Choreography
Participant B
- A **Choreography Task** represents an Interaction (Message Exchange) between two Participants.
- A **Sub-Choreography** contains a refined choreography with several Interactions.
- A **Call Choreography** is a wrapper for a globally defined Choreography Task or Sub-Choreography. A call to a Sub-Choreography is marked with a symbol.

Choreography Diagram



Collaboration Diagram



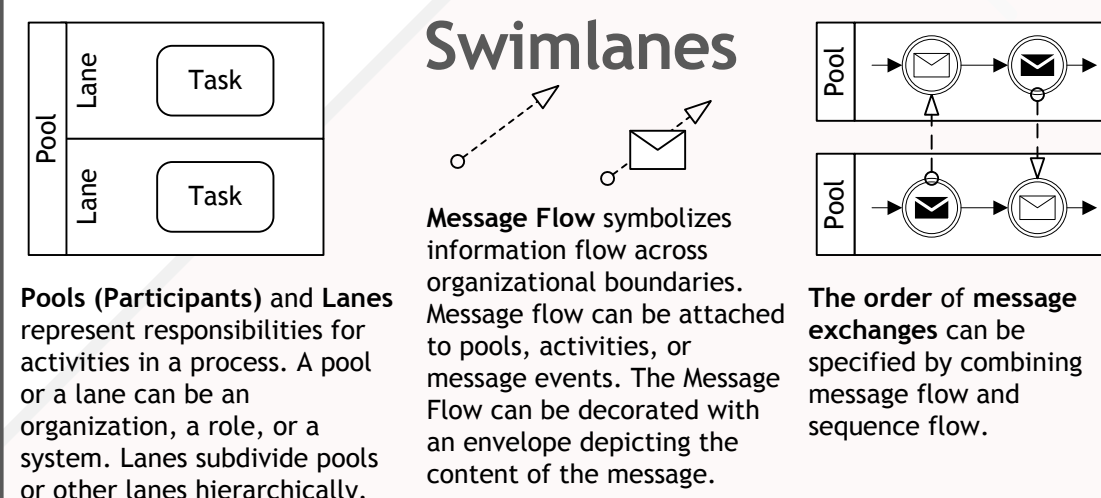
Events

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

- A **Data Object** represents information flowing through the process, such as business documents, e-mails, or letters.
- A **Collection Data Object** represents a collection of information, e.g., a list of order items.
- A **Data Input** is an external input for the entire process. A kind of input parameter.
- A **Data Output** is data result of the entire process. A kind of output parameter.
- A **Data Association** is used to associate data elements to Activities, Processes and Global Tasks.
- 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.

