OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG

INFORMATIK UND MATHEMATIK

Modeling and Verification (MOV)

Business Process Model and Notation (Introduction to BPMN and its semantics)

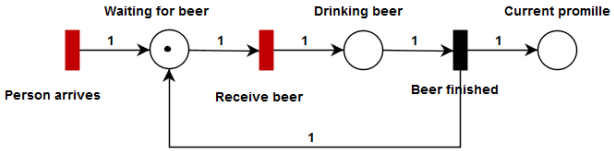
Prof. Dr. Carsten Kern

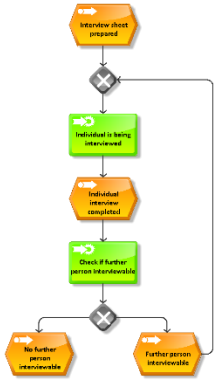
Ostbayerische Technische Hochschule
Fakultät Informatik und Mathematik

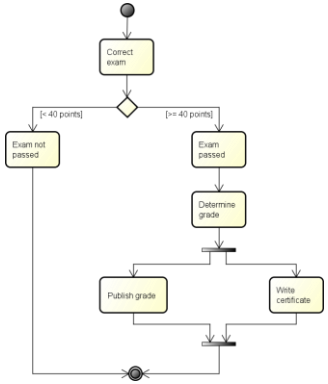
Last time …

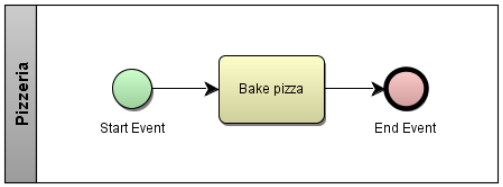
We learned:

- What is modeling?
- What is modeling good for?
- Which models are in the scope of this lecture?









- What is a business process?
- What is business process modelling and management?
- …

What do we have to do to create bad models? (positive formulations)

Think about the question in groups of 3-5 and collect phrases. (3 minutes)

Now …

We will detail on:

- The BPMN itself,
- Its notation,
- Its basic elements
- Its semantics

Introduction to BPMN

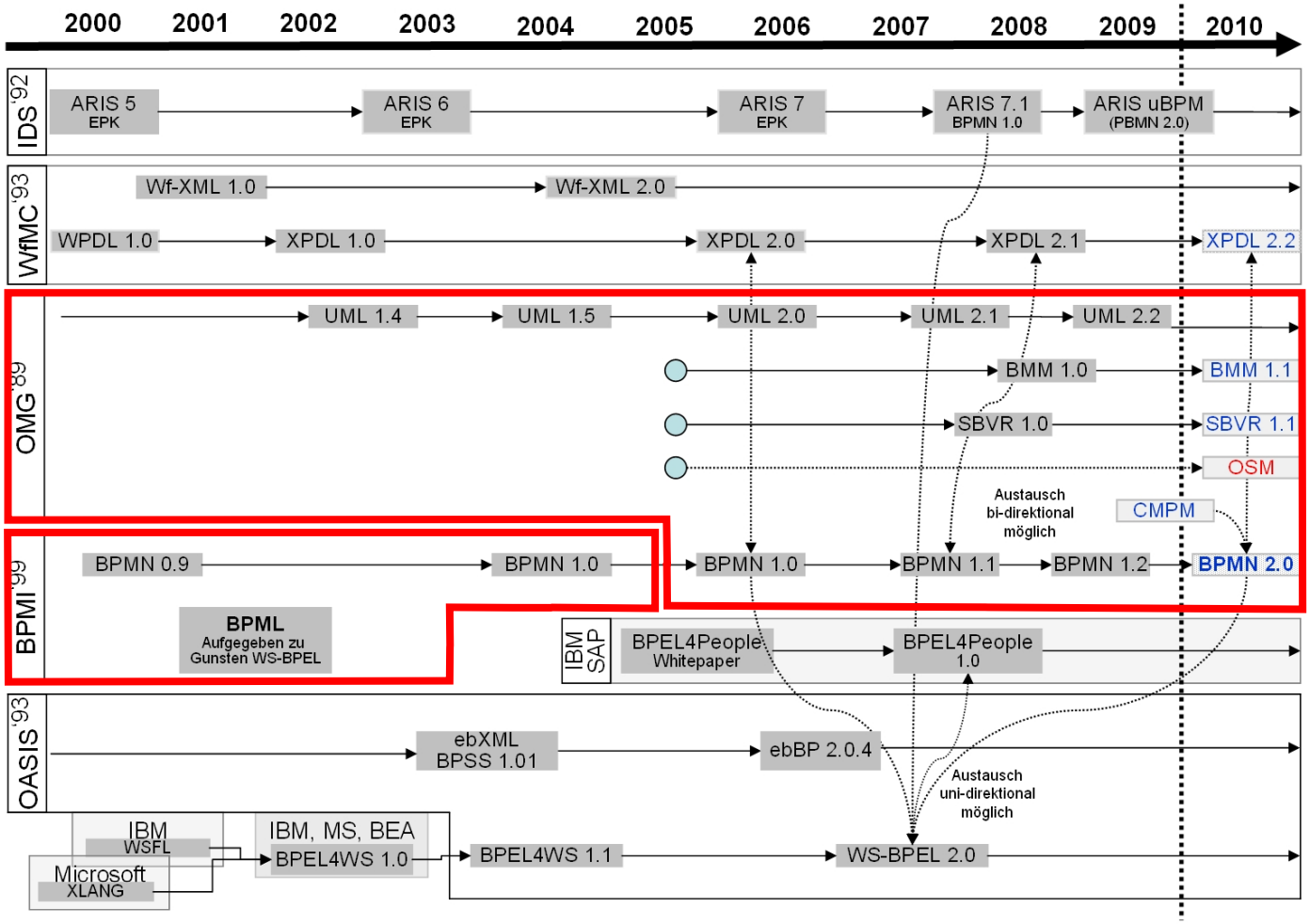
History, motivation,
basic elements and semantics

Business Process Model and Notation

Business Process Model and Notation:

