Camunda In Action

Michaël Moo Penn • 11/11/2020

Introduction

What is BPM?

Definition

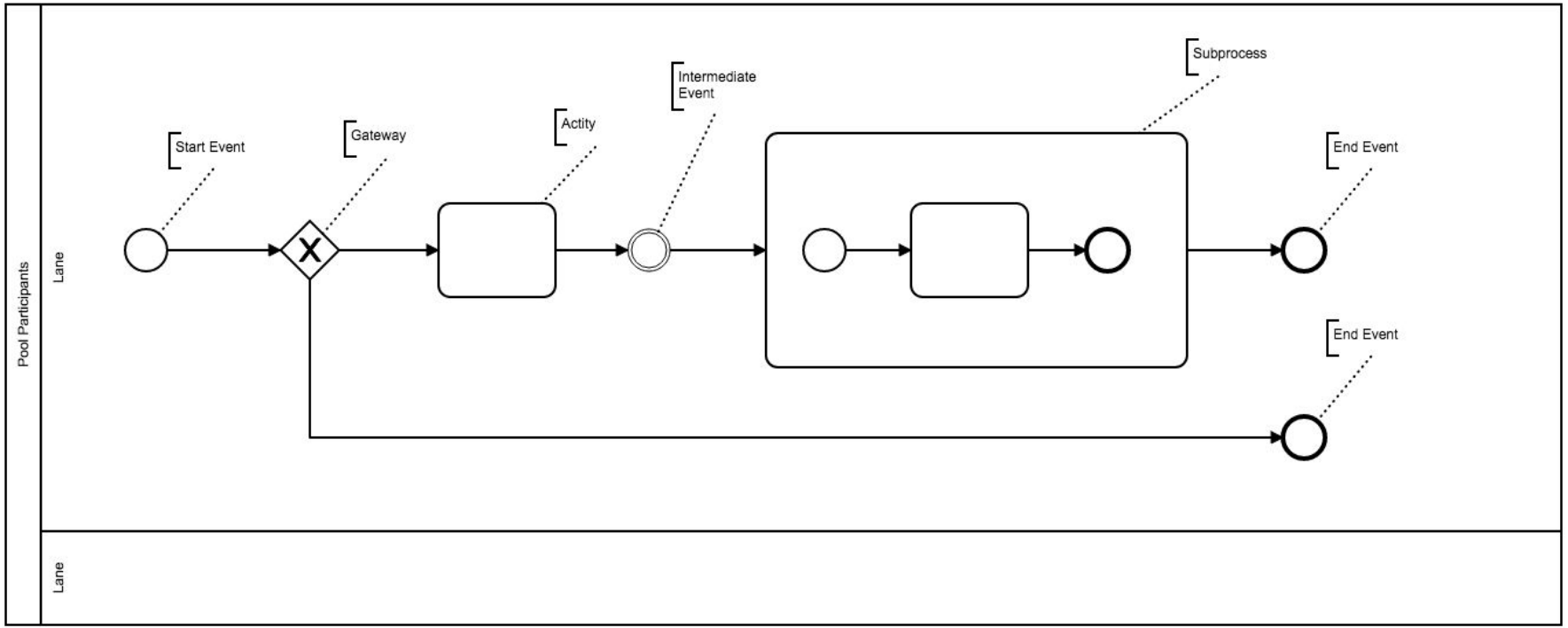
Business process management (BPM) is a discipline that uses various methods to discover, model, analyze, measure, improve and optimize business processes. A business process coordinates the behavior of people, systems, information and things to produce business outcomes in support of a business strategy.

