2. BPMN

2.2 Ausdrucksmittel
1. Activities
Activity

- **Activity** is a generic term for work that has to be performed
- An activity can be atomic or non-atomic (compound)
- The types of activities that are a part of a Process Model are:
  - Process
  - Subprocess
  - Task
- Tasks and subprocesses are rounded rectangles
- Processes are either unbounded or contained within a pool
Operational Semantics

- BPMN operational semantics is based on passing *tokens*

- An activity or gateway (see later…) is performed when “required” tokens arrived at the activity or gateway

- “Required” depends on…
  - the gateway governing the activity (see later)
  - the specified *start quantity* defined
  - tokens still on their way (*global semantics*! – see later!)
  - …

- At completion, an activity or gateway produces tokens on all of its outbound sequence flows (aka control connectors – see later…)
  - Conditions that evaluate to false consume these tokens immediately
Operational Semantics (cont.)

- A start event generates a token for each of its leaving sequence flows
- These tokens must eventually be consumed at end events
- If parallel sequence flow target an end event, then the tokens will be consumed as they arrive
- All tokens that were generated within the process must eventually be consumed by an end event before the process is completed
- If the process is a subprocess, it can be stopped prior to normal completion through interrupting intermediate events (see later)
  - In this situation the tokens will be consumed by an intermediate event attached to the boundary of the subprocess
BPMN Token Semantics: Example

- Message arrived: Start event generates a token
- Activity A gets activated
BPMN Token Semantics: Example

- Activity A completes
- Token is generated
- Parallel gateway gets activated
• Parallel gateway produces tokens for each leaving sequence flow
• Activities B and C may be performed
BPMN Token Semantics: Example

- Activity B completed and produced a token
- Exclusive gateway gets activated
  - It needs a single token only for activation
- Activity C has not yet been activated
  - E.g. because user is still busy
• Exclusive gateway produces a token
• Activity C has still not yet been activated
- Activity D completed and produced a token
- Activity C completed and produced a token
- End event does not consume token (that would terminate the process) because “upstream” token may reach it later on
• Exclusive gateway produced a token
• End event does still not consume token
• Activity D consumed the token, completed and produced a token
• End event consumed both tokens because no upstream tokens may reach it
• The process is ended
2. Tasks
A Task is an atomic activity that is included within a process

- This is unusual terminology – most often, tasks are activities to be performed by human beings
- The term *activity* is usually used in other standards and products

A Task is used when the work in the Process is not broken down to a finer level of Process Model detail

Generally, an end-user and/or an application is used to perform the Task when it is executed
Data Objects

- ...provide information about what activities require to be performed and/or what they produce
- ...are related via Associations ("dashed lines") to tasks, sequence flows

- A process can be modeled without Data Objects to reduce clutter
Tasks and Data Objects

- Data Objects may be shown as input/output of a process or a task
- Directionality added to the Association will show whether the Data Object is an input or an output
- State attribute of the Data Object can change to show the impact of the Process/Task on the Data Object
Service Task

- **Service Task**: A Task that uses some sort of service to provide an action
  - ...Web service or an automated application
User Task

- User Task: A typical “workflow” task where a human being performs the Task possibly with the assistance of a software application
  - Often, it is scheduled through a task list manager of some sort
Script Task

- **Script Task: ...is executed by a business process engine**
  - The modeler or implementer defines a script in a language that the engine understands how to execute
  - When the Task is ready to start, the engine will execute the script
  - When the script is completed, the Task will also be completed
Manual Task

- *Manual Task*: A Task that is expected to be performed without the help of any business process execution engine application
  - An example of this could be a telephone technician installing a telephone at a customer location
Send Task

- **Send Task**: A simple Task that is designed to send a message to an external participant
  - Once the message has been sent, the Task is completed

- **Note**: A send task (left) is same as the throwing message intermediate event (right)
Receive Task

- **Receive Task**: A simple Task that is designed to wait for a message to arrive
  - Once the message has been received, the Task is completed
  - A Receive Task is often used to start a Process

- **Note**: A receive task (left) is same as the catching message intermediate event (in the center, see later). Use the event gateway (right) to model a timeout for the wait
2. BPMN

2.2 Ausdrucksmittel