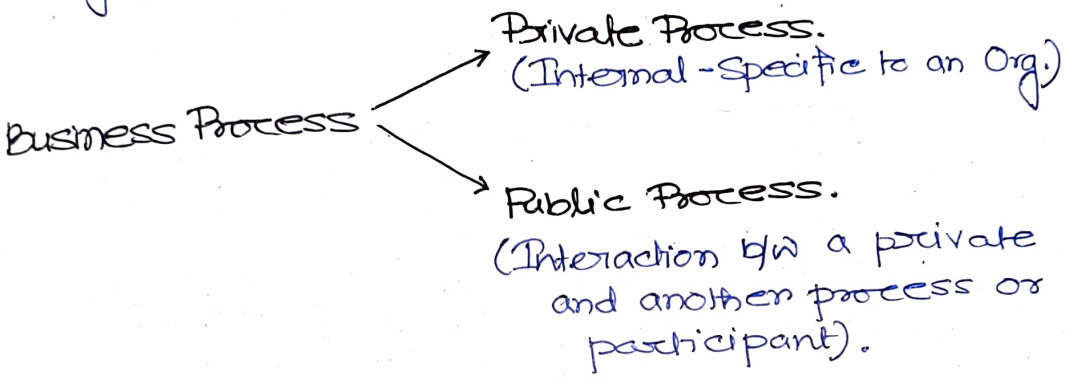
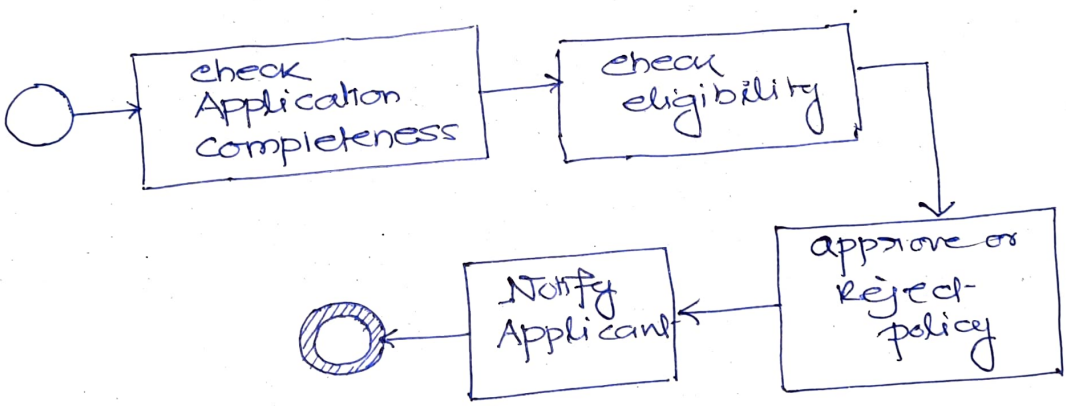
BPMN

BPMN stands for Business Process Model and Notation, is a graphical representation for specifying business processes in a business process model. Developed by the Object-Management-Group.



- Private Process:

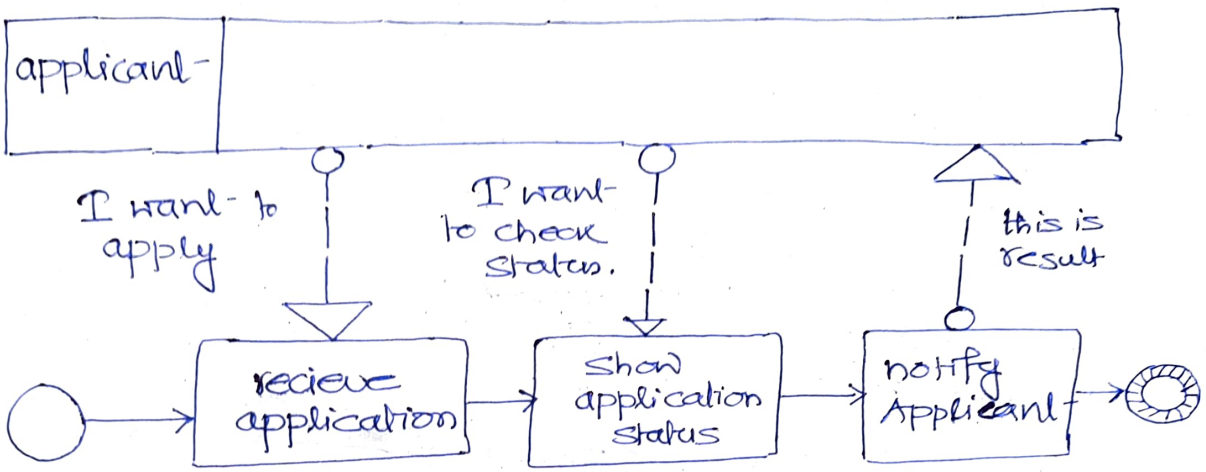


- Public Process:

1. Interaction b/w a private processes and another process or participant.

2. All internal activities of private process are not shown, but only those that are used to communicate with external parties are shown.