Gartner

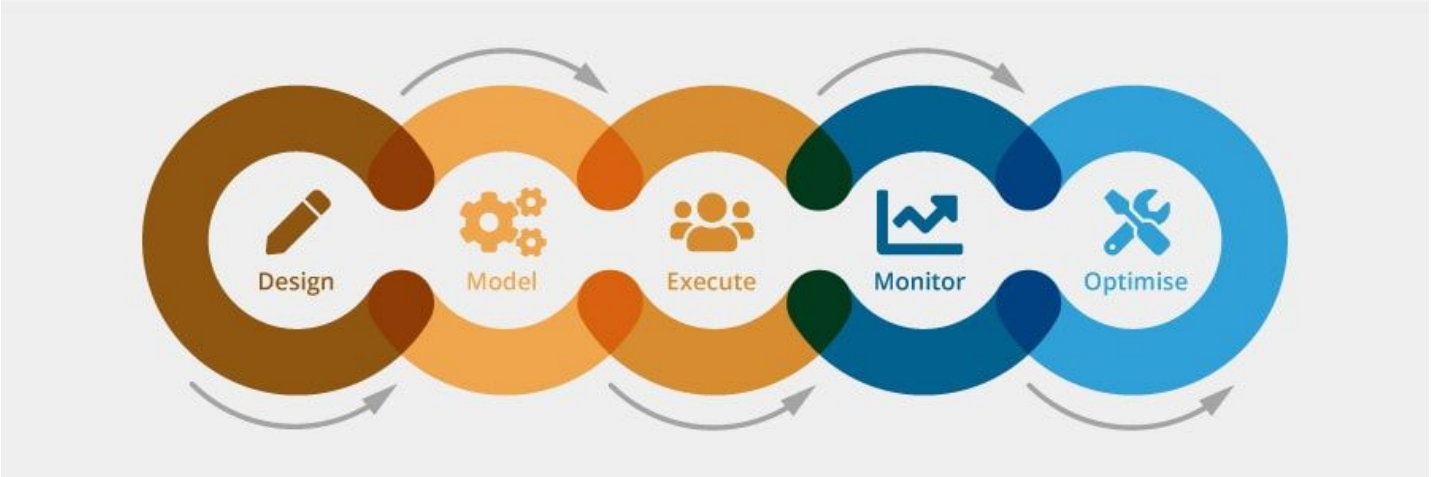
Key Infos

- OMG (Object Management Group)
- BPMN (Business Process Model and Notation - 2001)
- Current version 2.0.2
- Low Code

BPMN Components



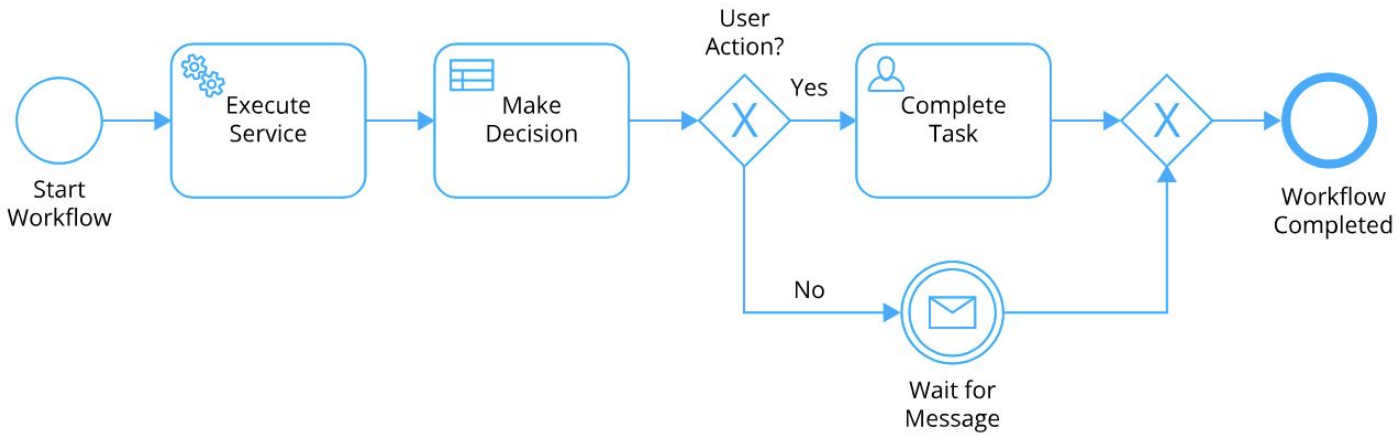
Project Life Cycle



Benefits

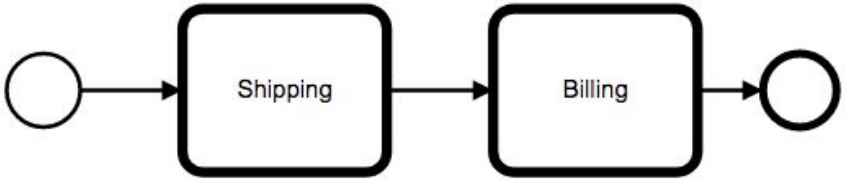
Transparency

BPMN allow to have a clear overview of the business processes in a company and how do they interact with each other.



