### Activities

**Task** - Unit of work. The work or job that needs to be accomplished. + indicates a sub-process or an activity that can be refined.

**Transaction** - activities that sensibly go together. May follow particular transactional protocol.

**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** is a wrapper for a globally defined Sub-Process or Task that is reused in the current process.

**Activity Markers**
- Sub-Process Marker
- Loop Marker
- Parallel MI Marker
- Sequential MI Marker
- Ad Hoc Marker
- Compensation Marker

**Task Types**
- 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

Communications that identify a group of logically related message exchanges. + shows a sub-conversation or compound conversation element.

**Conversation Link** connects Communications to Participants.

**Forked Conversation Link** attaches Communications and many Participants.

### Events

<table>
<thead>
<tr>
<th>Events</th>
<th>START</th>
<th>INTERMEDIATE</th>
<th>END</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-Level</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Sub-Process</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Sub-Processing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event Sub-Delivering</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catching</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boundary Catching</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boundary Delimiting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boundary Interrupting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Throwing</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- None: Untyped events, indicate start point, state or changes or final states.
- Message: Receiving and sending messages.
- Timer: Cyclic timer events, points in time, time spans or timeouts.
- Escalation: Escalating to a 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: Signaling 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.

### Data

**Data Input** is an external input for the entire process. It can be read by an activity.

**Data Output** is a variable available as result of the entire process.

**Data Object** represents information flowing through the process, such as business documents, e-mails, or letters.

**Collection Data Object** represents a collection of information, e.g., a list of order items.

**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** is used to depict the contents of a communication between two Participants.
Collaboration Diagram

Pool (Collapsed)

- Message Flow
- Collapsed Sub-Process
- Event-based Gateway
- Receive Task
- Ad-hoc Sub-Process
- Attached Intermediate Timer Event
- Manual Task
- End Event

Pool (Expanded)

- Lane
-Collapsed Sub-Process
- Event Sub-Process
- Conditional Start Event
- Error End Event
- Attached Intermediate Error Event
- Signal End Event
- Logged Sub-Process
- Start Event
- End Event
- Attached Intermediate Event
- Escalation End Event
- Call Activity
- Exclusive Gateway
- Parallel Gateway
- Link Intermediate Event
- Collection
- Message Flow
- Data Object

Gateways

- Exclusive Gateway – Split – routes sequence flow to a precise outgoing branch. Merge – waits for one incoming branch to finish before triggering an outgoing flow.
- Parallel Gateway – When splitting sequence flow – outgoing branches are initiated simultaneously. When merging parallel branches- the gateway awaits all incoming branches to finish before triggering outgoing flow.
- Event-based Gateway – Must be followed by a catching event(s) or a receiving task(s). Sequence flow is sent to the subsequent event/task which happens first.
- Inclusive Gateway – One or more branches are activated when splitting. All active incoming branches must complete before merging.
- Complex Gateway – Gateways that represent actions not captured by other gateways. Can be complex, merging or branching actions/behaviors.
- Exclusive Event-based Gateway – Each occurrence of a subsequent event starts a new process instance.
- Parallel Event-based Gateway (Instantiate) – The occurrence of all subsequent events begins a new process instance.

Choreographies

- Participant A
- Choreography Task
- Participant B
- Participant C
- Multiple Participants Marker – represents a group of participants that are of the same kind.
- Choreography Sub-Process – holds a refined choreography with a number of Interactions.

Swimlanes

- Pools (Participants) and Lanes – show what an activity is responsible for in a process. Pools or lanes can represent the organization as a whole, a system or a role. Lanes are used to hierarchically separate pools or other lanes.
- Message Flow – can be connected to pools, activities or message events. Illustrate how information flows across organizational borders.
- The order of message exchange can be determined by various combinations of message and sequence flows.

Copyright Visumpoint and any reuse only through permission.