- A graphical notation for specifying (business) processes in a (business) process model
- The syntax (and semantics) are closely related to UML activity diagrams
- Developed in 2001 by Stephen A. White (IBM)
- Published in 2004 by the Business Process Management Initiative (BPMI)
- Since 2005 maintained by the Object Management Group (OMG)
  - Since 2006 official release (version 1.0)
  - 2008 version 1.1
  - 2009 version 1.2
  - 2011 version 2.0 (formal description of process execution: *execution semantics* and other extensions)
  - 2013 BPMN is becoming ISO-standard
- It is a standard for (business) process modeling

BPMN and other modeling languages



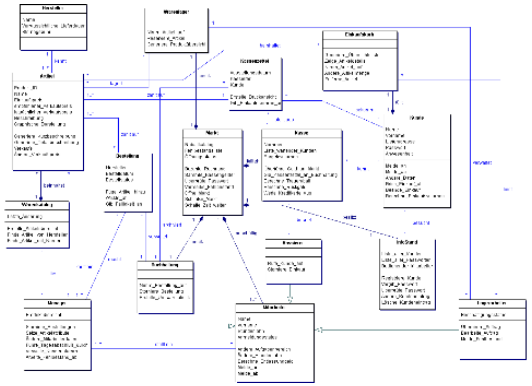
Dr. Martin Bartonitz, Nov. 2009

Main goal of BPMN according to Stephen A. White

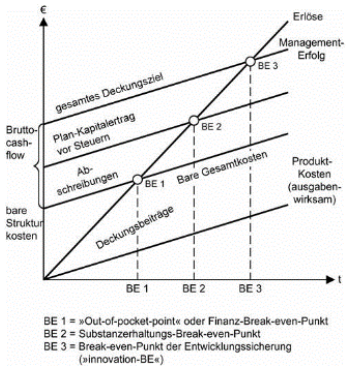
"The primary goal of the BPMN effort was to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes. BPMN will also be supported with an internal model that will enable the generation of executable BPEL4WS. Thus, BPMN creates a standardized bridge for the gap between the business process design and process implementation."

The BPMN is aiming at…

- Comprehensibility (using a simple graphical representation instead of a complex mathematical notation)
- Simplicity (widely free of implementation details)
- Clarity (despite complex business processes)
- Easy semantics (understandable for "*non-geeks*")
- Being basis for communication about business processes between different divisions



vs.

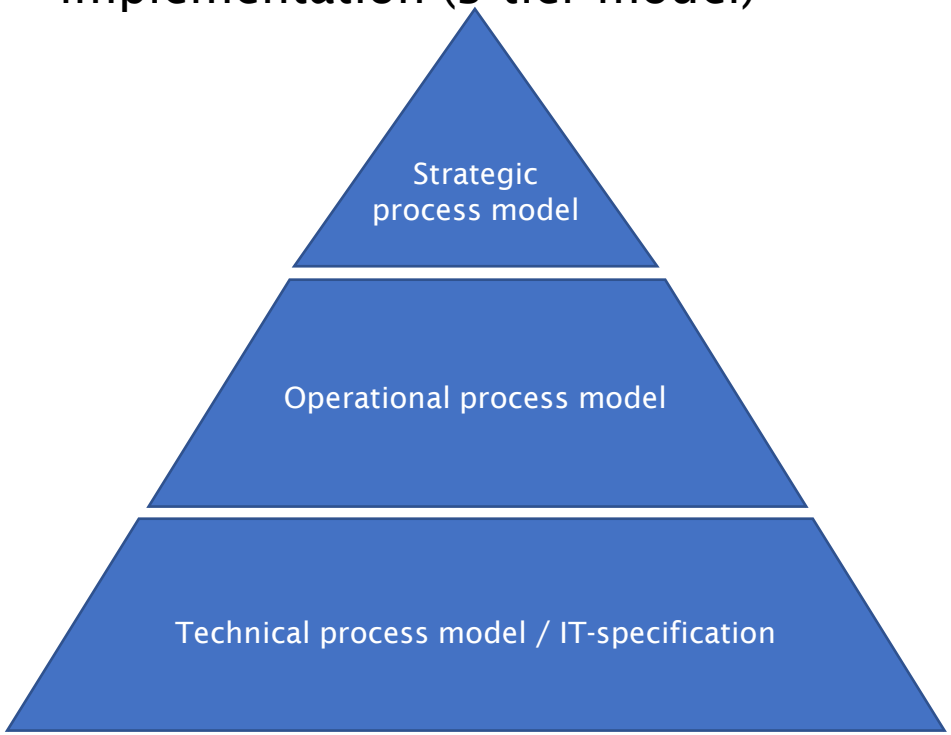


- Documenting processes for quality management

What are advantages of using such a language wrt. the people involved?

Using BPMN at different layers of abstraction

- Graphical specification language for modeling, documenting, measuring, optimizing and executing business processes (workflows)
- It focuses on the interfaces between high- and low-level design and implementation (3-tier model)