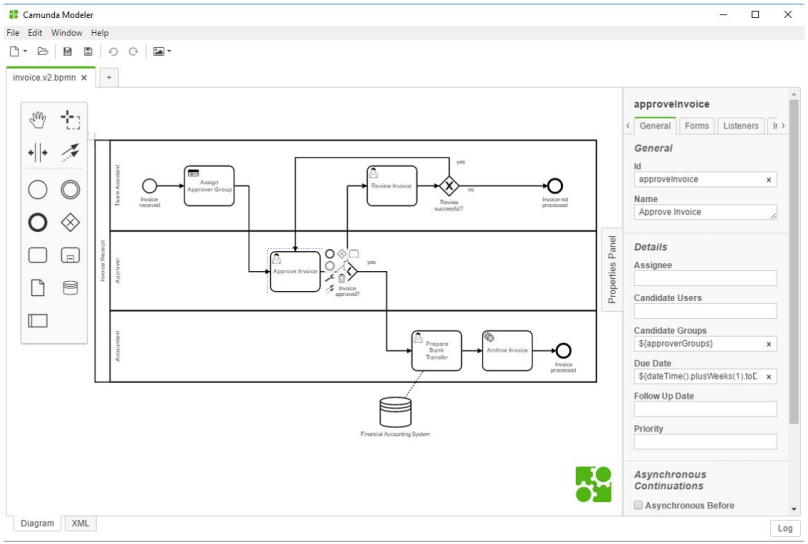
Process Consistency

BPM helps with process standardization – creating the one way of doing things that better than the rest. Process are reusable and can be called by another process.



Agility

BPMN model are really easy to update with the modeler, the BPM engine manage the different versions of your business process model. This allow to change your business processes with a better time to market.



Measurability

The BPM engine stores KPI of the deployed process model out of the box. It allows to do BAM (Business Activity Monitoring) and find what needs to be improved.

At The End

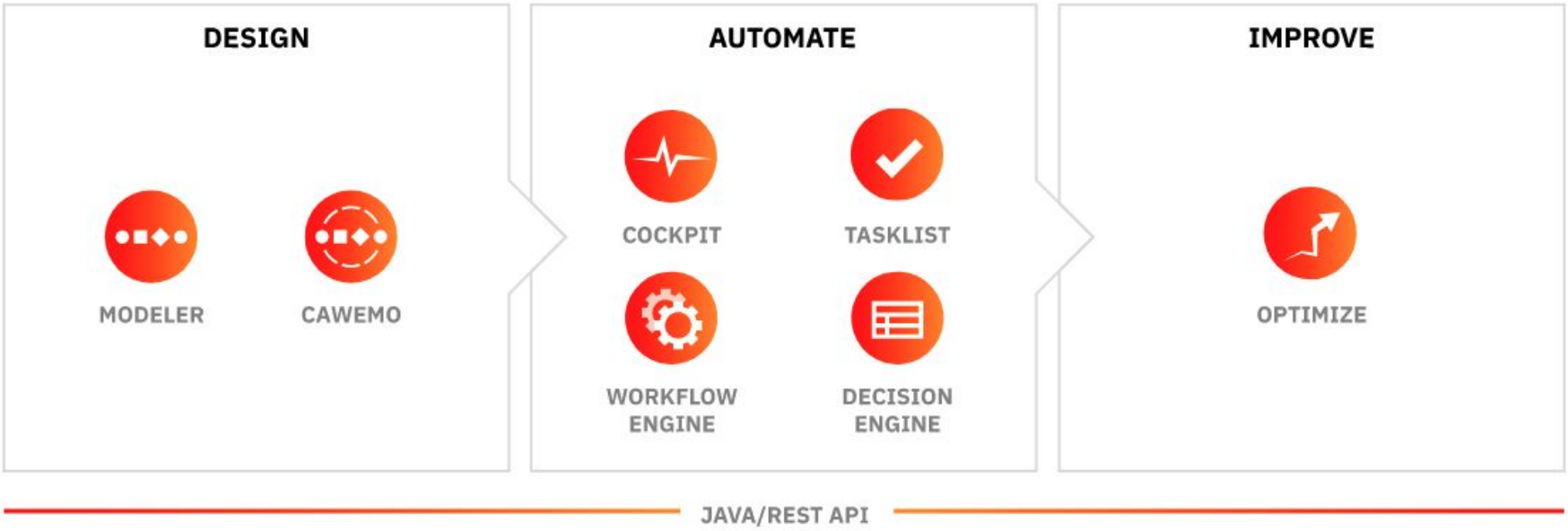
- BPM allow to improve productivity, lowered expenses, mitigate risks and Move toward digital transformation

Camunda Overview

Key Infos

- March 2013, Camunda forked the Activiti project (Jakob Freund and Bernd Rücker) to launch Camunda BPM as an open-source project.
- Camunda BPM is a lightweight, Java-based framework for BPM - “Developer Friendly”
- Current version 7.14.0

Project Life Cycle

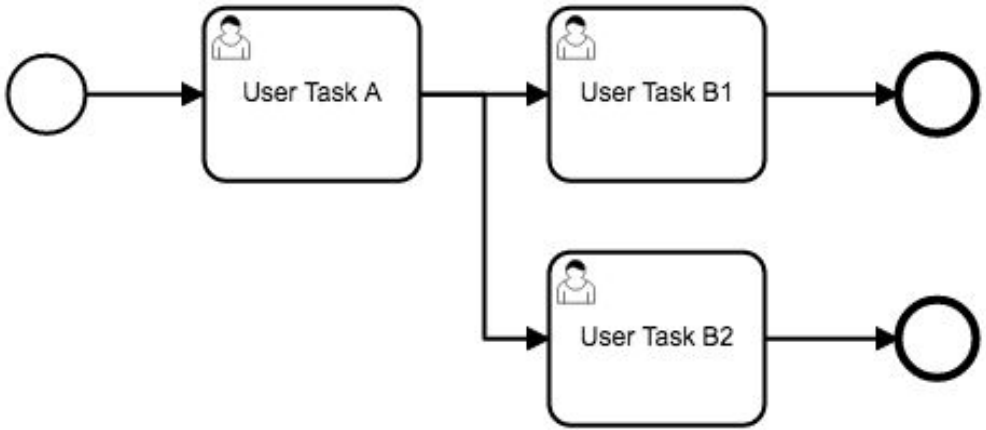


BPMN In Details

BPMN Gateways

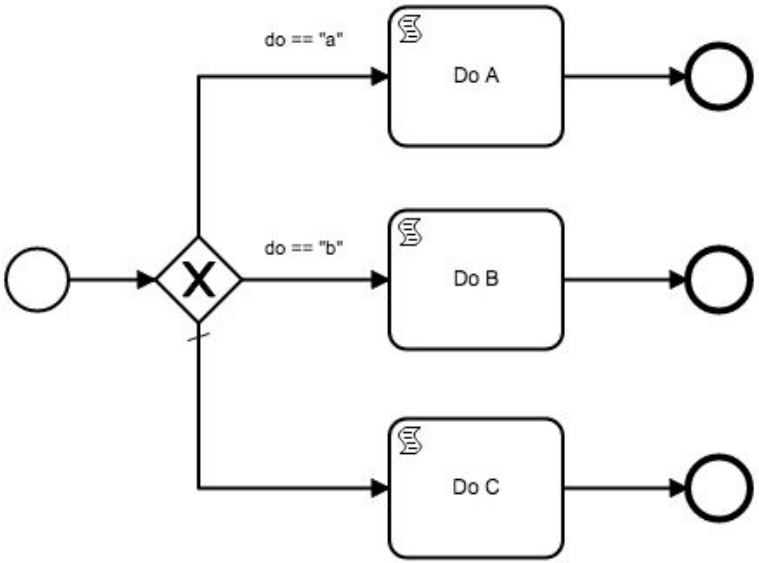
Conditional Sequence Flow

All outgoing sequence flows are followed. This means that the default nature of BPMN 2.0 is to be parallel.



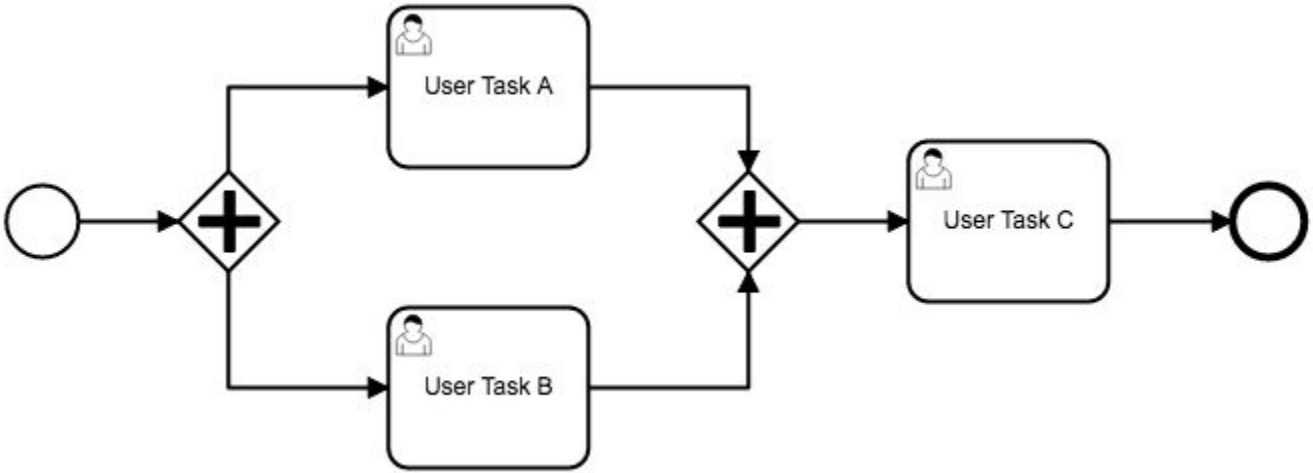
Data-based Exclusive Gateway

Only one sequence flow is selected when using the exclusive gateway. In case multiple sequence flow have a condition that evaluates to ‘true’, the first one defined in the XML is exclusively selected for continuing the process.