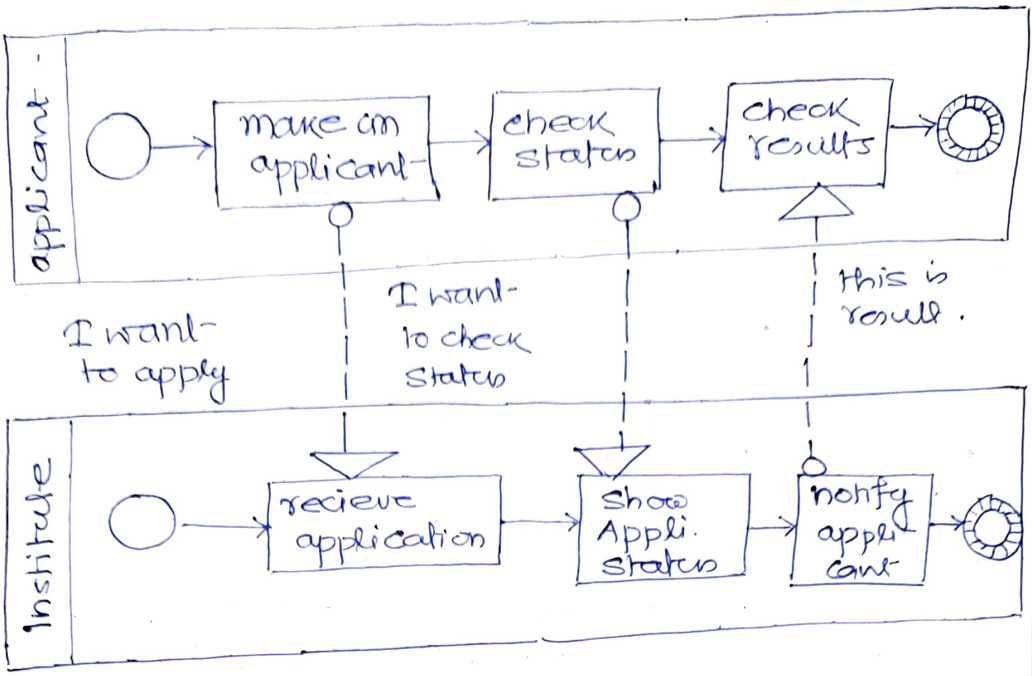
3. It shows the external world the msg flows and their order needed to interact with the process.



▷ Collaborations:

- Interaction between two business entities.
- Two or more pool representing participants.
- message exchange is shown by message flows that connects pool.
- messages associated with message flows can also be shown.
- touch points: collaboration activities in a public process.
- Corresponding internal processes may have much more activities than what is shown in public process.

Eg.



BPMN Notation:

The main elements of BPMN are-

1. Flow objects: Activities, Events & Gateways.
2. Connecting objects: Message Flow, Seq. Flow, Association.
3. Swimlanes: Pool or Lanes.
4. Artifacts: Group, Data object and Annotation.

1.1. Activities:

Activities denote the "tasks" to be performed by a business process. Represented as rounded corner rectangular shapes.

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

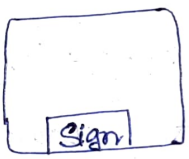
An Event sub-process is placed into a process or sub process. It's activated when its start event gets triggered and can interrupt the higher level process context or run in parallel (non-interrupting) depending on the start event.