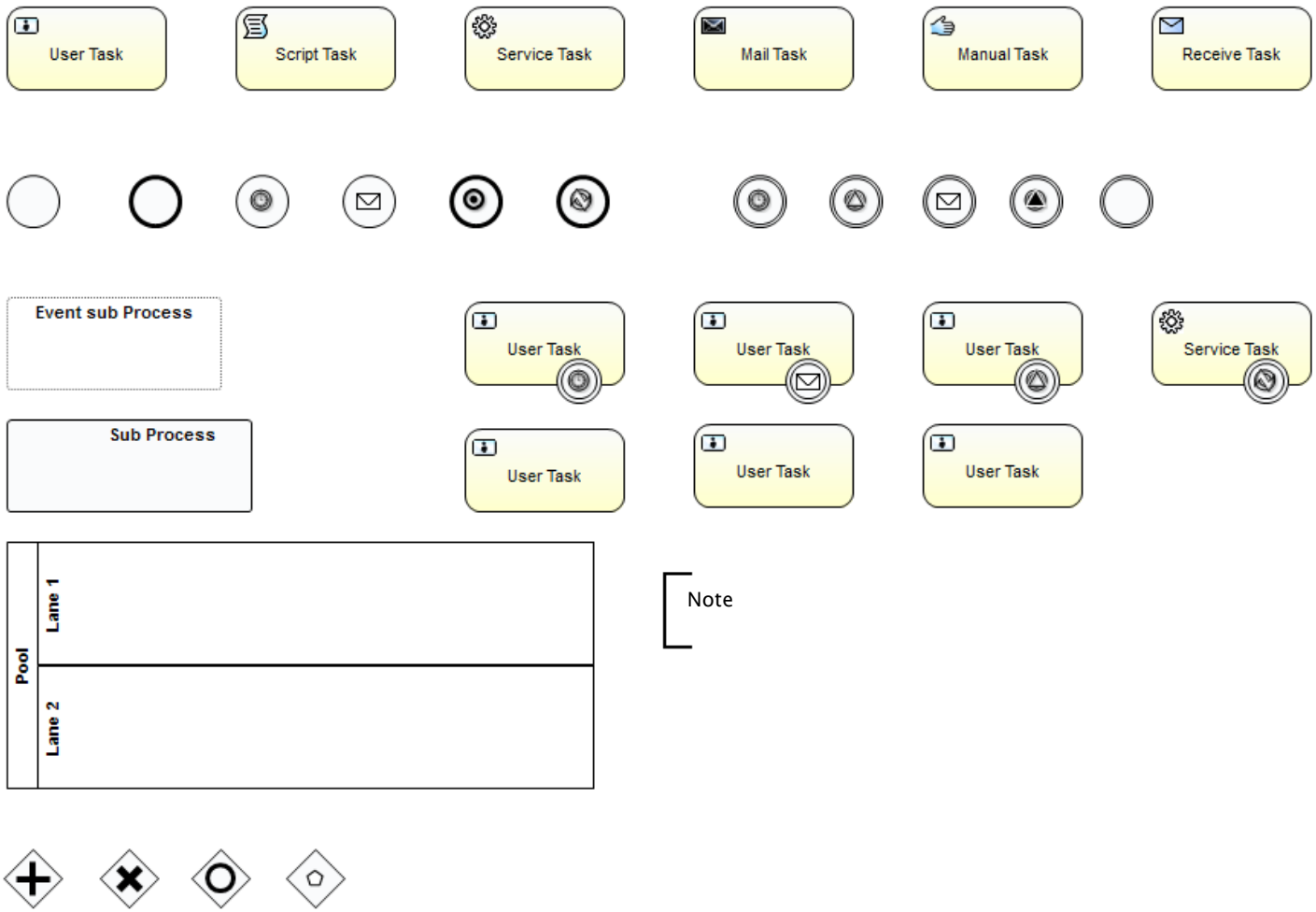
Gain insights about process
Focus on fast comprehension

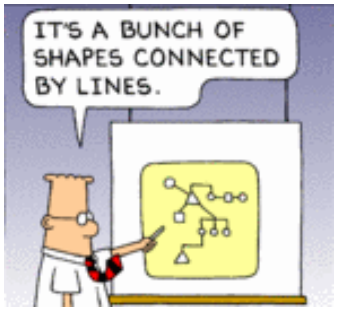
Operational activities
Clarify details

Technically detailed
Technical implementation

[cf. Jakob Freund, Bernd Rücker: *Praxishandbuch BPMN 2.0*]

BPMN: basic notation and elements





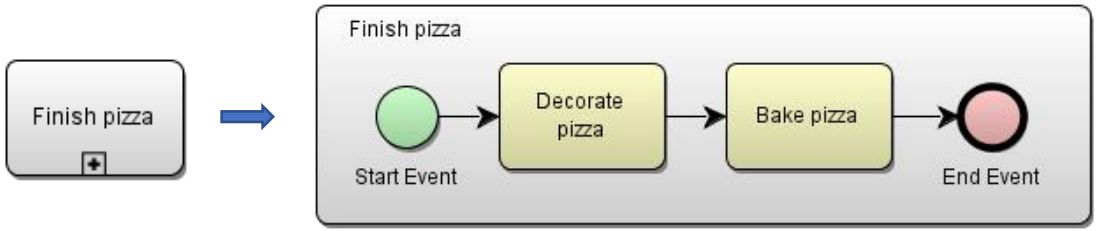
Basic elements: Flow Objects

Activities ((sub-) process, task)

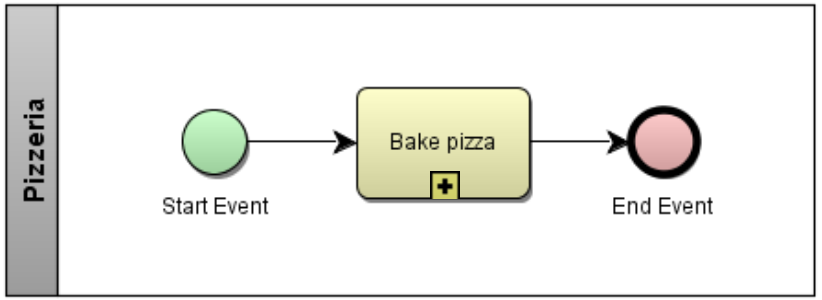
- Task which is to be executed within a business process
- Notation: rectangle with rounded corners
- Convention: verb + object (e.g.: "*Bake pizza*", "*Cut pizza*")

- Normal Activity/Task:

- Complex Task:



- Process:



Basic elements: Flow Objects

Events (i.e.: start-, intermediate, end-events)

- Denote: point in time during process
- Notation: circle or double circle with or without label
- Convention: object + participle (e.g. "*Mission acknowledged*", "*Email sent*")