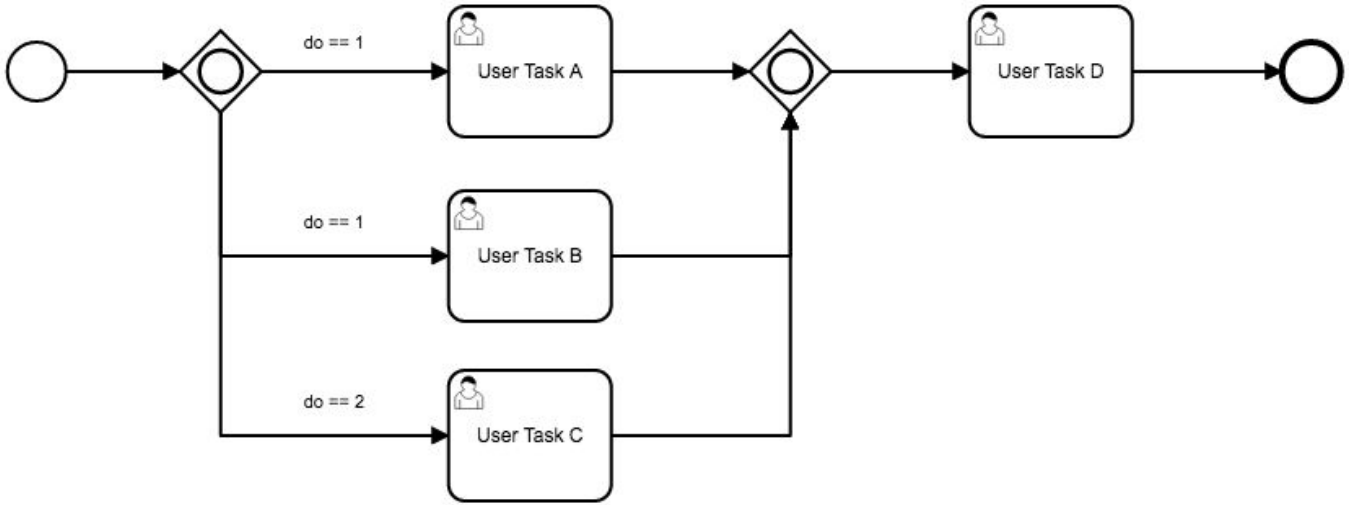
Parallel Gateway

Fork: all outgoing sequence flows are followed in parallel, conditions are not evaluated.
Join: all concurrent executions arriving at the parallel gateway wait at the gateway until an execution has arrived for each of the incoming sequence flows.



Inclusive Gateway

Fork: all outgoing sequence flow conditions are evaluated and for the sequence flow conditions that evaluate to ‘true’, the flows are followed in parallel.
Join: the inclusive gateway will only wait for the incoming sequence flows that are executed.



BPMN Tasks

BPMN Tasks

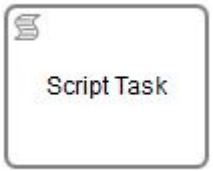
A Service Task is used to invoke services.

A Send Task is used to send a message. The Send Task has the same behavior as a Service Task.

A User Task is used to model work that needs to be done by a human actor.

BPMN Tasks

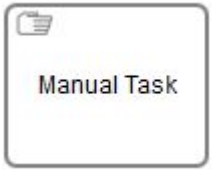
A Business Rule Task is used to synchronously execute one or more rules.



A Script Task is an automated activity. When a process execution arrives at the Script Task, the corresponding script is executed (Groovy, JavaScript, JRuby and Jython).

A Receive Task is a simple task that waits for the arrival of a certain message.

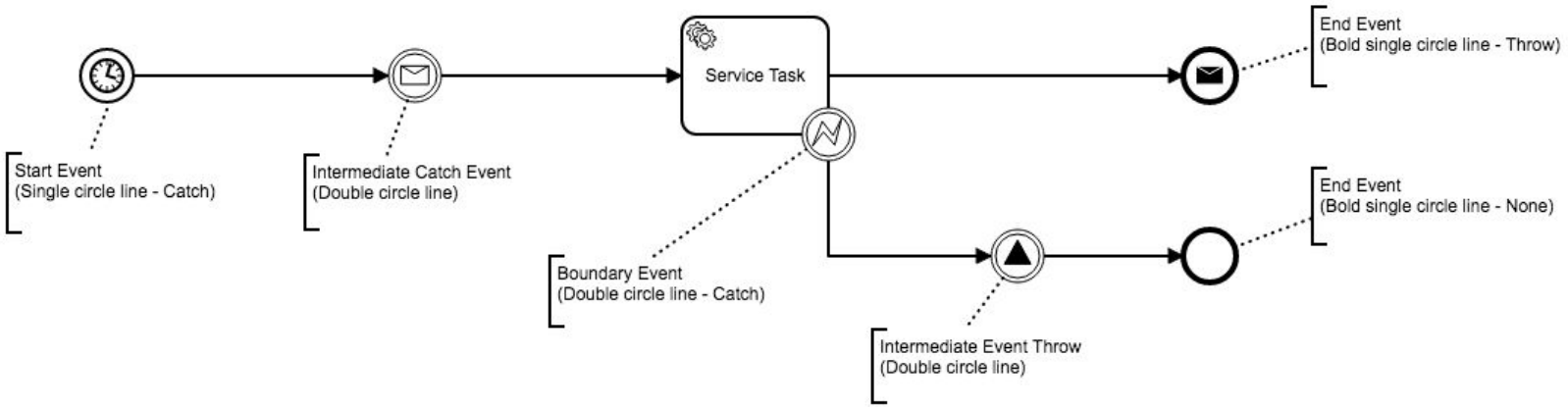
BPMN Tasks



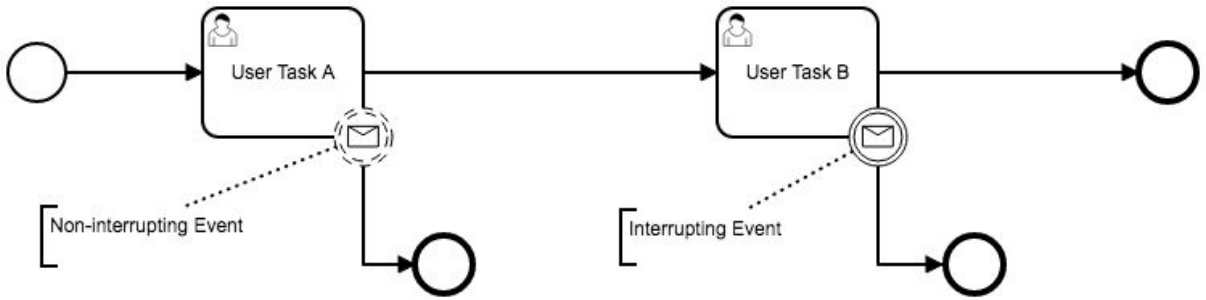
A Manual Task defines a task that is external to the BPM engine. It is handled as a pass-through activity, automatically continuing the process at the moment the process execution arrives at it.

BPMN Events

BPMN Events Positions

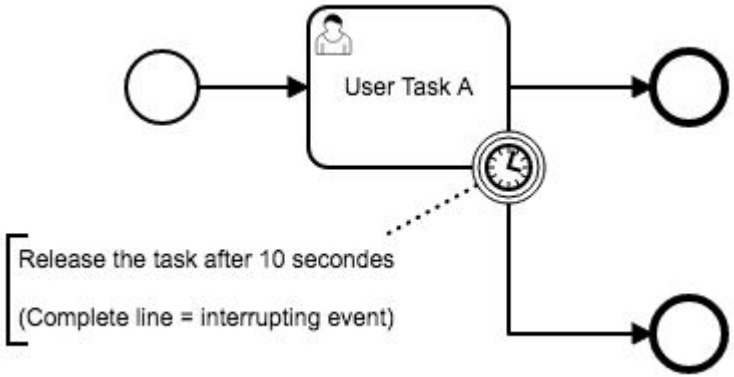


Interrupting & Non-interrupting Events



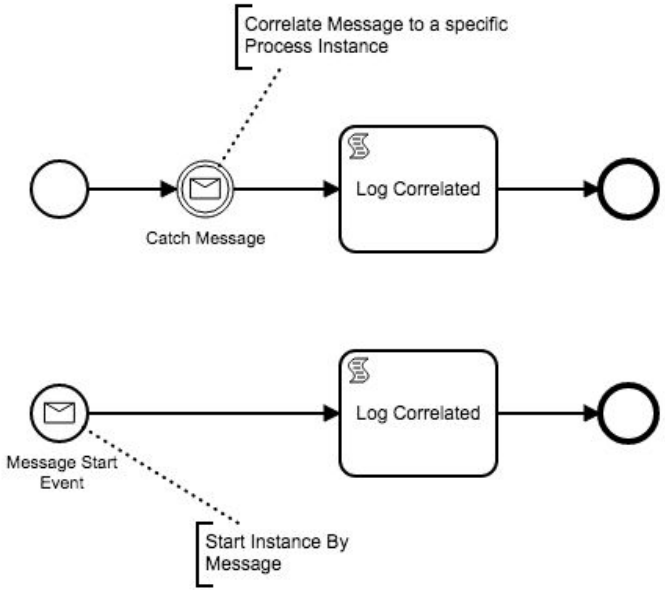
Timer Events

Timer events are events which are triggered by a defined timer (*ISO 8601*). They can be used as start event, intermediate event or boundary event. Boundary events can be interrupting or not.



