IN5140

Smart processes and agile methods in Software Engineering

Group session 2

Agenda

First hour:

- Some general info
- A small repetition of BPMN
- Solve weekly task

Second hour:

- Interactive live solving of the weekly task

Remember to submit project group

- Size of groups: 3-5 students.

- Either you want us to put you in a group, or you want to form group yourselves, you need to submit this form.
- The form closes **this sunday** @ 23:59.
- If you have not filled out the form by the deadline, you will be placed in an arbitrary group.

BPMN

Business Process Modelling Notation

“A project process model defines **Who** is doing **What**, **When** and **How** to reach a specific goal”

...this is what we want to visualize with BPMN!

Why use BPMN?

- BPMN is a flow chart method that models the steps of a planned **business process** from end to end.
- Used for Business Process Management,
- Visually depicts a detailed sequence of **business activities and information flows** needed to **complete a process**.

BPMN - tools

- Diagrams.net *(previously Draw.io)*
  - https://www.diagrams.net/
- **Lucidchart**
  - https://www.lucidchart.com/
- yEd Graph Editor
  - https://www.yworks.com/downloads#yEd
- Yaoqiang BPMN Editor - Open Source
  - https://sourceforge.net/projects/bpmn/
- Signavio
  - https://www.signavio.com/ (free 30-day trial)
- Pen and paper

Pools and Swimlanes

- **A pool** represents major participants in a process. A different pool may be in a different company or department but still involved in the process.

- **Swimlanes** within a pool show the activities and flow for a certain role or participant, defining who is accountable for what parts of the process.

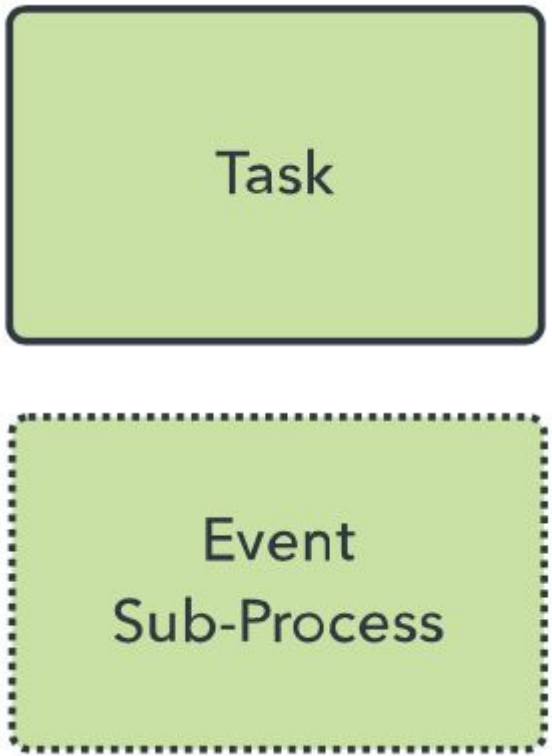
(Taken from Lucidchart)

Events

- A trigger that starts, modifies or completes a process.
- They are shown by circles containing other symbols based on event type.

Activities

- A particular activity or task performed by a person or system. It’s shown by a rectangle with rounded corners.
- They can become more detailed with for example sub-processes.



Gateways

- Decision point that can adjust the path based on conditions or events. They are shown as diamonds.
- Gateways can also be an empty diamond

Flow

- **Sequence flow:** shows the order of activities to be performed

- **Message flow:** depicts messages that flow across “pools,” or organization boundaries such as departments. It shouldn’t connect events or activities within a pool.

How to engage BPMN

- **Skim through** the textual description
  - Try not to drown in the text
- Note potential **pools**
- Identify the **roles** of each pool
  - Swim lanes
  - What are their tasks
  - What are their dependencies
- Model the **tasks** with metadata
- Check that you are still answering the exercise
- Do not make the BPMN diagram too extensive and complicated

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.

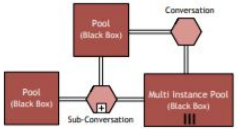
Conversations

A **Conversation** defines a set of logically related message exchanges. When marked with a [+] symbol it indicates a **Sub-Conversation**, a compound conversation element.