- **Start:**

… describes trigger of a process

- **End:**

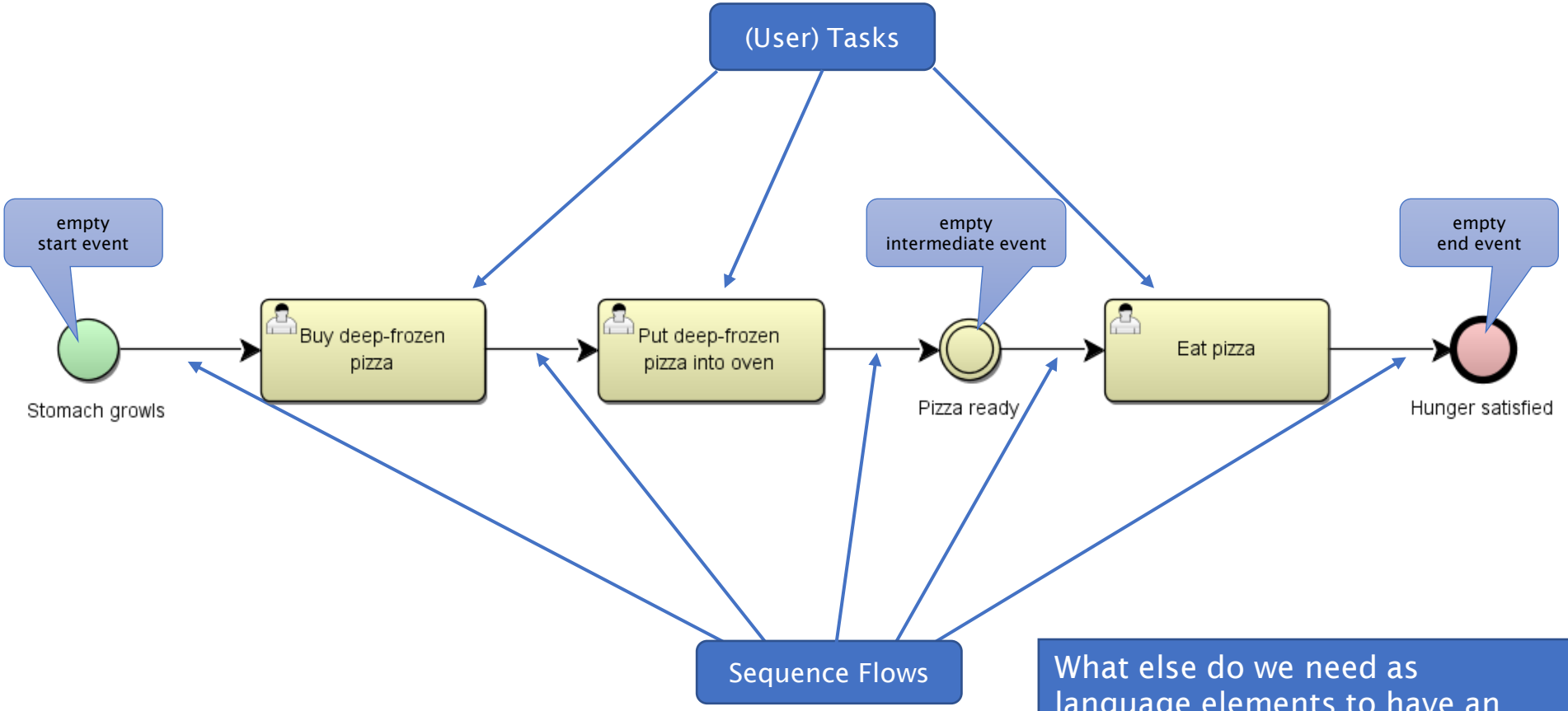
… describes result of a process

- **Intermediate:**

… describes state changes within a process

Exercise: a first example

Model the process: "A typical student dinner"



What else do we need as language elements to have an expressive modeling language ?

Basic elements: Flow Objects

Gateways (decision points)

- Function: defines process flow logic
- Notation: diamond with or without content

- XOR:

exclusive gateway
(i.e.: "*take exactly one outgoing edge*")

- AND:

parallel gateway
(i.e.: "*take all outgoing edges in parallel*")

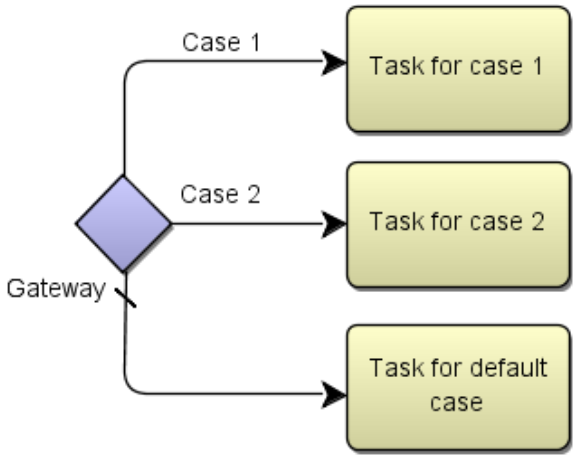
- OR:

inclusive gateway
(i.e.: "*take at least one outgoing edge*")

Basic elements: Flow Objects

Default flows (using (xor-) gateways)

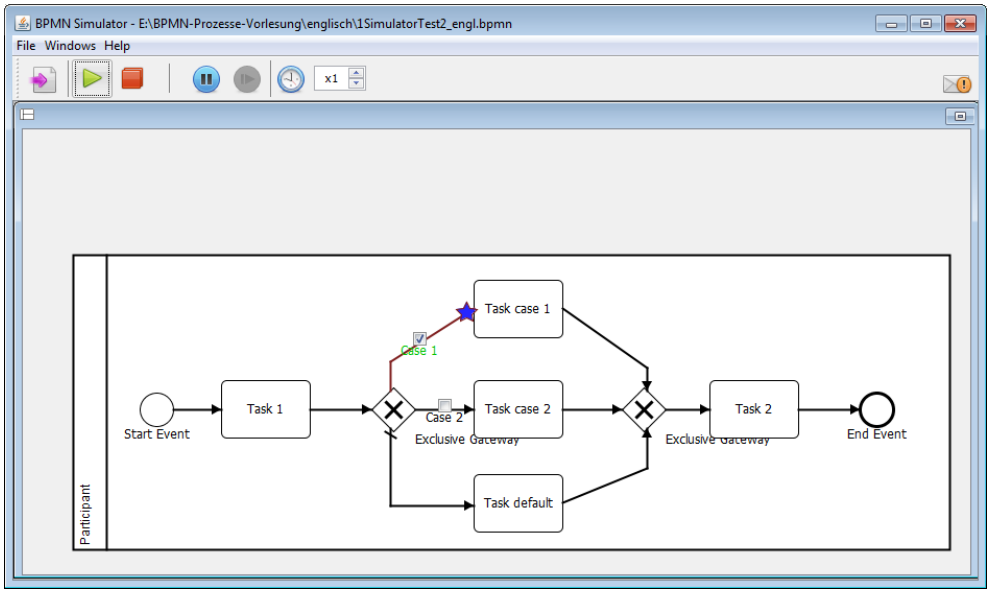
- Gateway:



  - Notation: default edge denoted by a backslash
  - If neither case 1 nor case 2 occur the default edge is automatically taken

BPMN flow semantics (execution semantics)

Flow simulator for basic BPMN diagrams:



- The concept of tokens: (BPMN's *flow-* or *execution semantics*)
  - Token is born in start event
  - It runs through the process
  - It can be cloned into several or several tokens may be joined into one
  - It dies when reaching the end event
  - As long as there is at least one living token, the process instance is still active

BPMN flow semantics (execution semantics)

Branching gateways:

- XOR gateway:

  - If a token arrives, it will be forwarded to exactly one outgoing edge

- OR gateway:

  - If a token arrives, it will be forwarded to at least one outgoing edge
  - If it is forwarded to several edges, it will be split into several tokens first

- AND gateway:

  - If a token arrives, it will be split into as many tokens as outgoing edges exist
  - Each token is forwarded to a different outgoing edge in parallel

BPMN flow semantics (execution semantics)

Merging gateways: (carefully handle problem cases!)

- XOR gateway:

  - A token will be forwarded as soon as it arrives at the merging XOR

- OR gateway:

  - Depending on whether the branching OR has produced one or more tokens, the merging OR will merge or synchronize before forwarding the token

- AND gateway:

(Synchronization)

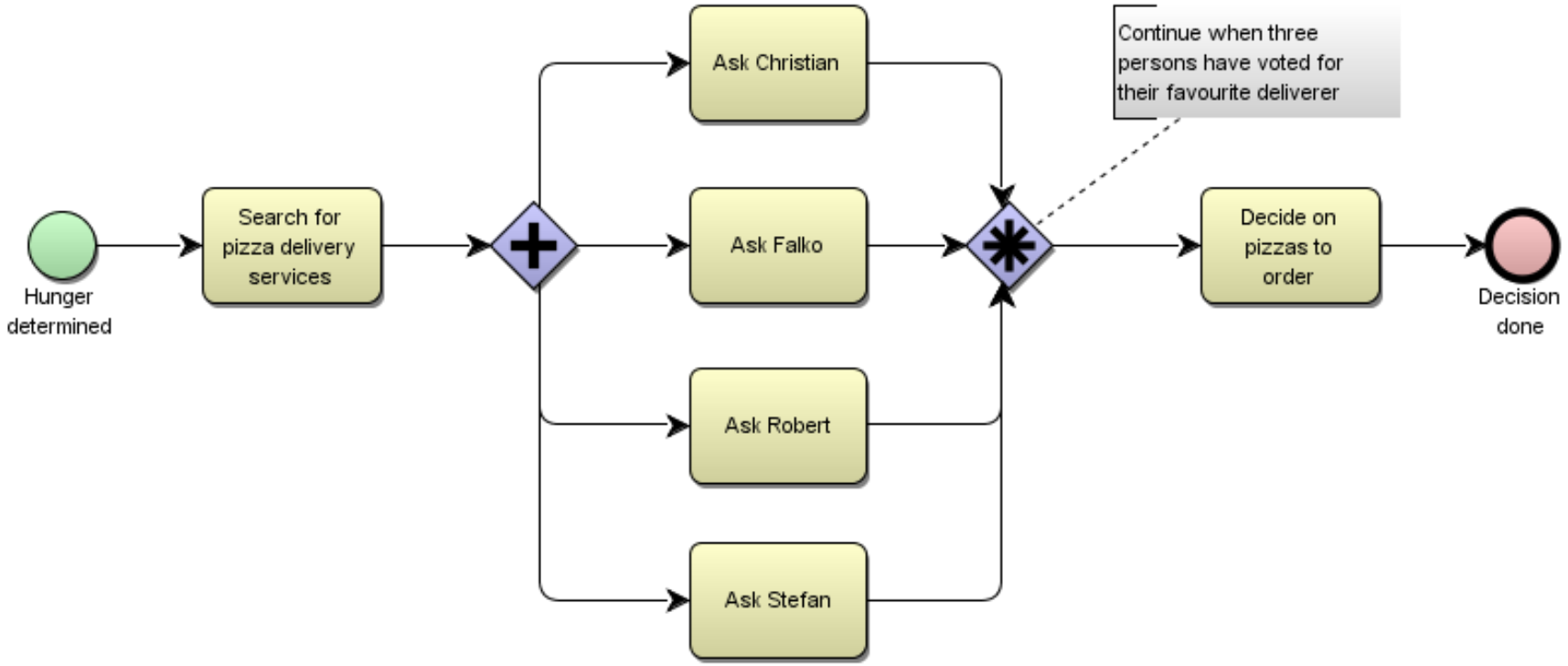
  - One token is forwarded as soon as all ingoing edges are filled with one token

- Complex gateway:

  - If a token arrives, the gateway acts according to the given annotation, e.g.: forward as soon as the first two tokens have arrived

BPMN flow semantics (execution semantics)

Example: complex gateway (merge)

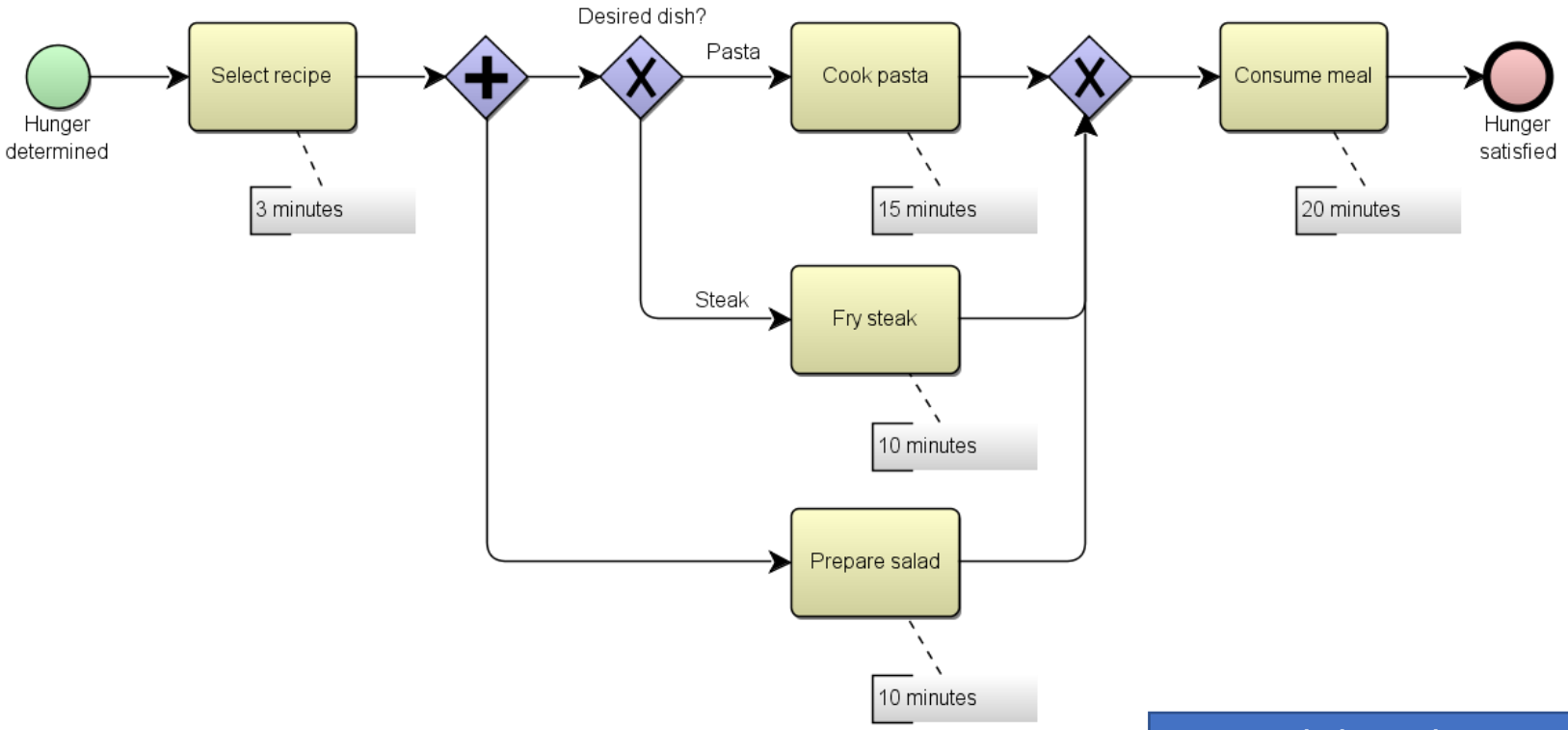


Remark: as branching gateway the complex gateway is (almost) never used

[cf. Jakob Freund, Bernd Rücker: *Praxishandbuch BPMN 2.0* for slides 20-24]

BPMN flow semantics

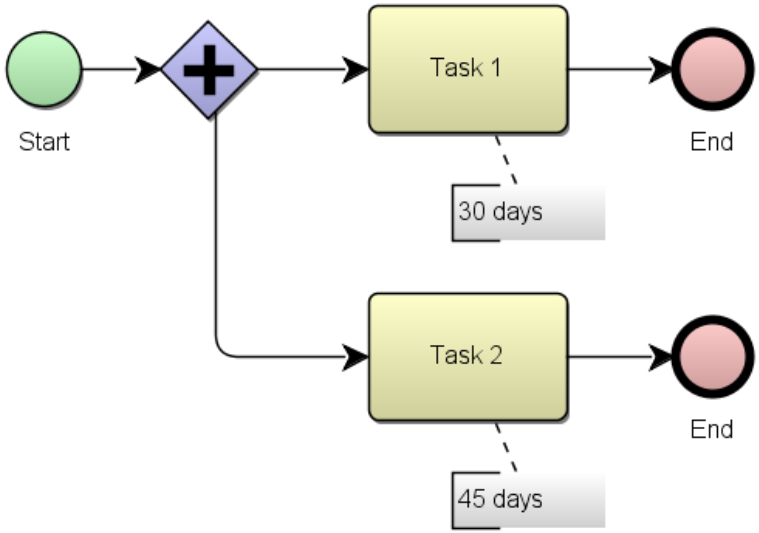
Quiz 1: What happens when executing the following process?



1. Find the solution yourself
2. Discuss results in groups
3. Find a final result

BPMN flow semantics

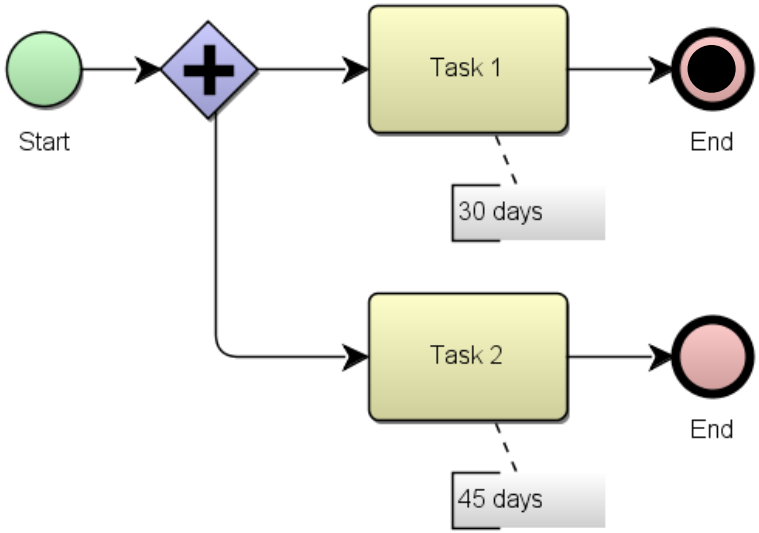
Quiz 2: What happens when executing the following process?



1. Find the solution yourself
2. Discuss results in groups
3. Find a final result

BPMN flow semantics

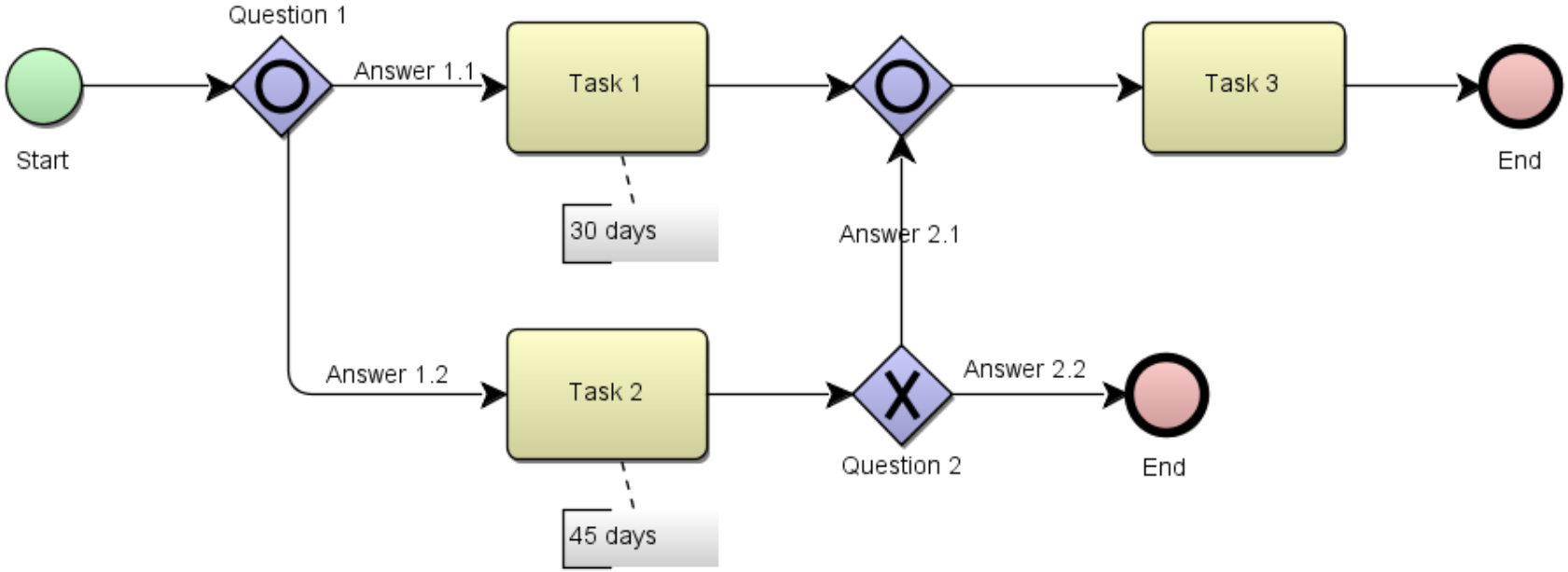
Quiz 3: What happens when executing the following process?



1. Find the solution yourself
2. Discuss results in groups
3. Find a final result

BPMN flow semantics

Quiz 4: What happens when executing the following process?



→ OR merges may be problematic

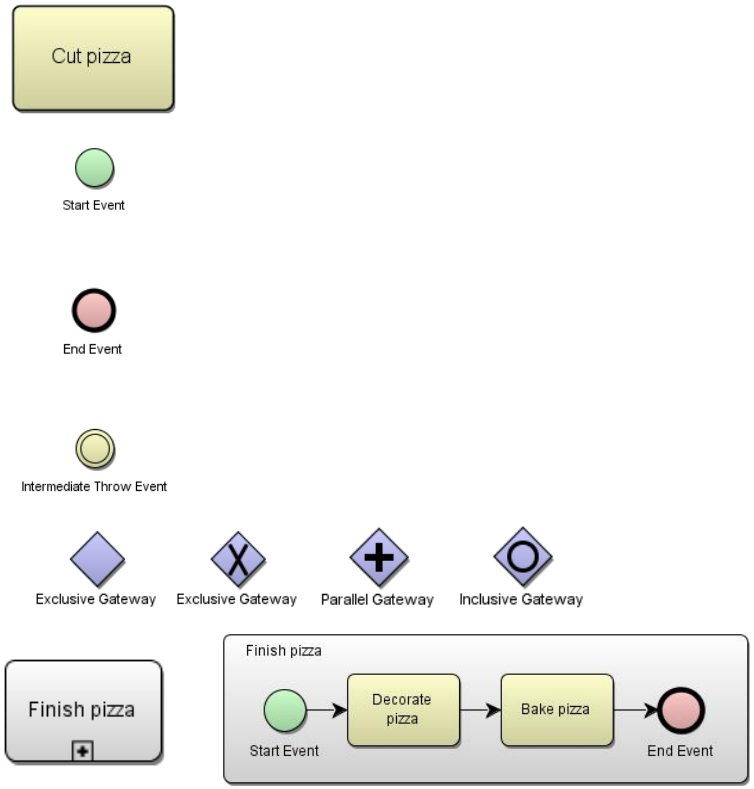
→ Thus, only use them with care!

For how long does the second merging OR gateway have to wait?

Exercises

A well-known example: the (Regensburg-) bouncer process:

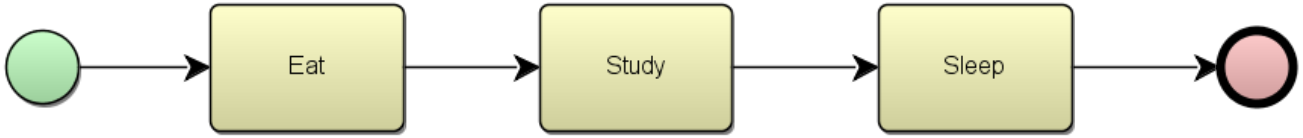
- Model it and try to use all 3 standard gateway types introduced previously
- Model elements to (possibly) include:



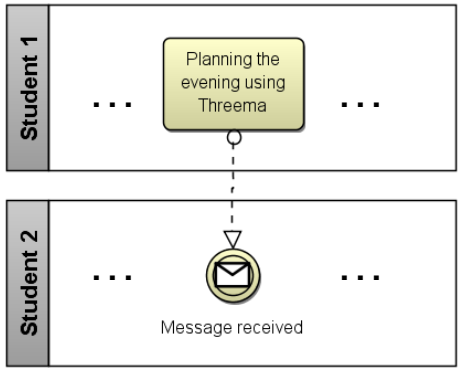
Basic elements: Connecting Objects

Flow:

- Sequence flow:
  - Function: defines the execution order within a pool (cf. later slides)
  - Notation: continuous arrow



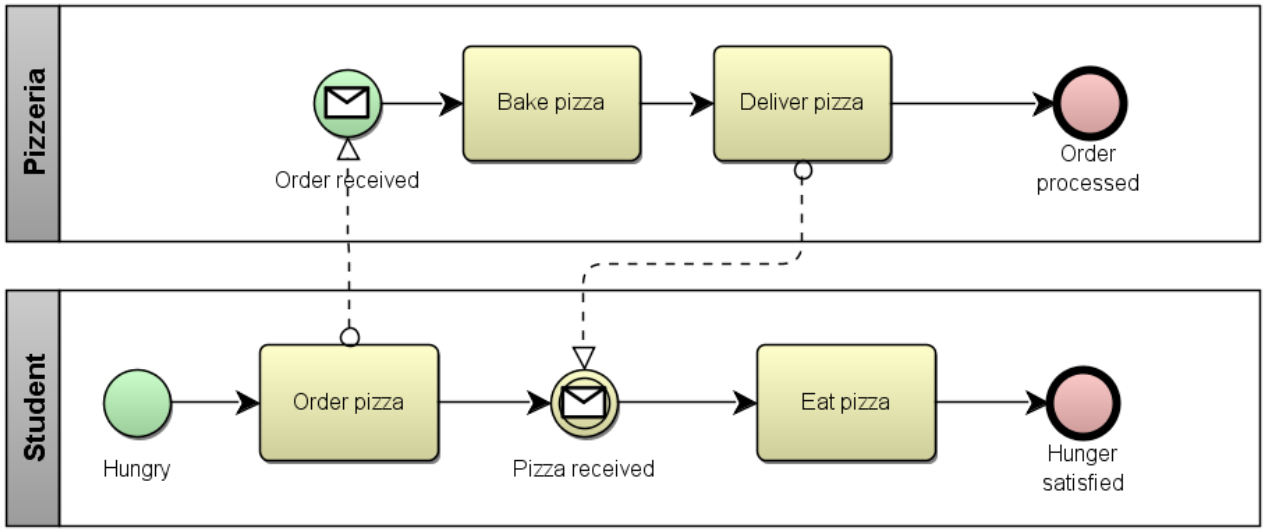
- Message flow:
  - Function: message exchange between pools
  - Notation: dashed arrow



Basic elements: Structuring Objects

Pools (Participants)

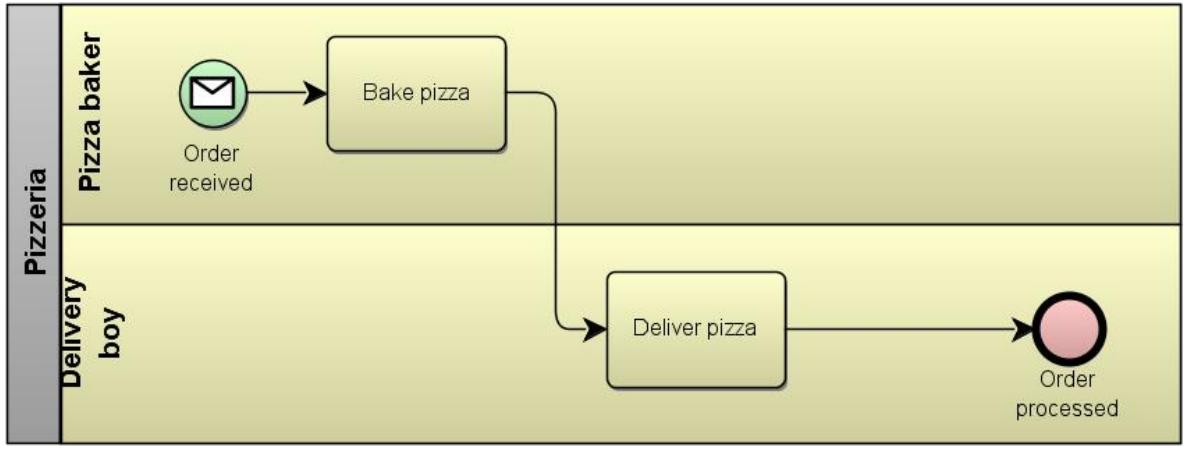
- Function: graphical container for a process
  boundary of responsibility/competence/influence
- Notation: horizontal (or vertical) rectangles labeled with a name on the left side (or top side in case of vertical pools)



Basic elements: Structuring Objects

Lanes (partition of a pool/lane into sub divisions)

- Function: only for visual structuring within a process (pool)
- Notation: embedded into a pool or lane (i.e.: lanes may be hierarchical)



Exercise: "Considering a credit application"

Model the following business process in BPMN:

- A bank is considering a credit application
- An application can be granted or refused
- If an application is granted the bank develops a financing concept and afterwards a draft treaty
- If an application is refused, the bank writes a refusal to the customer
- Finally, the bank informs the applicant about her decision

A plethora of (freely available) editors

Desktop applications:

- Bizagi Process Modeler
- Intalio BPMS (Community Edition)
- Camunda Modeler
- Activiti Modeler (enthalten in Activiti)
- Yaoqiang BPMN Editor
- Eclipse BPMN Modeler
- iyopro (Improve Your Processes)
- Aris Express
- BOC ADONIS (Community Edition 3.0)
- Visio Plugins (z.B. interfacing.com)
- The Oryx Project (- 30.09.2011)

Web-Tools:

- Camunda (https://demo.bpmn.io)
- Gliffy (https://www.gliffy.com)
- GenMyModel (https://www.genmymodel.com/bpmn-online-tool)
- …