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 indicates execution behavior of activities:



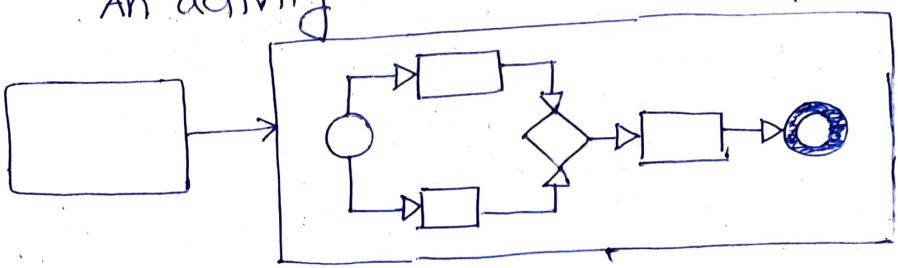
[+] : Sub-Process Markers.
↺ : Loop Marker.
||| : Parallel MI Marker.
☰ : Sequential MI Marker. (Multi-instance).
~ : Ad Hoc Marker
⏪ : Compensation marker.

- **Task:** Atomic Activity with in a process flow. "Cannot be further broken down."

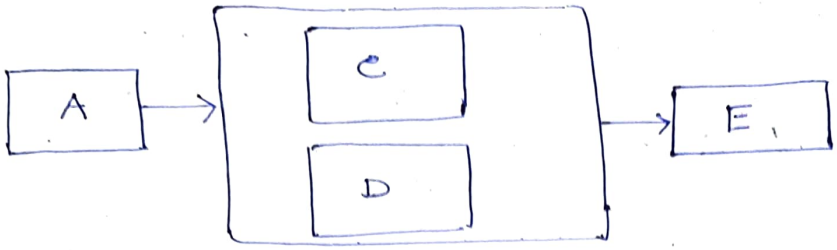
Collect reciept

- **Sub-process:**

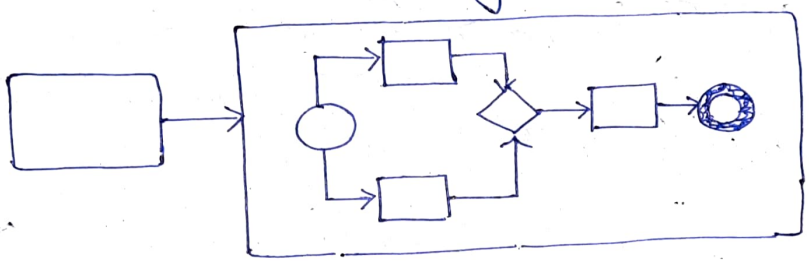
An activity which is expanded further.



Parallel boxes in Subprocess:



- **Call Activity:**
  - → A global process or a global task is used.
  - → Execution is transferred to the global task/process.
  - → shown in thick boundary.



- **Transactions:**
  - → A sub-process.
  - → Double Line boundary is used.
  - → should be completed fully or cancelled.

- **Task Types:** Types specify the nature of the action to be performed,—

 Send Task.

 Recieve Task.

 User Task.

 Manual task.

 Business Rule Task

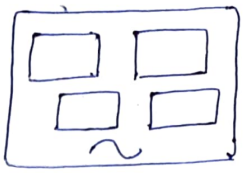
 Service task.

 Script-task.

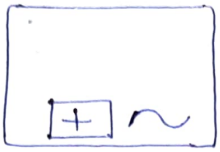
 Recieve task that instantiates a process.

 Collapsed Subprocess.

- Ad-hoc subprocess: Activities in it do not have sequence relationship.

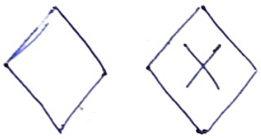


- Collapsed Ad-hoc subprocess.

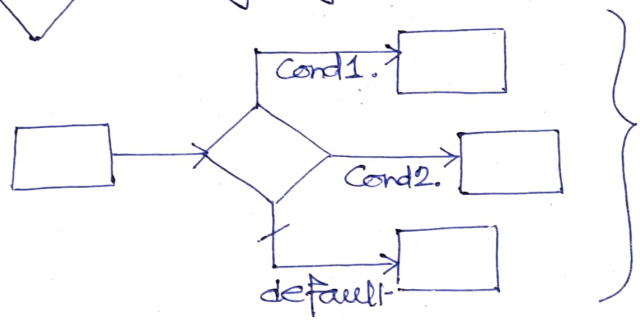


1.2. GATEWAYS:

- Exclusive Gateway:

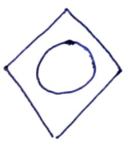


When splitting, it routes the sequence flow to exactly one of the outgoing branches. When merging, it awaits one incoming branch to complete before triggering the outgoing flow.