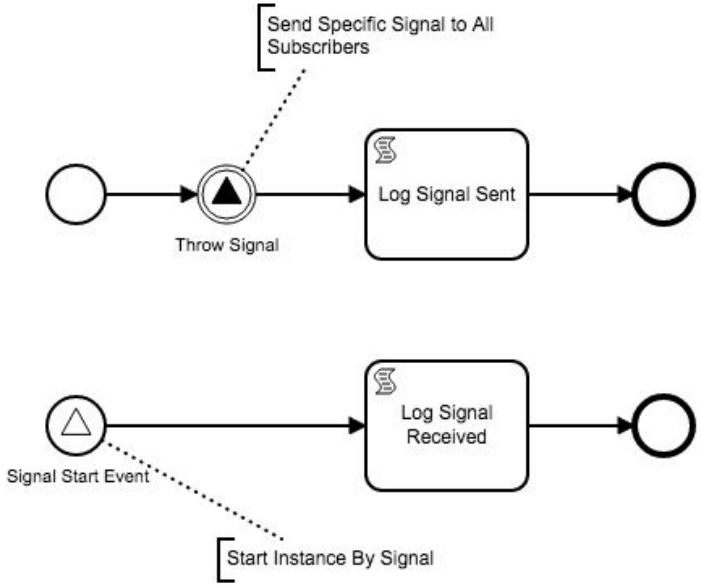
Message Events

Message events are events which reference a named message. A message has a name and a payload. Unlike a signal, a message event is always directed at a single recipient.



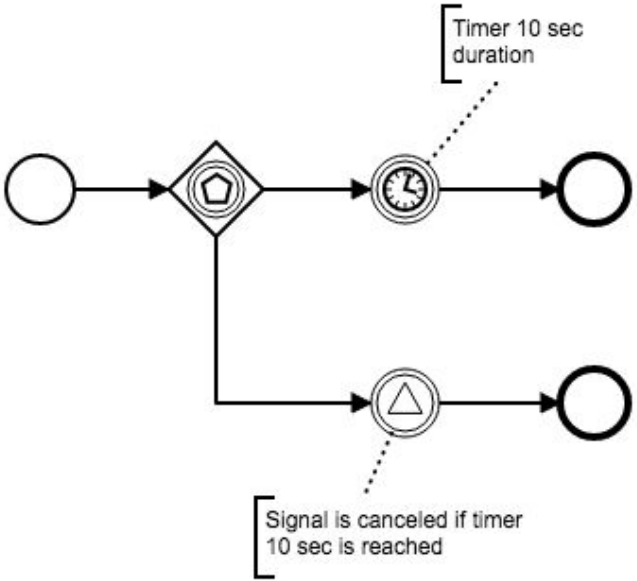
Signal Events

Signal events are events which reference a named signal. A signal is an event of global scope (*broadcast semantics*) and is delivered to all active handlers.



Event-based Gateway

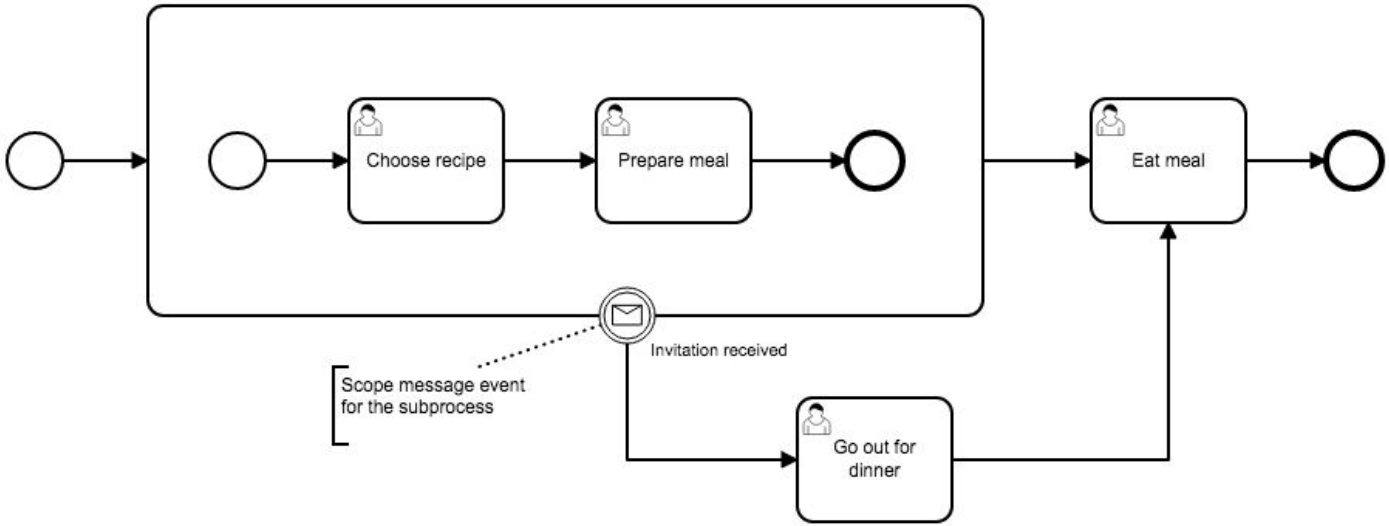
The event-based Gateway allows you to make a decision based on events. Each outgoing sequence flow of the gateway *(must have 2 or more)* needs to be connected to an intermediate catching event.



BPMN Subprocess

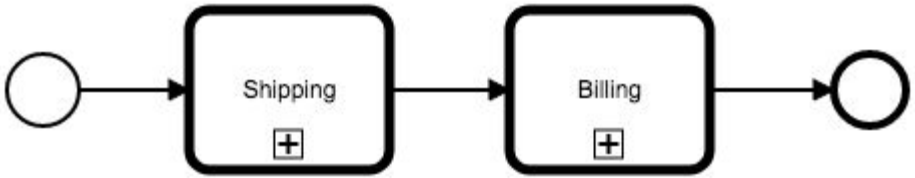
Embedded Subprocess

A subprocess is an activity that contains other activities, gateways, events, etc., which itself forms a process that is part of a bigger process.
Subprocesses have two major use cases: hierarchical modeling and to create a new scope for events.



Call Activity

Call activity allow to reference a process that is external to the process definition. The main use case for the call activity is to have a reusable process definition that can be called from multiple other process definitions.

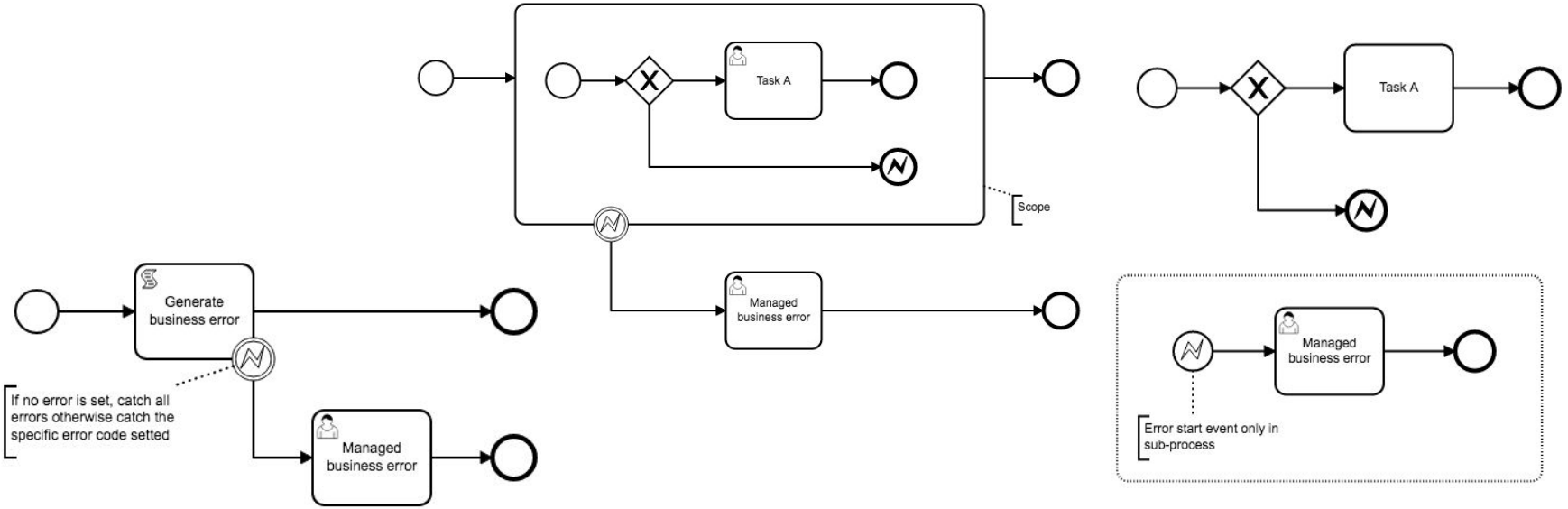


Advanced Development with Camunda

BPMN Error Event

Error Events

A BPMN error is meant for business errors - which are different than technical exceptions.



Bonus

Useful URLs

- BPMN 2.0 poster : http://www.bpmb.de/images/BPMN2_0_Poster_EN.pdf
- Camunda documentations : https://docs.camunda.org
- Camunda Best Practices : https://camunda.com/best-practices
- Camunda project initializer : https://start.camunda.com
- Camunda Blog : https://camunda.com/blog
- GitHub : https://github.com/camunda
- BPMN.io : https://bpmn.io