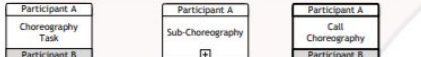
A **Call Conversation** is a wrapper for a globally defined Conversation or Sub-Conversation. A call to a Sub-conversation is marked with a [+] symbol.

A **Conversation Link** connects Conversations and Participants.

Conversation Diagram



Choreographies



A **Choreography Task** represents an interaction (Message Exchange) between two Participants.

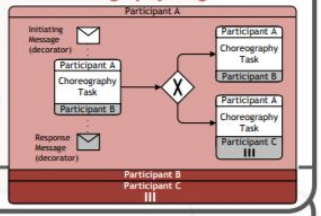
A **Sub-Choreography** contains a refined choreography with several Interactions.

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.

Multiple Participants Marker — denotes a set of Participants of the same kind.

Message — a decorator depicting the content of the message. It can only be attached to Choreography Tasks.

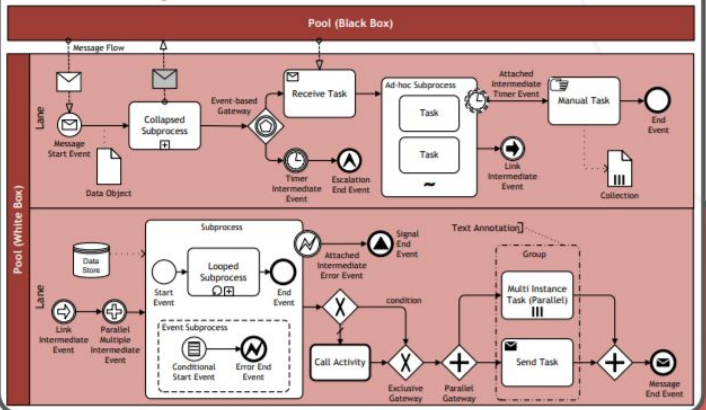
Choreography Diagram



Events

	Start: Standard	Start: Event Sub-Process Interrupting	Start: Event Sub-Process Non-Interrupting	Intermediate: Catching	Intermediate: Boundary Interrupting	Intermediate: Boundary Non-Interrupting	Intermediate: Throwing	End: 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.								●

Collaboration Diagram



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.

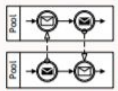
Complex Gateway — Complex merging and branching behavior that is not captured by other gateways.

Parallel Event-based Gateway (instantiate) — The occurrence of all subsequent events starts a new process instance.

Swimlanes

Pools (Participants) and **Lanes** represent responsibilities for activities in a process. A pool or a lane can be an organization, a role, or a system. Lanes subdivide pools or other lanes hierarchically.

Message Flow symbolizes information flow across organizational boundaries. Message flow can be attached to pools, activities, or message events. The Message Flow can be decorated with an envelope depicting the content of the message.



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

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.

BPM Offensive Berlin

Hasso Plattner Institut

camunda

inubit

SIGNAVIO

What we value in your assignments

Syntax:

- Correct lanes and pools
- Activities vs gateways
- Start and stop event
- Sequence flow
- Collaborations

In general:

- Your diagram should be readable and understandable

BPMN modeling

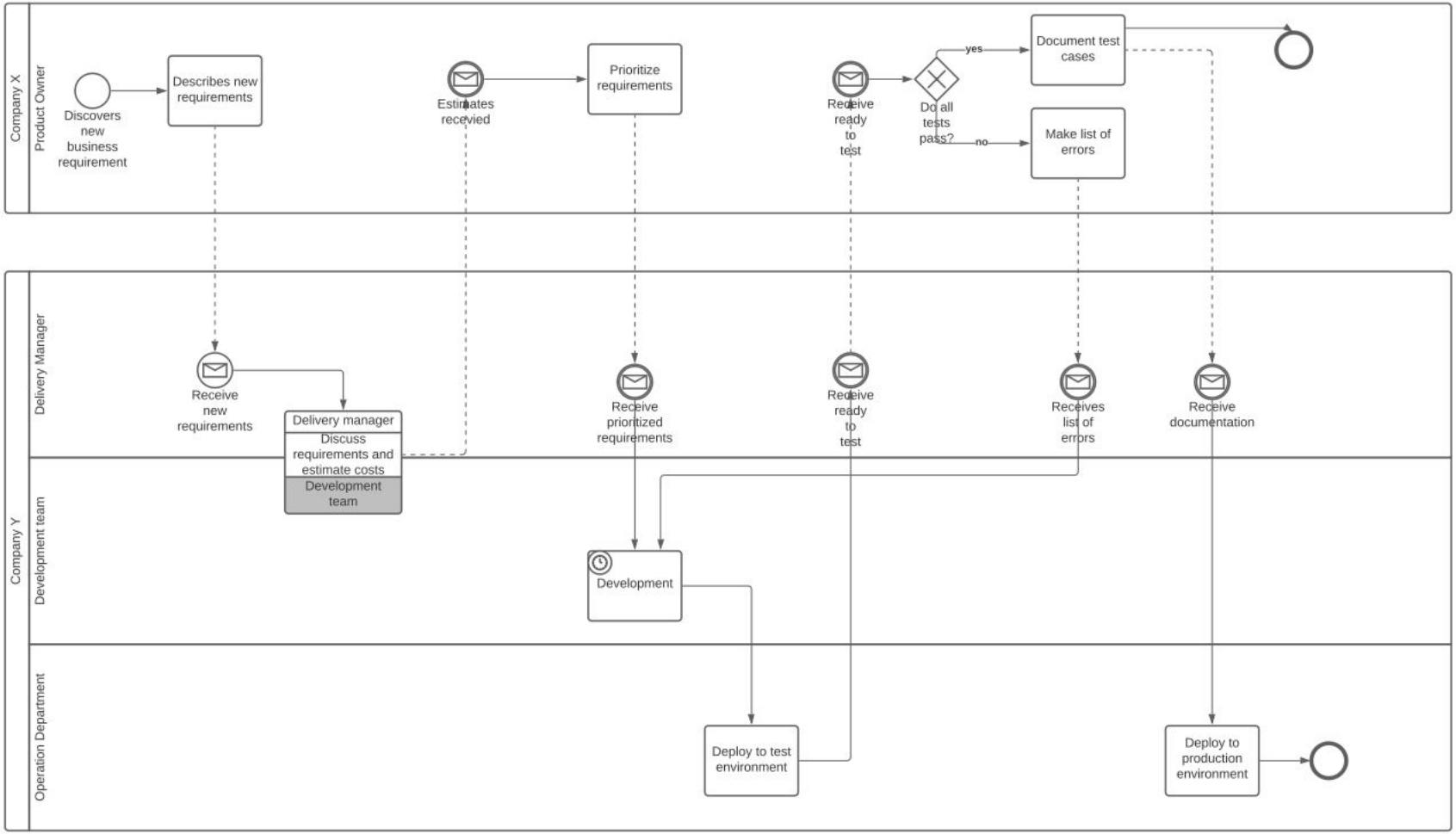
- There are many different ways to model BPMN-diagrams
- No “clear solution” as long as you are answering the exercise

Discuss and solve the weekly exercise in groups!

The exercise is published on the course page.

Break ☕
We'll start again at 1315 🕰️

Live interactive solution!


Company X
Product Owner
Discovers new business requirement
Describes new requirements
Estimates recevied
Prioritize requirements
Receive ready to test
Do all tests pass?
yes
no
Document test cases
Make list of errors
Company Y
Delivery Manager
Receive new requirements
Delivery manager
Discuss requirements and estimate costs
Development team
Receive prioritized requirements
Receive ready to test
Receives list of errors
Receive documentation
Development team
Development
Operation Department
Deploy to test environment
Deploy to production environment

This video saved our exam! 🤩

Questions?

Next Week:

Standard software process models & frameworks for process improvements