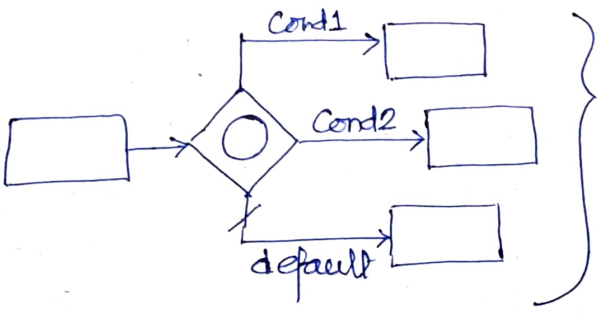


only 1 of the diversions is taken.

- Inclusive-based Gateway



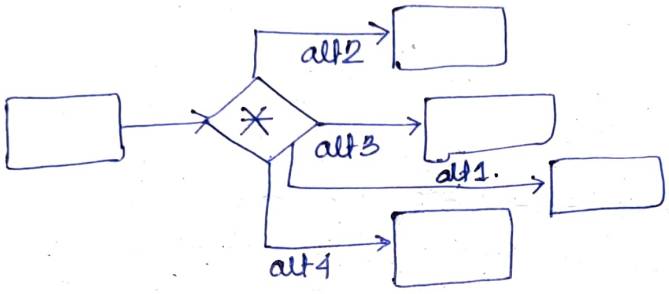
Is always followed by catching events or recieve tasks. Sequence flow is routed to the subsequent-event/task which happens first.



All ~~inclusive~~ conditions are evaluated and all true evaluation are followed (it could be zero to all paths).

- **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.

- **Complex Gateway:** Complex merging and branching behaviour that is not captured by other gateways.



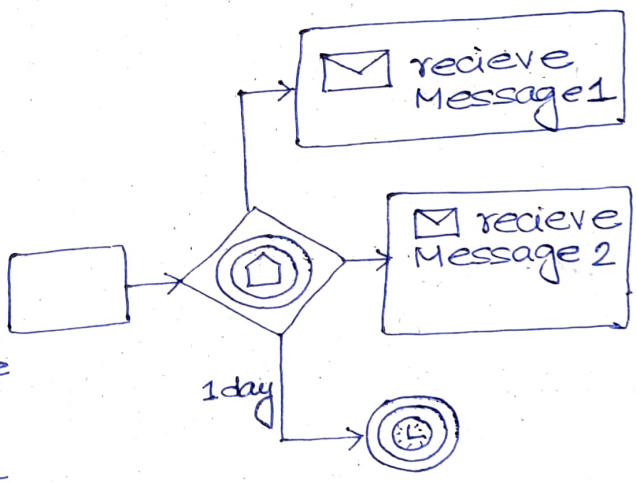
→ A complex synchronization.

→ k out of n tokens are produced.

- **Event-based Gateways:**

→ eg. When a message arrives follow the gateway.

→ The decision gets made by another participant- from where a message arrives.



→ Gateway does not get followed due to the process itself.

→ Outgoing flow does not carry a conditional expression.

→ The figure shows use of recieve tasks and timer event.

Exclusive Event-based Gateway (instantiate):

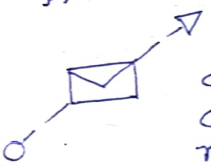
Each occurence of a subsequent-event starts a new process instance.

Parallel Event-based Gateway (instantiable):

The occurence of all subsequent-events starts a new process instance.

1.3. Message Flow:

Being represented by a dashed line with a circle at the start and an arrow at the end, message flow is a symbol to be used across pools or lanes to send information. It is the way to get work done in different organizations or departments.



depicting the content of the message.

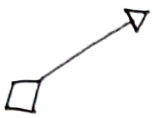
1.4. Sequence Flow:

It is a symbol used to represent the pure flow of work. The sequence flow is necessary to connect different activities and show their order through a straight-line ending with an arrow.

: is the default flow, default-branch to be chosen if all other conditions evaluate to false.

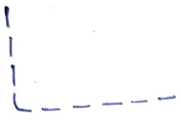
Default flow.

Conditional Flow : Has a condition assigned that defines whether or not the flow is used.



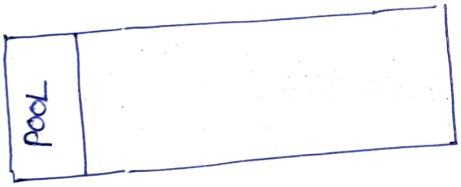
1.5. Association :

It is a dotted line used to link the event- or text to a business or gateway.



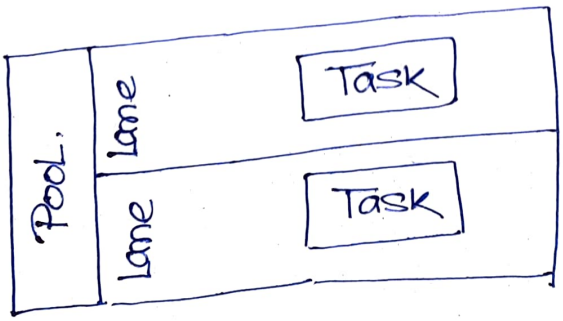
1.6. Pool :

Pools represents who is doing the job. They indicate the whole workflow upon which different- business processes are getting done. Pools show participants such as companies, departments.

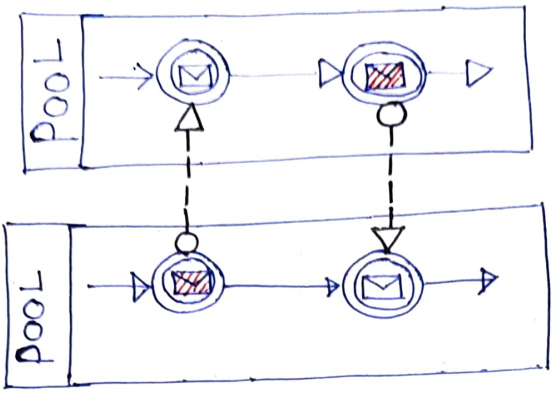


1.7. Lanes :

Lanes represents one parallel within the process and shows both responsibilities as well as task of each department-. Lanes are roles or specific position positions in different- organizations. Lanes subdivide pools or other lanes hierarchially.



The order of message exchanges can be specified by combining message flow and sequence flow.



1.8. Events:

- Something that happens during flow.

eg. example: start of an activity, end of an activity, a message that arrives, change in data state.

- Event driven processes can be described

- Start Events indicate where a process will start.

- End Events indicate where a process will end.

- Intermediate events. indicate something happening during process execution.

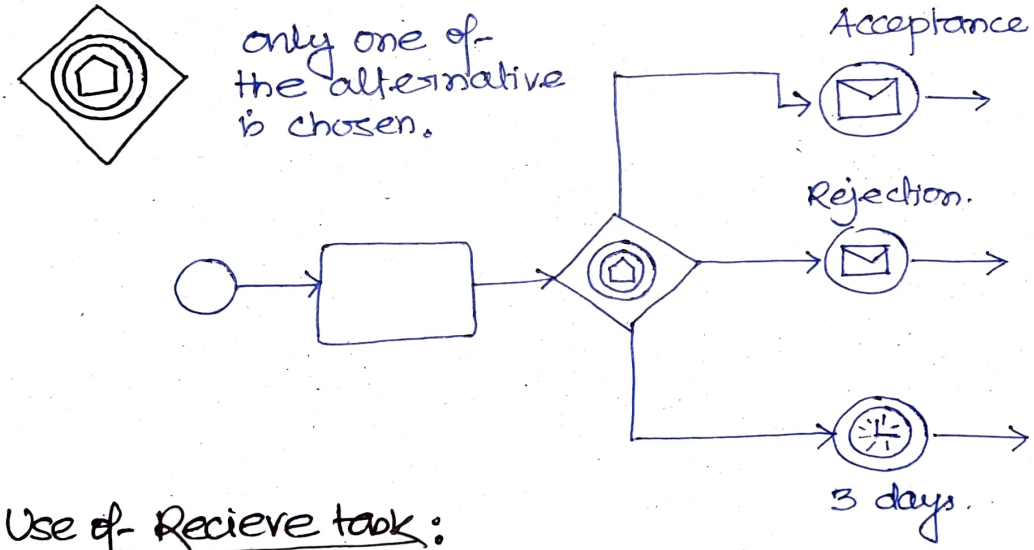
▷ Catching Events:

- Some events catch a trigger.
- all start events, some intermediate events.

▷ Throwing Events:

- Some events throw a result.
- all end events, some intermediate event throw a result.
- A thrown result may be caught by another event. (trigger carries the info. from throwing scope into catching scope).

▨ Event-Based Gateways: Use of intermediate events:



- Use of Recieve task: