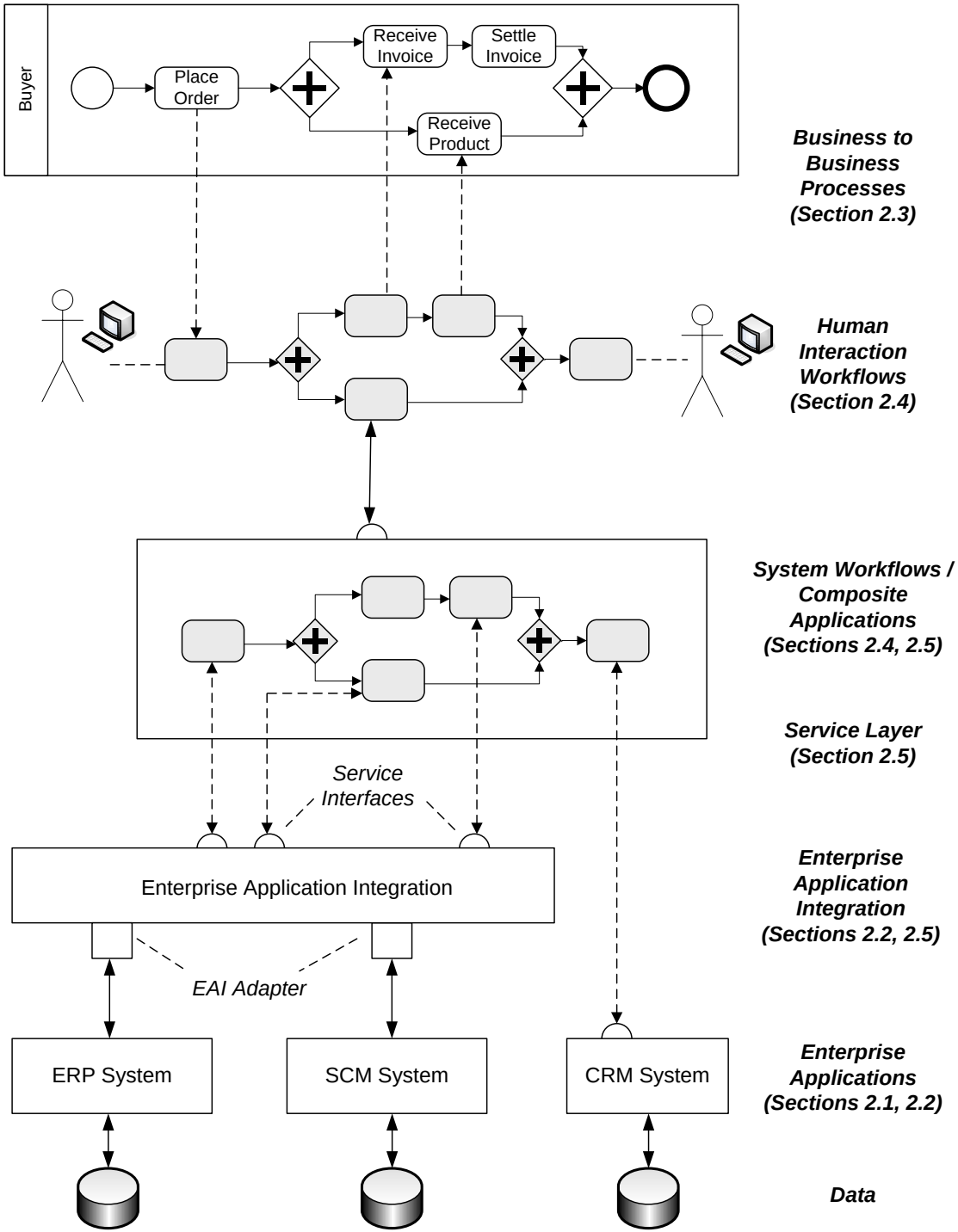
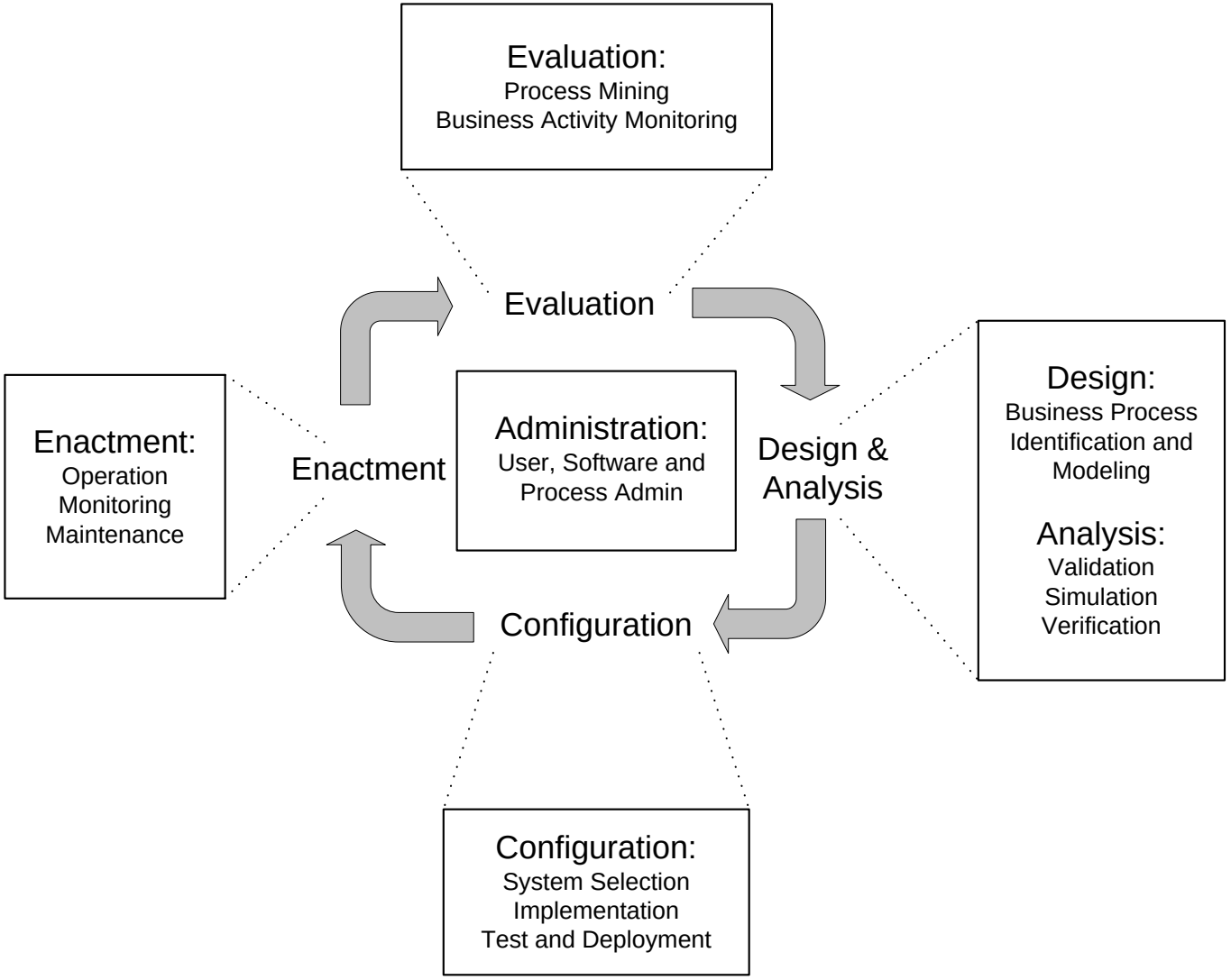
Overall Picture



from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

Business Process Lifecycle

- Idea
  - Business processes have different phases in which specific problems are treated using different techniques
- Phase model
  - In each phase artifacts are created that are used as input for the subsequent phase
  - By iteration of the life cycle, continuous process improvement is achieved



from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

Design & Analysis

- Design
  - Identification of business processes
  - Explicit representation through process models
  - Use of process modeling languages
- Analysis
  - Validation of the process models through discussions and reviews
  - Simulation of models can discover weaknesses
  - Verification of formal properties of business processes
- Remark
  - Design and analysis independent from concrete implementation
- Artifact: Operational Business Processes

Configuration

- Implementation of the process in the organization
  - Target implementation, KPIs
  - Restructuring
  - Distribution of responsibilities
  - Trainings, Change Management
- Implementation of the process by software systems
  - Completing technical aspects
  - Adaptation of existing applications and selection of new applications
  - Integration of information systems, introduction of a POIS
  - Migration
  - Shutdown of legacy systems
- Artifact: Technical business process (Workflow)

Enactment

- Process coordination
  - Business process instances are executed under the control of the process manager / POIS
  - Execution data are collected (execution log) (Process mining)
  - Indicators are measured
- Monitoring
  - Monitor the process instances (Compliance with regulations, SLAs,…)
  - Accurate information is therefore available
- Maintenance
  - Maintenance of POIS and applications
- Artifact: version information of the business process instances

Evaluation

- Usage of execution data in order to discover vulnerabilities
  - This information is used in the design phase to improve the process models and in the configuration phase to improve the implementation
- Business Activity Monitoring
  - Monitoring of business processes from an application perspective
- Process Mining
  - Analysis of execution logs to, e.g., discover weaknesses

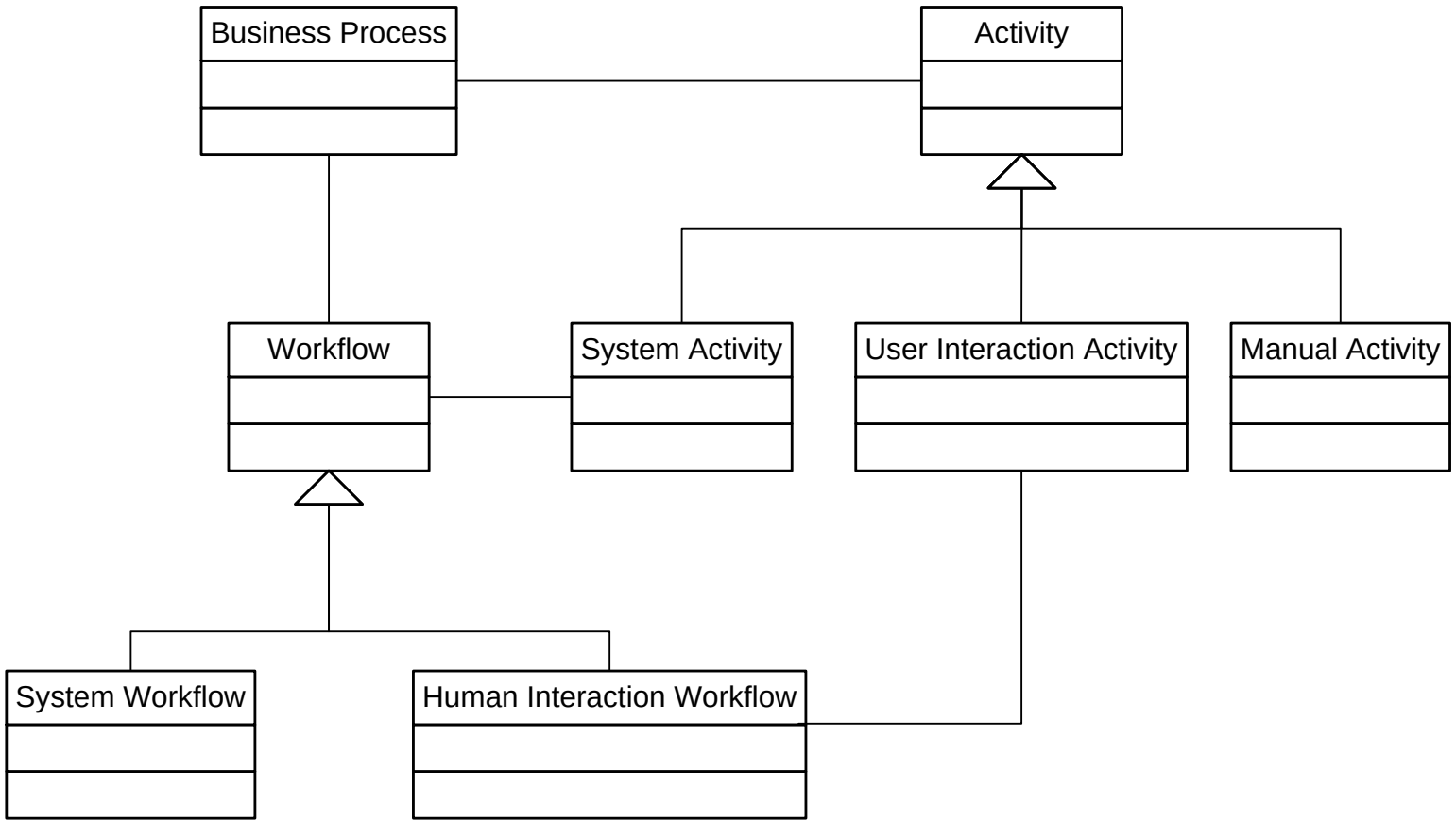
2 Business Process Modeling

- Conceptual Modeling
- Value Chains and Business Processes
- Process Metamodel
- Business Process Model and Notation (BPMN)

Conceptual Modeling

- Structuring the problem space
  - Operational business processes
  - Workflow (Implemented business processes)
  - System-workflow
  - Interaction-workflow
  - Activities
  - System activities
  - Interaction activities
  - Manual Activities
- Approach
  - UML Class diagrams

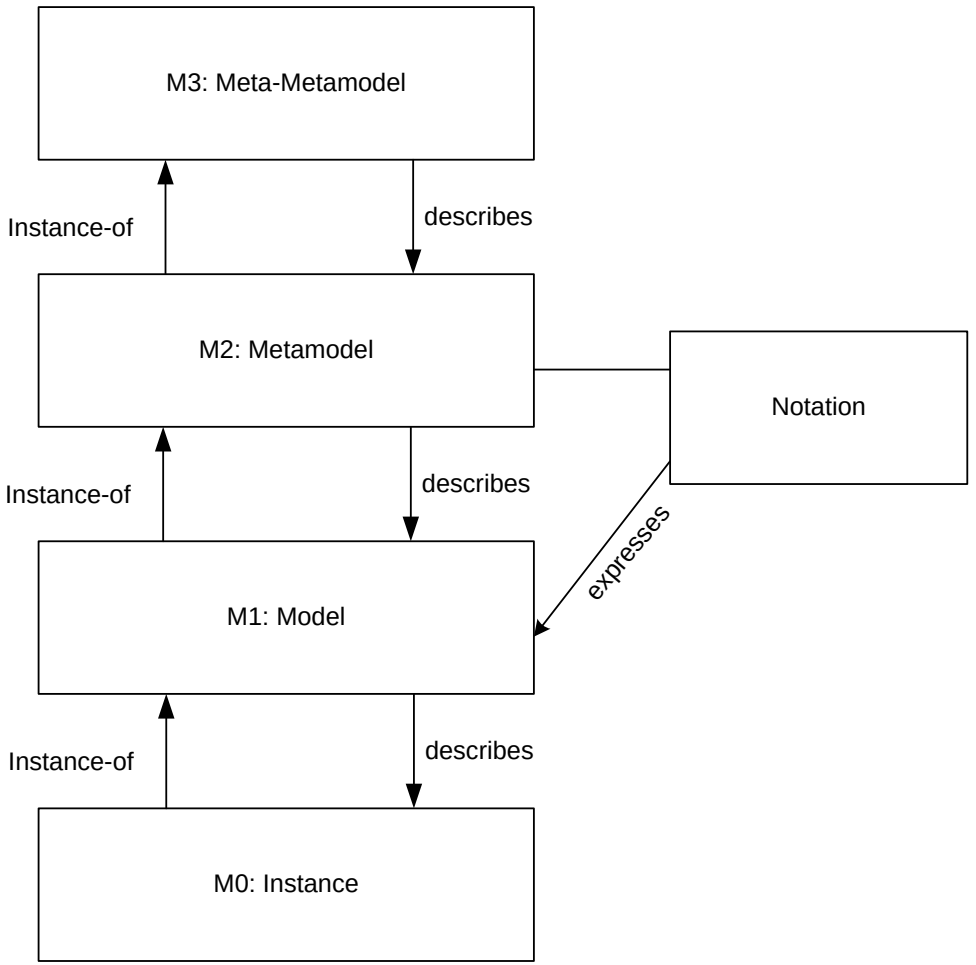
A Conceptual Model



from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

Abstractions

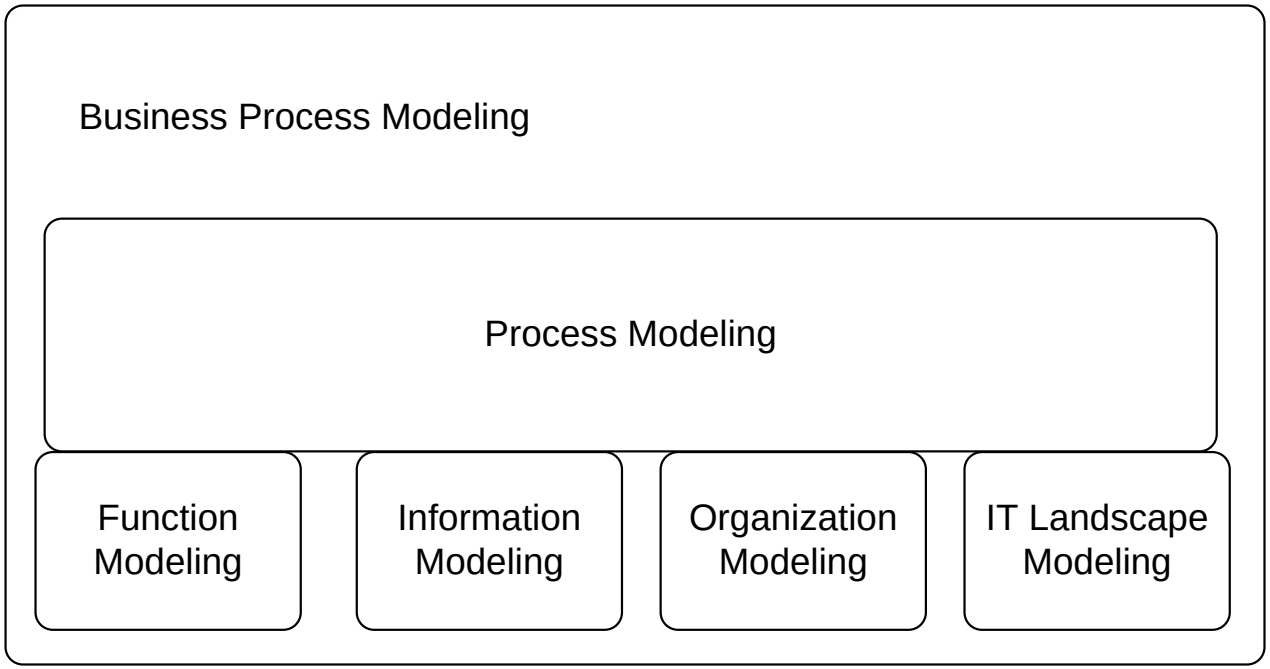
- Horizontal abstraction
  - Instance: concrete case
  - Model: describes several instances
  - Metamodel: describes modeling elements and how they can be used to gether to create a model
  - Meta-Metamodel: Provides the concepts to describe the meta-model



from M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2007

Abstraction Concepts

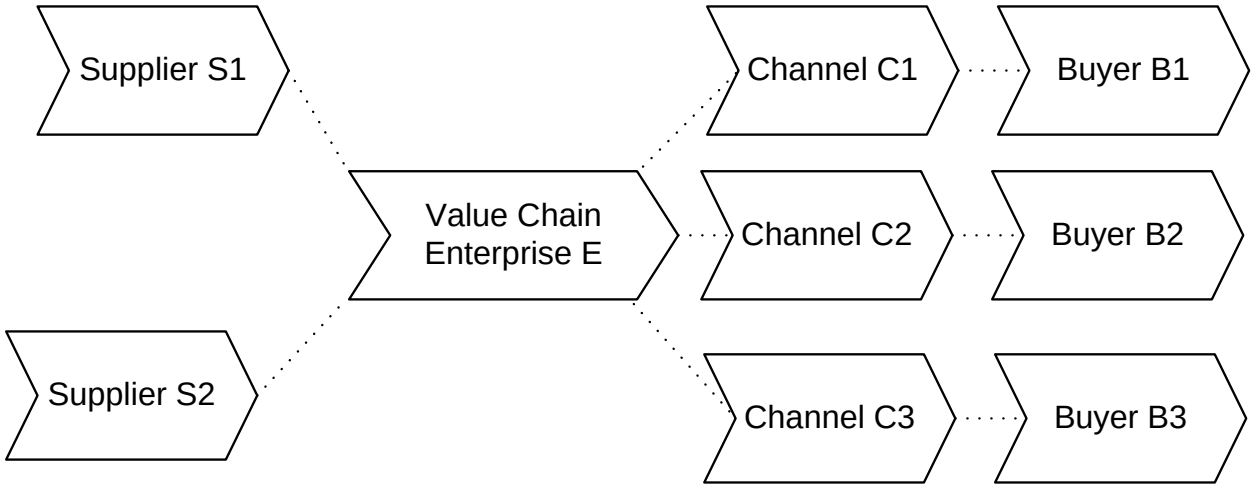
- Vertical abstraction
  - Separate modeling of different parts of business processes and their execution environment
  - Sections are examined in more detail in the following



from M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2007

Modeling of functions

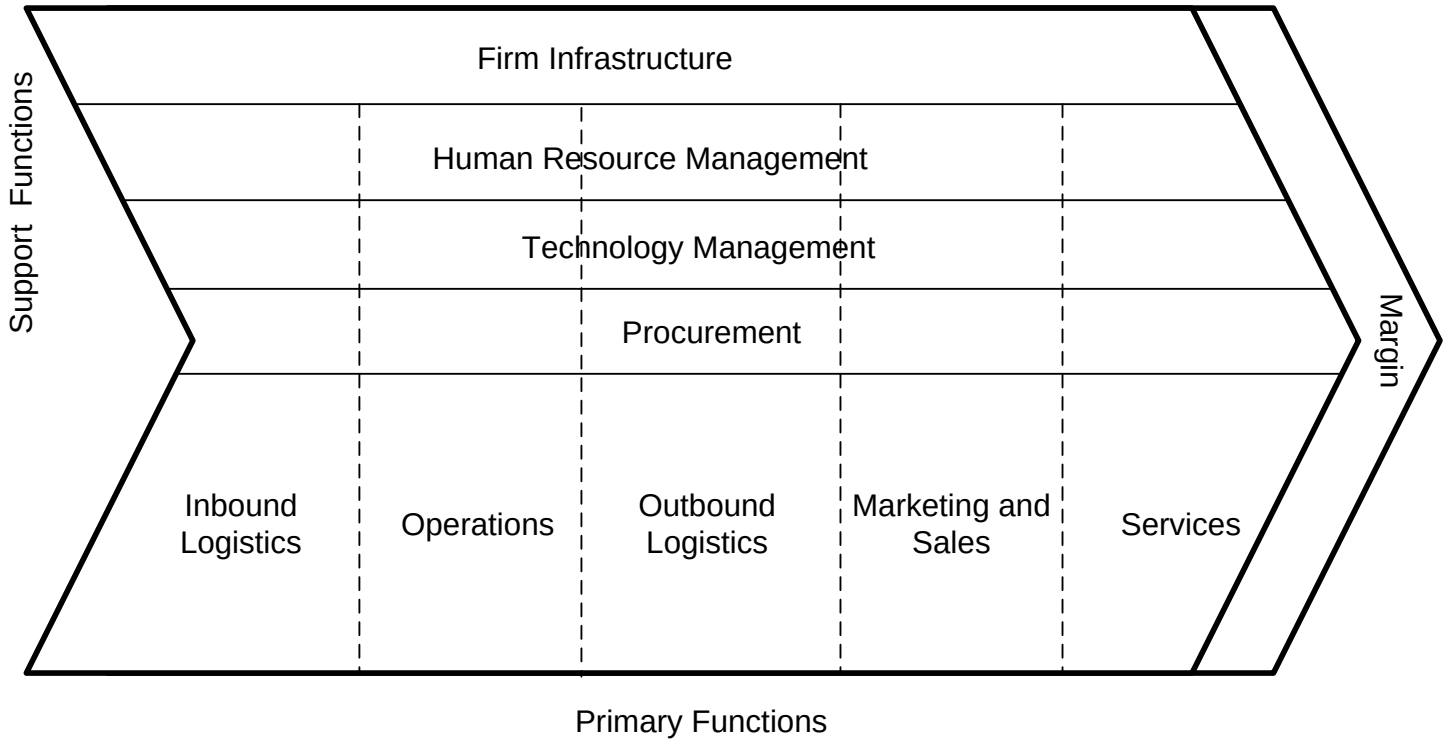
- Idea
  - Overall picture from coarse grain business functions to operational activities and business processes
- Starting point: value chains
  - Each organization is represented by exactly one value chain which forms, together with those of business partners, *value system*



from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

Modeling of functions

- Remark
  - Each chain contains the coarsely granular divisions of the organization, based on the functional structure

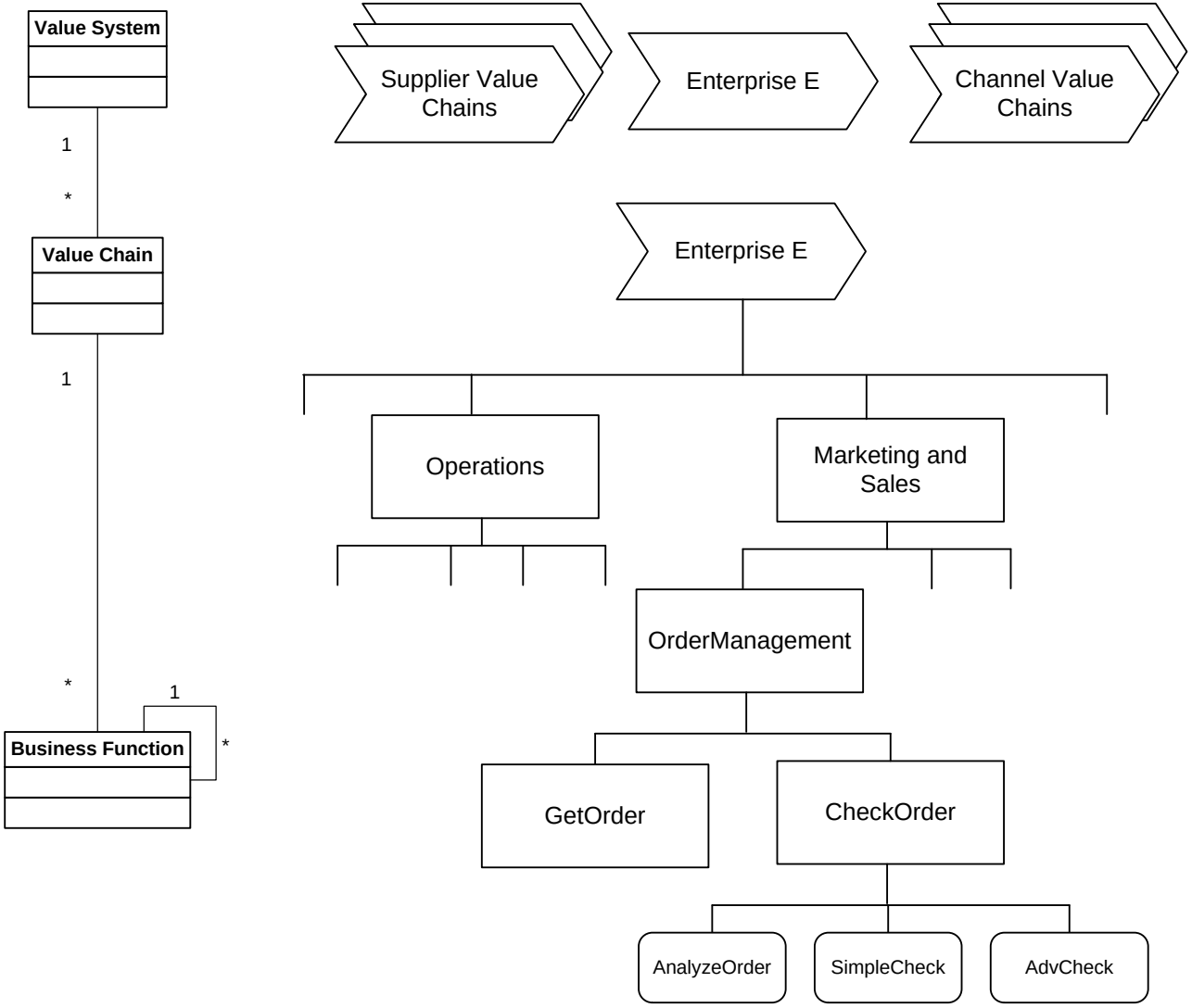


from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

From Value Chains to Business Processes

- Approach
  - Value chains provide the framework within which organizational functions are executed
  - They can be functionally decomposed; a coarse granular function is refined by a set of functions on the lower level
- Remark
  - Coarse granular functions represent a similar extent as organizational business processes
  - The process concept is not explored in the context of value chains, the focus is on functional decomposition and analysis of value-added functions

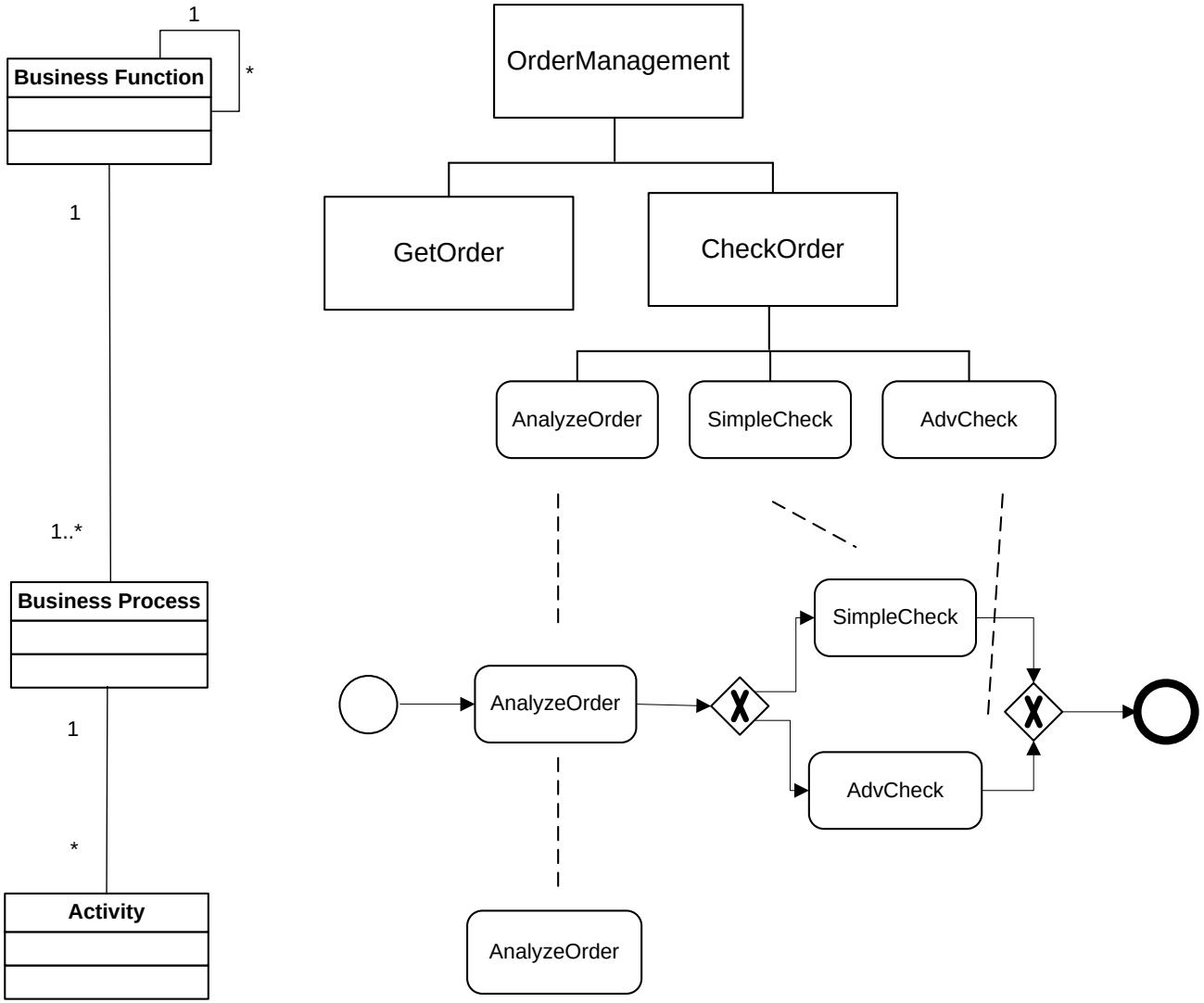
Functional Decomposition



Operational Business Processes

- Transition from business functions to process activities
  - Fine-grained functions are the activities of operational business processes
  - This creates a connection between the organizational modeling and operational business processes

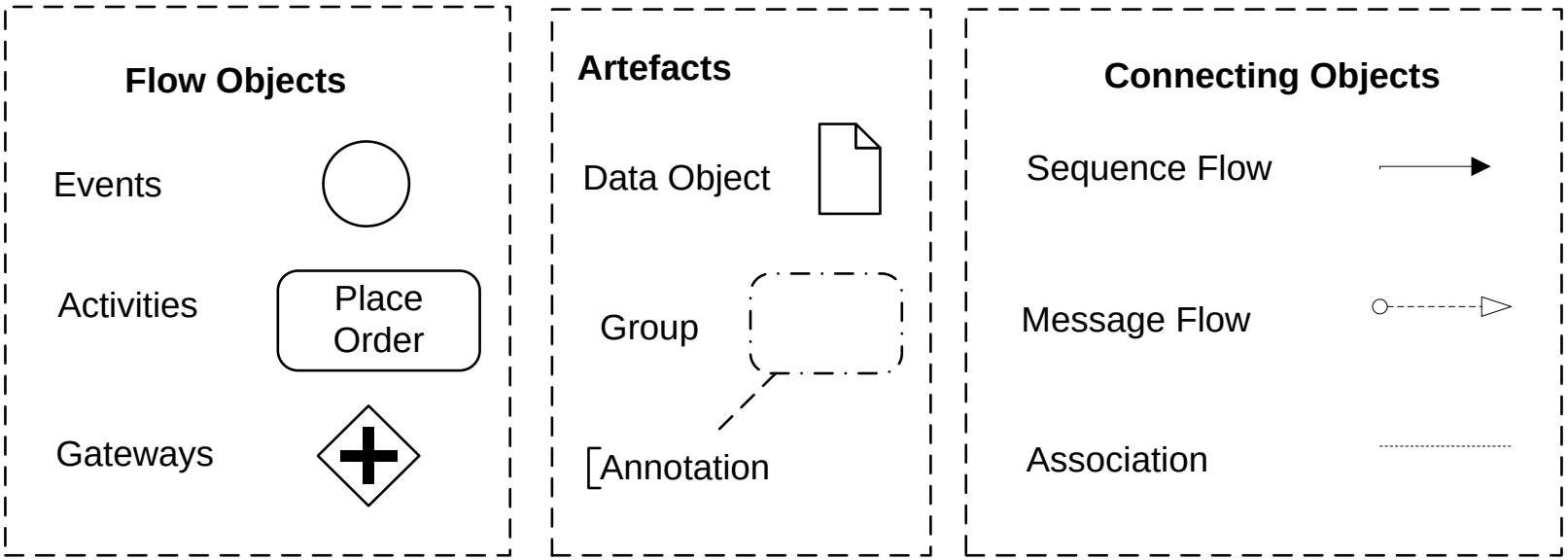
Activities of Business Processes



Business Process Model and Notation

- BPMN, Version 2.0
  - Industry standard of Object Management Group (OMG)
  - Formerly (Version 1.X: Business Process *Modeling* Notation)
- Expressive modeling language targeting mainly functional modeling of business process models
  - MOF-compliant Metamodel
  - Precise execution semantics for almost all constructs
- BPMN is not intended for modeling:
  - Process landscapes
  - Organization structure
  - Data structures
  - IT-landscapes

Categories of Elements



from M. Weske: Business Process Management, © Springer-Verlag Berlin Heidelberg 2007

Basics

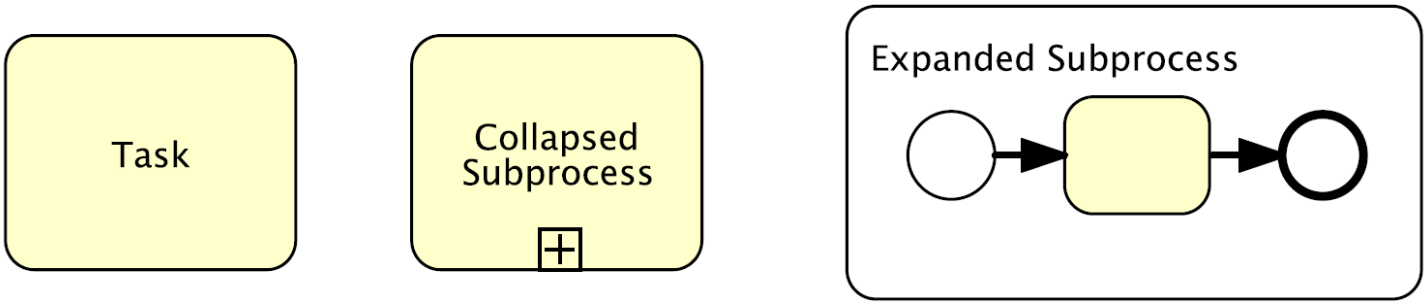
- Notation
  - Concepts of Process modeling, e.g., Activity, Event
  - Attributes of Concepts
  - Graphical Representation
- Syntax
  - Correctness of models
  - Rules for proper composition (Connect, containment) of the concepts
- Semantics
  - Meaning of the modeling concepts
  - Execution rules, transformation rules, ...

Attributes in BPMN

- Idea
  - Attributes enrich the graphical representation of business process diagrams
  - Only a subset of the attributes can be represented graphically
  - Result: The graphical representation is not complete!
- Attributes of Business Process Diagrams
  - Technical information (id, name, version, author, language)
  - Expression language, e.g., to describe edge conditions
  - Attributes can be of simple values or instances of classes
- Hint
  - BPMN can therefore be viewed as a framework that allows different concrete languages for expressions

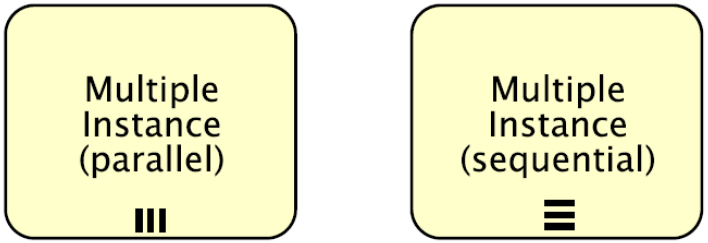
Activities

- Activities represent work to be done
  - Activities take time
  - Activities are either atomic (Task) or composite (Subprocesses)
  - Subprocesses can also be collapsed, if their internal structure is irrelevant for the given model.



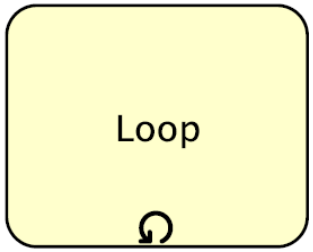
Activities

- Multiple Instances
  - Compact presentation of activities that must be performed several times
  - Example: Activity that must be performed for all order items of an order
  - Note: This is the for loop in programming languages, at least for sequential execution
  - Attribute: `LoopCharacteristics` is of type `MultiInstanceLoopCharacteristics`, with `isSequential =` {true, false}



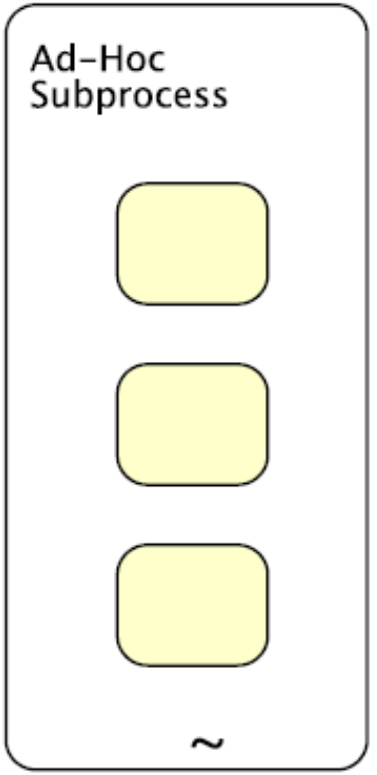
Activities

- Loop-Activity
  - Repeated execution of an activity can also be represented by loop activity
  - Here, a condition determines the repition execution of the activity
  - Attribute: `LoopCharacteristics` is of type `StandardLoopCharacteristics`
  - Note: like the *While* loop or the *Repeat-Until* loop in programming languages (depending on the `testedBefore` = {true, false})



Activities

- Ad-hoc Subprocess
  - Allows the modeling of unstructured parts of a process
  - Designer models the tasks that can be executed at this point, without defining an explicit sequence
  - contained tasks can be often repeated
  - End Condition decide about termination of Activity



Sequence Flow

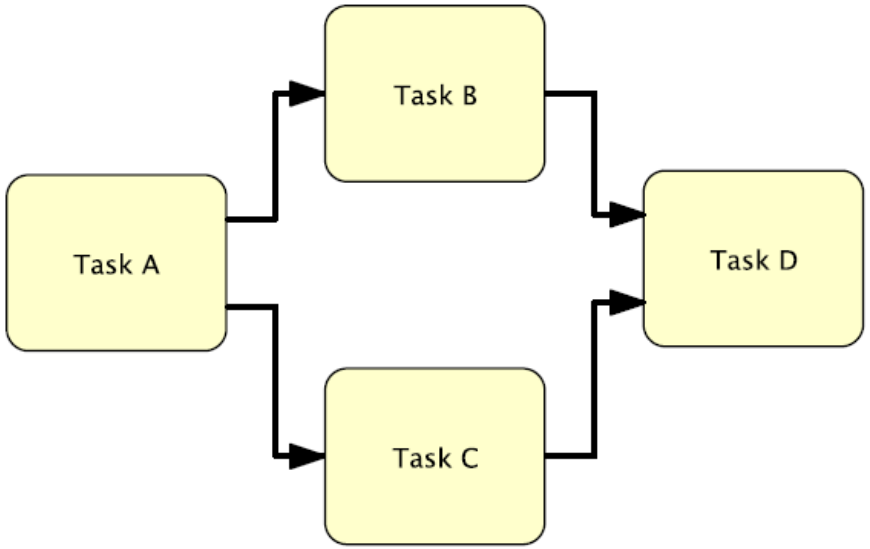
- Execution order of the activities of a process is defined by sequence flow edges
- Execution semantics: What does it exactly mean? A → B
  - The activity B can only be started if the activity A is finished
- Is achieved by *signaling* of edges
  - When A completes, the sequence flow edge signals
  - When B receives this signal, B can be started

Execution conditions

- With a sequence flow, we can only define sequences of activities, which is often not sufficient
- More complex execution conditions can be expressed by gateways
- Gateways have a basic shape (diamond), which can be complemented by different symbols to represent the different behavior graphically
- Most commonly used
  - Data-based exclusive Gateway (XOR Gateway)
  - Parallel Gateway (AND Gateway)

Sequence Flow – Uncontrolled Flow

- Tasks can have more than one incoming/outgoing edges
- “Uncontrolled” Flow semantic
  - Every outgoing edge is signaled
  - Every incoming *signaled* edge triggers execution of the task
- Are often cause for confusion
  Best Practice: Do not use it!

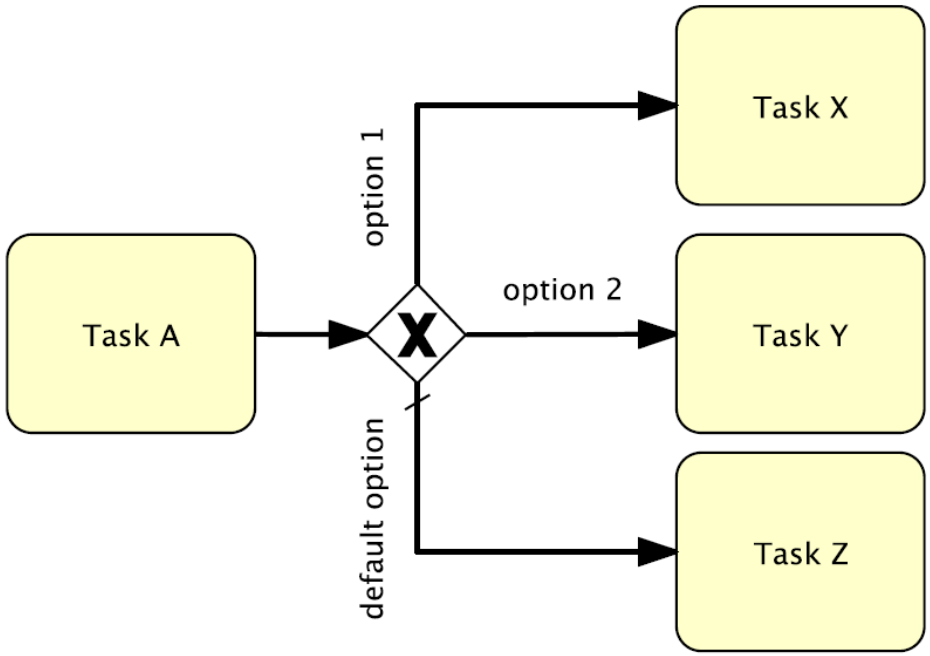


Gateways

- Data-based exclusive Gateway, as Split-node
  - From a set of alternatives, just one alternative is selected on the basis of existing data
  - Each sequence flow that leaves the gateway, has set attribute `ConditionType` set to "`Expression`", and a `ConditionExpression`
  - The path of the first sequence flow that evaluate to true is signaled, all other alternatives are discarded
- Data-based exclusive Gateway, as Join-node
  - Alternative paths are merged
  - Once an incoming edge is signaled, the outgoing edge is signaled

Gateways

- Refined sequence flow
  - A sequence flow edge can be set as a default flow, which is signaled when all other alternatives are not met. That is, their conditions are evaluated to false



Gateways

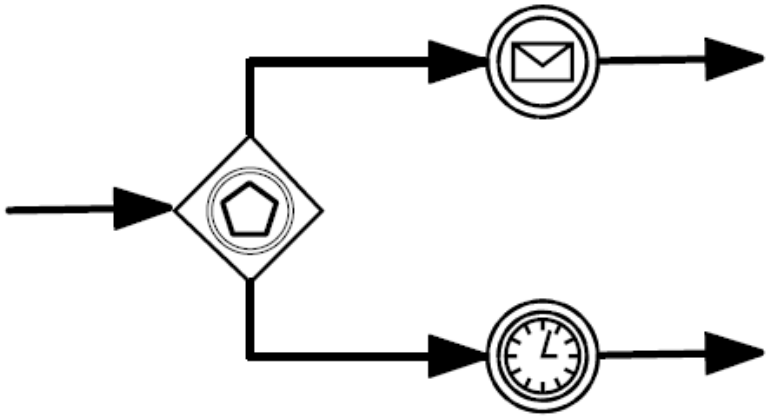
- Parallel Gateway, as Split-nodes
  - All outgoing paths of the parallel gateway are signaled
  - Thus, concurrent activities are modeled
- Parallel Gateway, as Join-nodes
  - Parallel join gateway fires only when all incoming edges were signaled (synchronization)
  - The concurrent processing threads are merged again

Gateways

- Inclusive OR Gateway, as split-nodes
  - At branching time, an arbitrary (nonempty) subset of outgoing edges are signaled, at least one and at most all
  - The inclusive gateway can thus be regarded as a generalization of the above two gateways
- Inclusive OR Gateway, as join-nodes
  - The join node waits until all the incoming edges, which are signaled for the current process before it signals its outgoing edge
  - Popular due to its flexibility, but complex semantics

Gateways

- Event-based Gateway, only as Split-nodes
  - Is used to catch environment based conditions, e.g., receving an event or a timeout
  - Only the outgoing edge of the first occurring event is signaled
  - Join semantics is carried out using the XOR-join



Events and Their Types

- Characterization
  - Events do not require time
  - can be, received or thrown
  - have type: Start, intermediate, end event
- Start event (received)
  - Causes the start of a new process instance
  - Notation: Circle with a thin border
- End event (thrown)
  - Occurs when a process instance is terminated
  - Notation: Circle with thick border
- Intermediate event (Received/thrown)
  - Occurs between start and end events
  - Notation: Circle with double border

Event Types(Causes)

- Idea
  - In addition to the types of events (start, intermediate, end), there are further types of events (causes)
- Approach
  - Event types describe the specific cause of events, such as the arrival of a message
  - In this way, process models are enriched with specialized information, so that people can better understand processes
  - Each event has an event type and an event cause, but not all combinations are allowed!

Meaning of Start Events

Plain

- Event untyped, e.g. manual start of a process

Conditional

- If the condition is true, the process starts

Message

- Reception of a message

Signal

- Occurrence of a business event

Multiple

- Several options to start a process

Meaning of End Events

Plain

- Stops execution of a path, not necessarily the whole process instance

Termination

- Occurrence of the event ends the process instance immediately

Message

- Finally, send a message

Signal

- Triggering a signal that can be received by another process instance to react upon

Intermediate Events

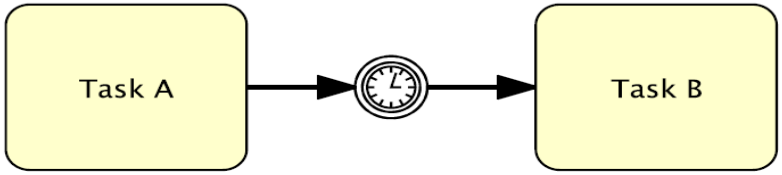
- Intermediate events occur during the process execution
- They can be either received or thrown
  - Received: the process waits until the intermediate event occurs
  - Thrown: The process throws the event and proceeds
- They are always associated with the process
  - By sequence flow (received or thrown)
  - Associated with activities (only received)

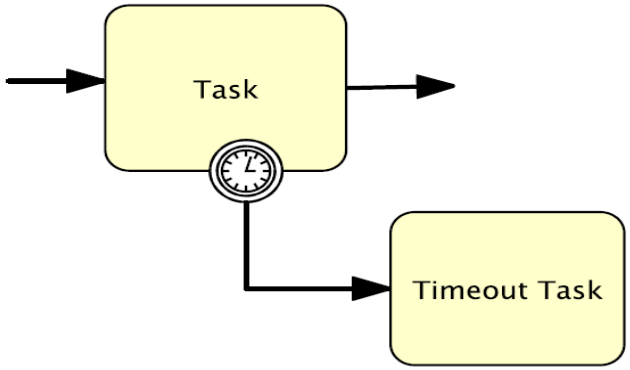
Message Events

- Represent receiving (received) or (throwing) of messages
- Sending a message occurs directly
- When receiving, the process instance is suspended until message is received

Timer Event

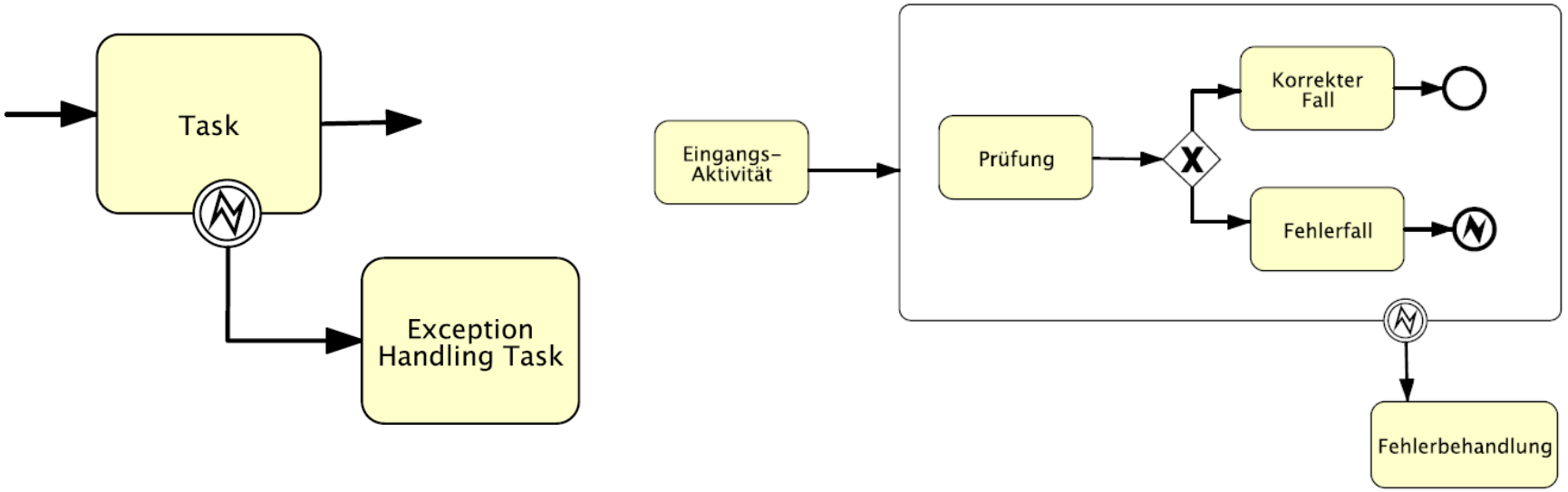
- Time-Out: process instance is waiting for the occurrence of the time out
- Attached intermediate event: when the timer expires before the activity was completely executed, the outgoing edge of the event is signaled
- Time for the continuation of process instances is defined as:
  - Time span (10 min, 15 days, ...)
  - Time point, absolute / relative (8:00 clock, two days before travel, ...)





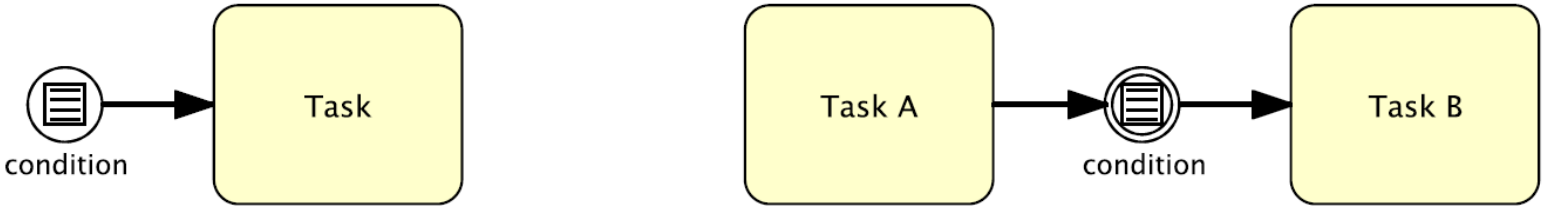
Error/Exception Event

- Response to error conditions; can only occur and be attached to activity as an intermediate event
- In the case of failure, the proposed error handling logic is executed



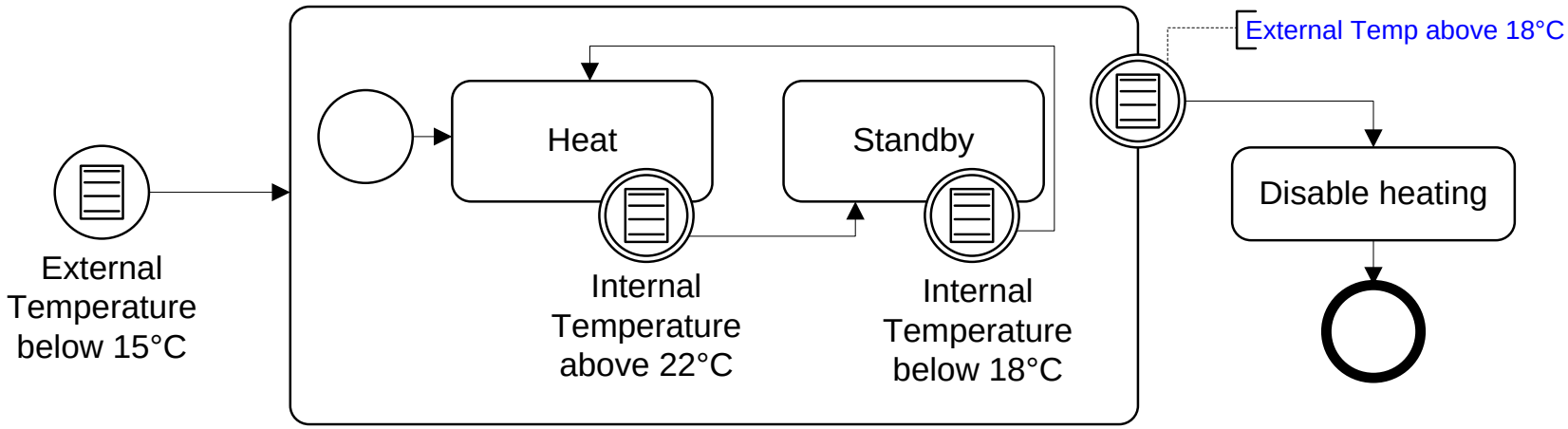
Conditional Events

- Idea
  - To model an automatic response to changing conditions in the process, conditional events can be used (eg: temperature > 25 degrees)
- Start-Conditional-Event
  - When condition is satisfied, a process instnace is started
- Intermediate-Conditional-Event
  - When condition is satisfied, the process instnace is resumed



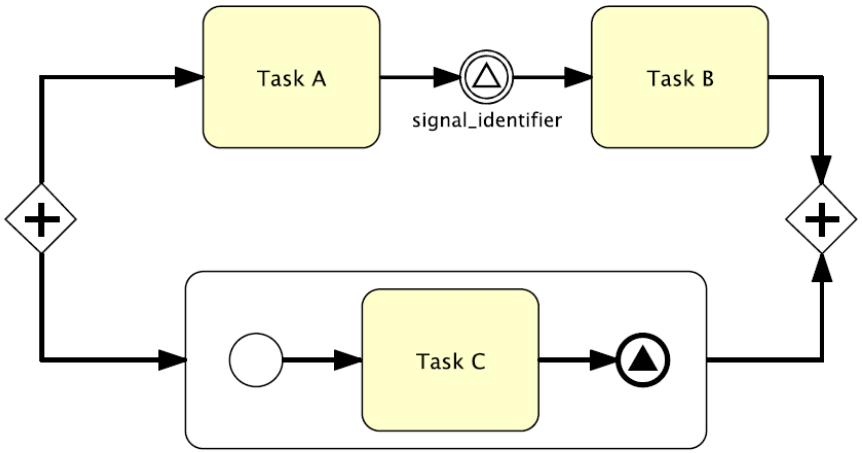
Example: Conditional Event

- Events are based on conditions
  - Event occurs when a rule is evaluated to true
    - Start event: Process instance is started
  - Intermediate rule event to an activity boundary (*exception flow)*



Signal Events

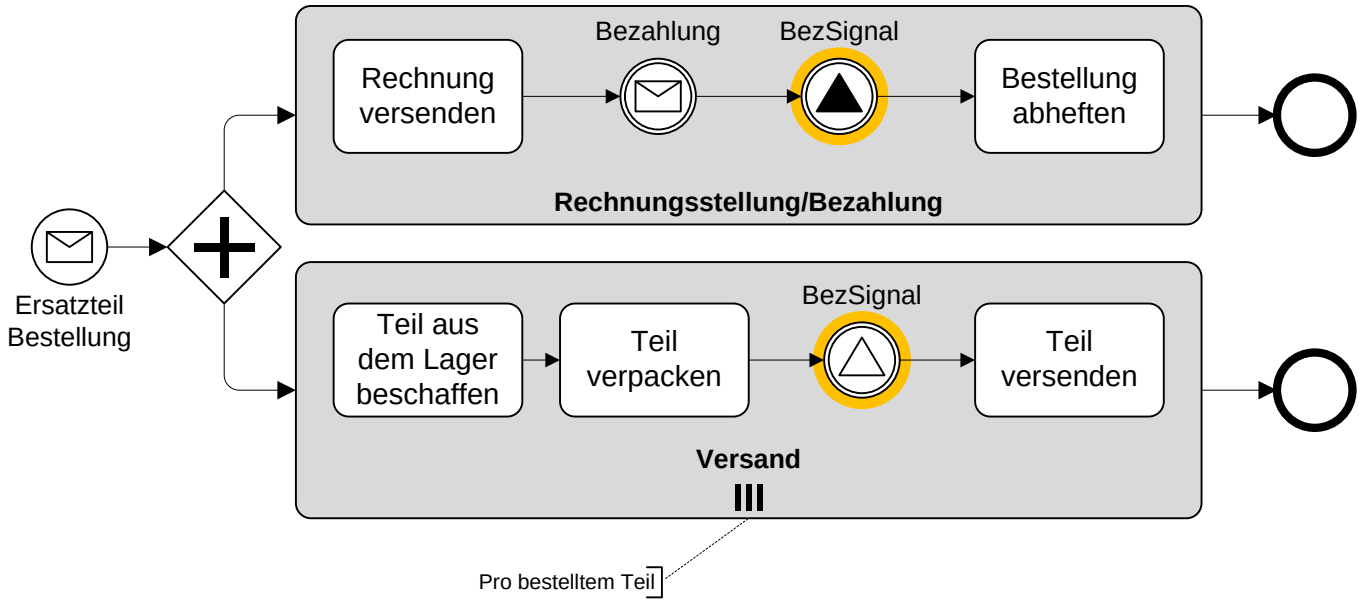
- Send a public signal within a process and/or across process boundaries
- Signal Broadcasting
  - A thrown signal can be processed in several places
- Representation of complex control flow scenarios
  - e.g., synchronization across boundaries of a subprocess
  - e.g., synchronization across boundaries of a process



Signal Events

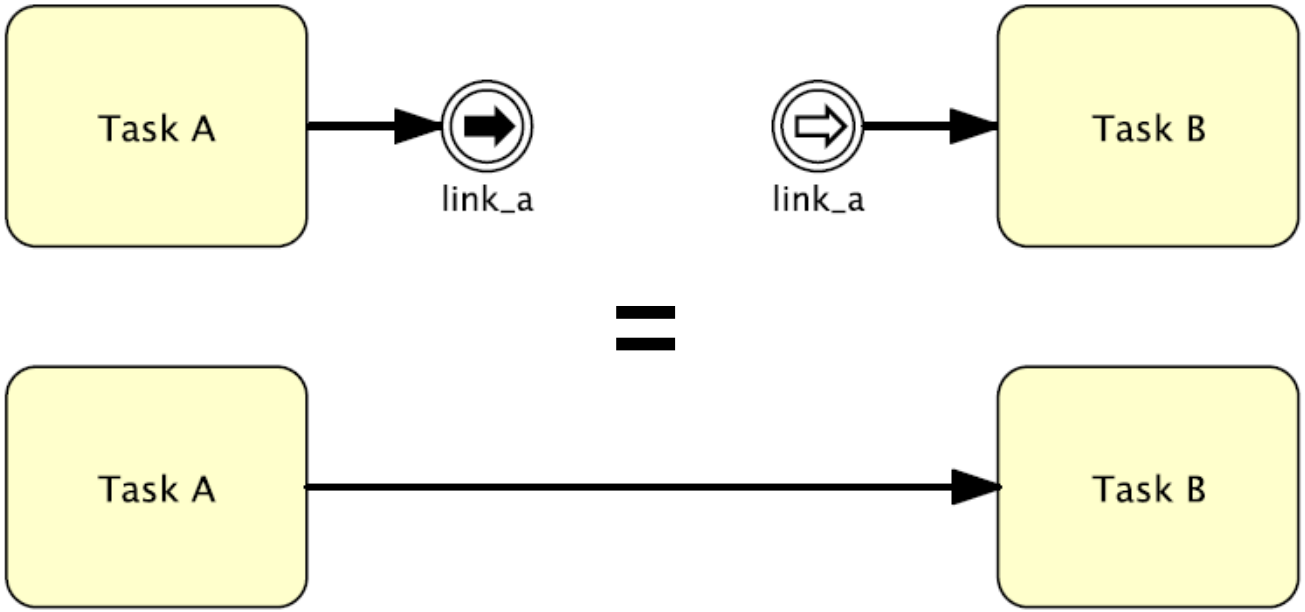
"A BPMN Signal is similar to a signal flare that shot into the sky for anyone who might be interested to notice and then react."

(BPMN v2.0 Beta 1 Specification)



Link Events

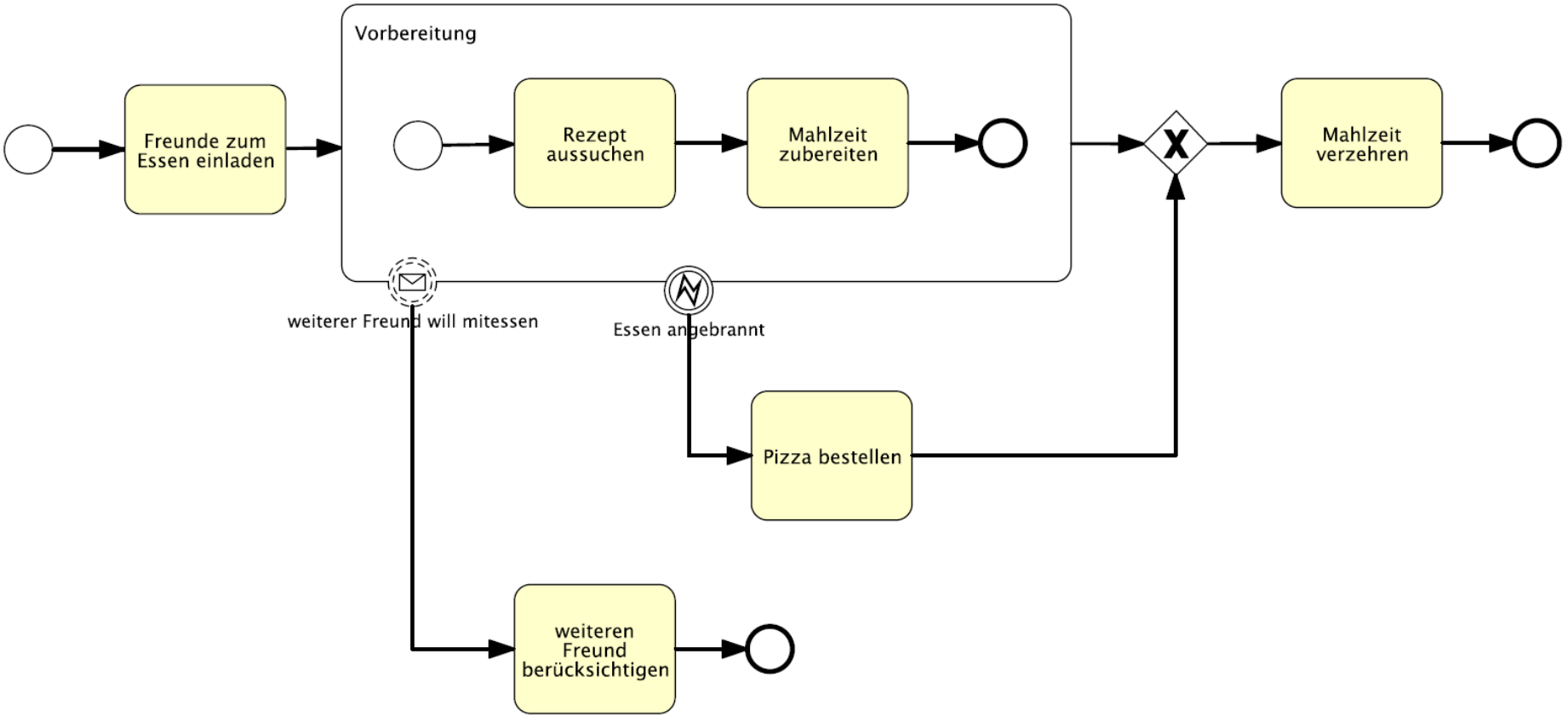
- Idea
  - If a process model does not fit on the available drawing area, a connection to another process diagram by a link event will be made
  - Matched link events behave like a sequence flow



Events' Behavior

- Received, attached intermediate events intercept certain events during the execution of the activity
- Different behavior (not available for all types of events)
  - *interrupting*
    - Activity is cancelled
    - Notation: solid, thin border (such as start events)
  - *non-interrupting*
    - Activity continues to run, outgoing path of the event is run in parallel
    - Notation: dashed, thin border

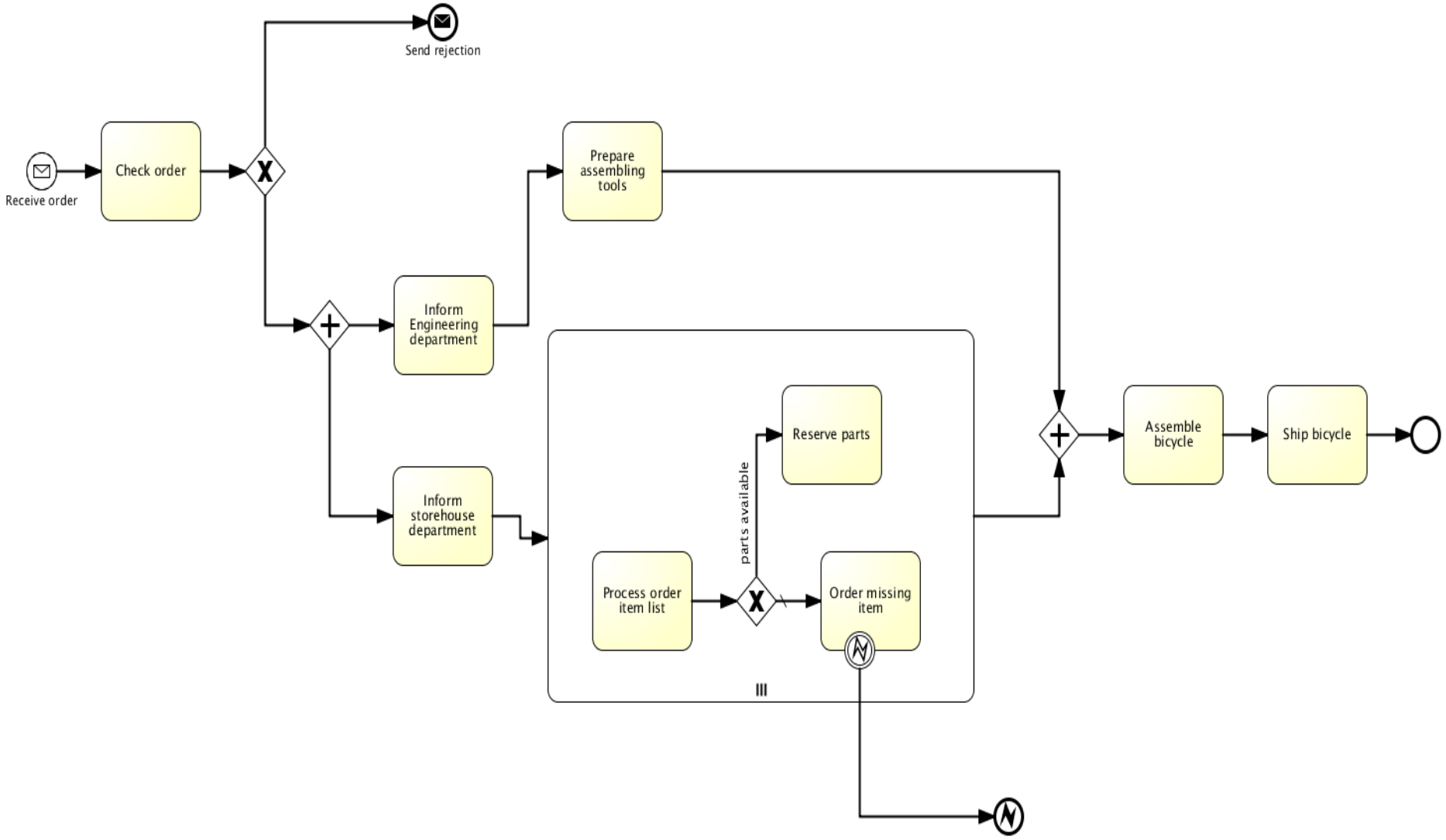
Example: Intermediate Events



Overview of Events in BPMN

Let’s model this

A small company manufactures customized bicycles. Whenever the sales department receives an order, a new process instance is created. A member of the sales department can then reject or accept the order for a customized bike. In the former case, the process instance is finished. In the latter case, the storehouse and the engineering department are informed. The storehouse immediately processes the part list of the order and checks the required quantity of each part. If the part is available in-house, it is reserved. If it is not available, it is back-ordered. This procedure is repeated for each item on the part list. In the meantime, the engineering department prepares everything for the assembling of the ordered bicycle. If the storehouse has successfully reserved or back-ordered every item of the part list and the preparation activity has finished, the engineering department assembles the bicycle. Afterwards, the sales department ships the bicycle to the customer and finishes the process instance.

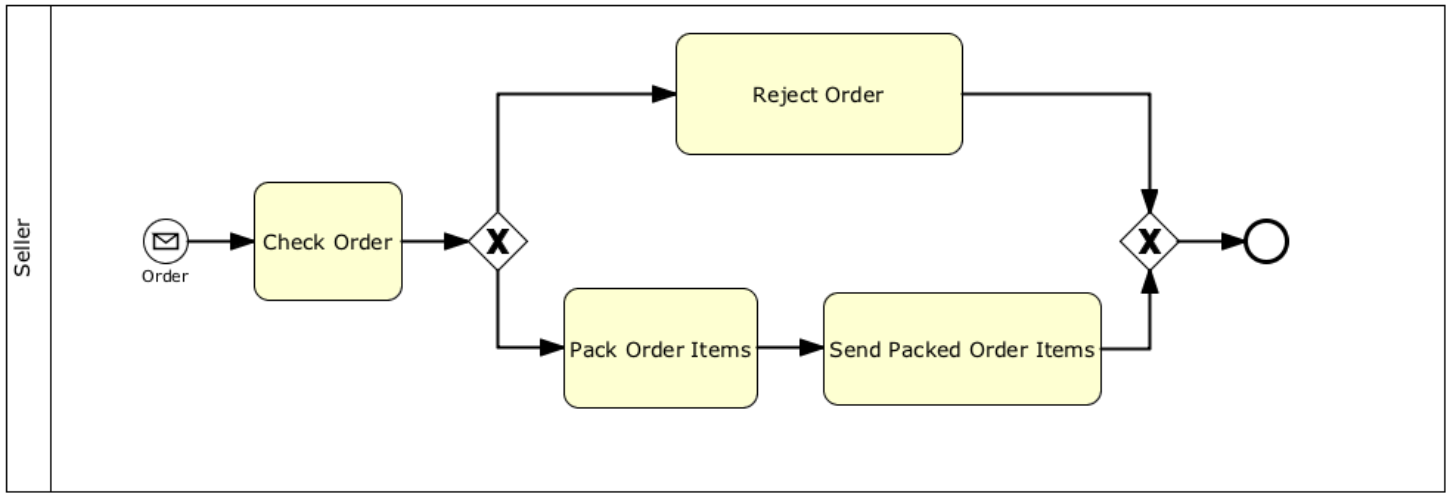

Send rejection
Receive order
Check order
Prepare assembling tools
Inform Engineering department
Inform storehouse department
Reserve parts
parts available
Process order item list
Order missing item
Assemble bicycle
Ship bicycle

4 Data and Processes

- Motivation for data in processes
- Modeling data
- Data-based process control
- Consistency between processes and data
- Analysis of data anomalies in business processes

Motivation for data in processes

- Traditionally
  - Focus is on activities and control flow
- But data are also recognized
  - as identifiers of items (activities, events)
  - Occasionally, as data objects (such as in BPMN, not in classical Petri nets)

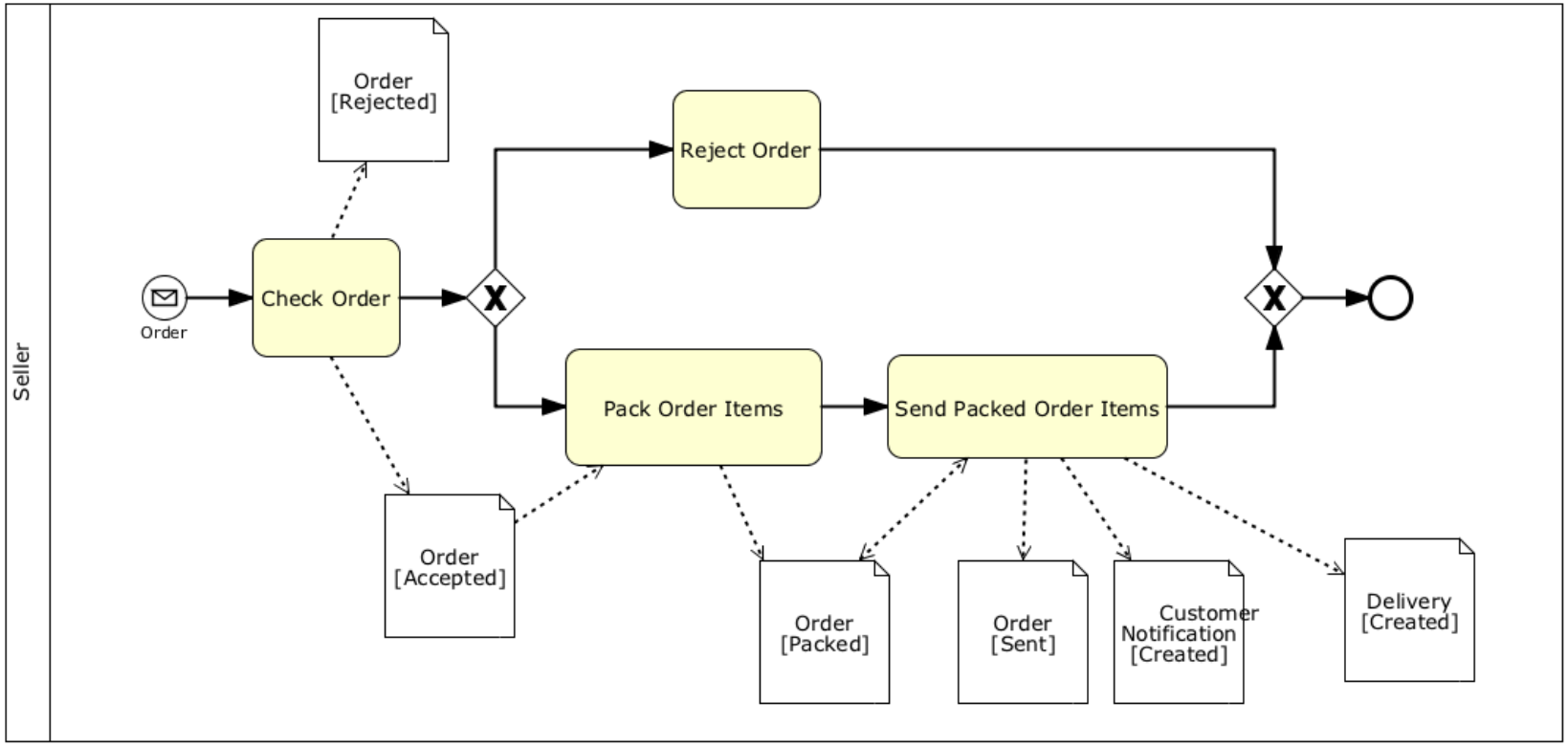


Motivation for data in processes

- Executing processes
  - IT system-supported process execution comes to the foreground
  - Executable processes require precise information needed to create data objects
- What is covered
  - Modeling data objects
  - Dependencies among different data objects
  - States of a data object within the process execution
  - Interaction between activities and data
  - Data on business/technical level

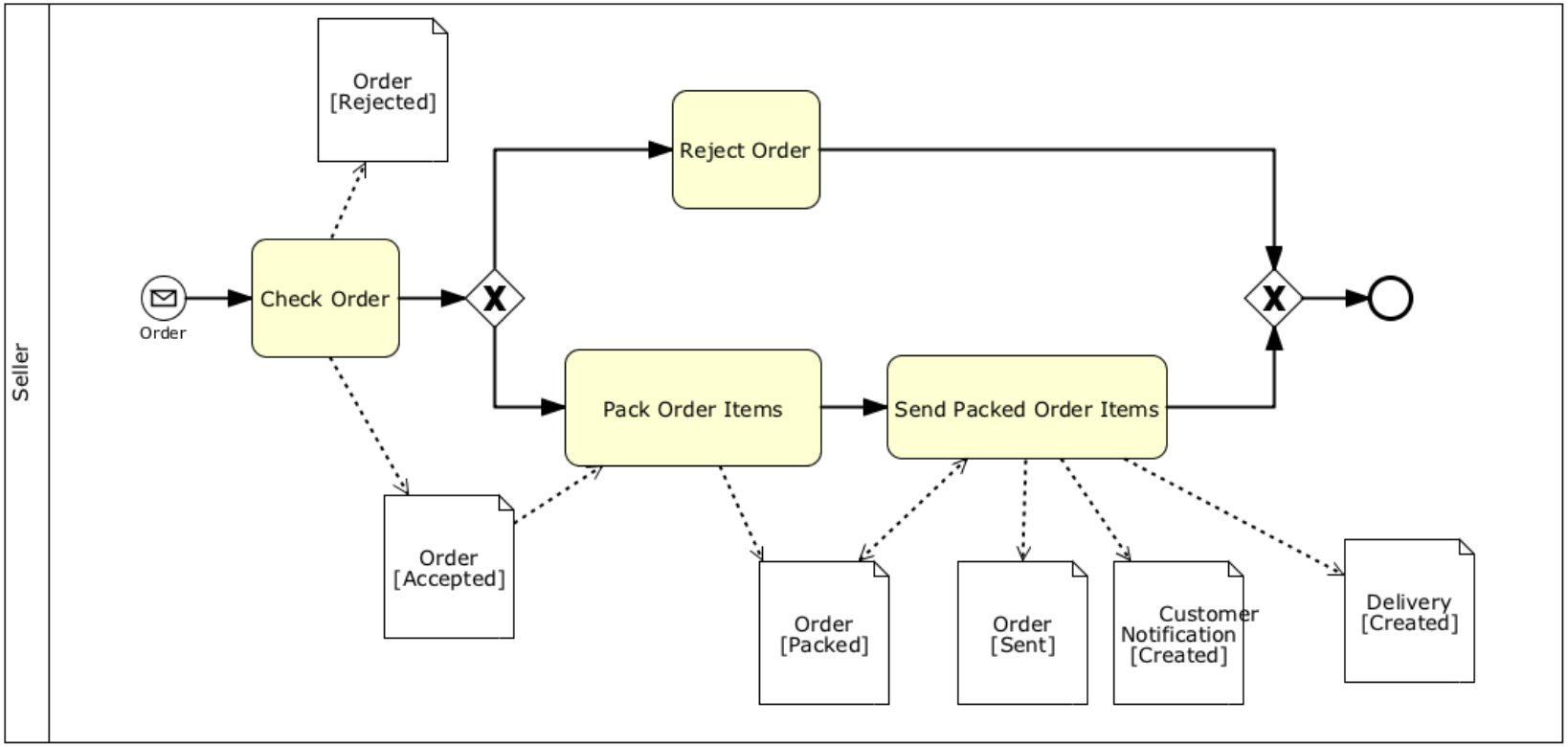
Example: Data objects

- The “order processing” process works on a variety of data objects
- Read / Write by edge directions



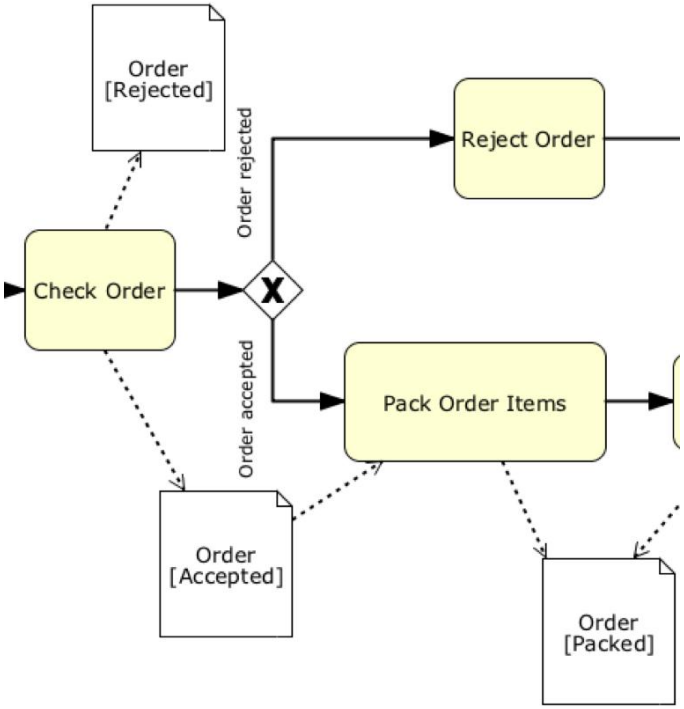
Example: Data Objects States

- Data objects can have states



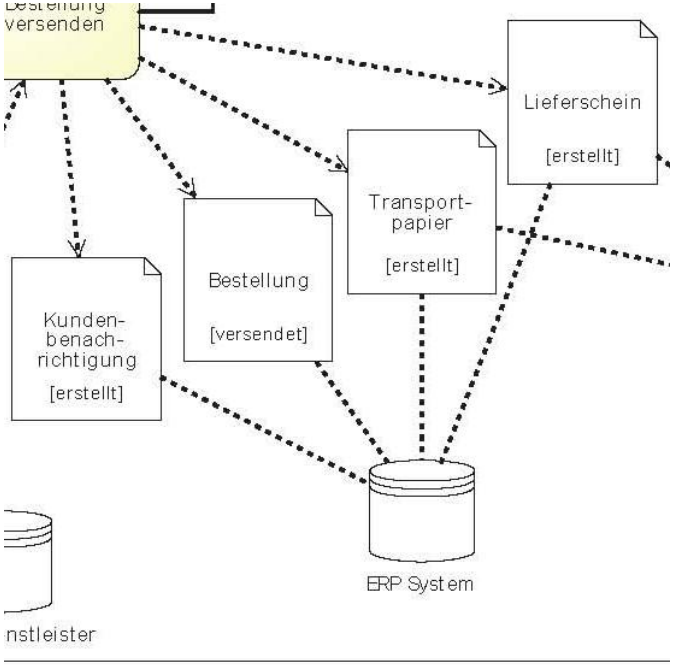
Data in Processes: Decisions

- Visualization of data objects facilitates understanding of data flow and data dependencies
- Decisions at gateways can be based on data, as data-based XOR-Split
- Examples
  - Credit risk is “high”
  - Total price < 3000 €
  - See Figure
    - Decision based on the current
    - state of a data object

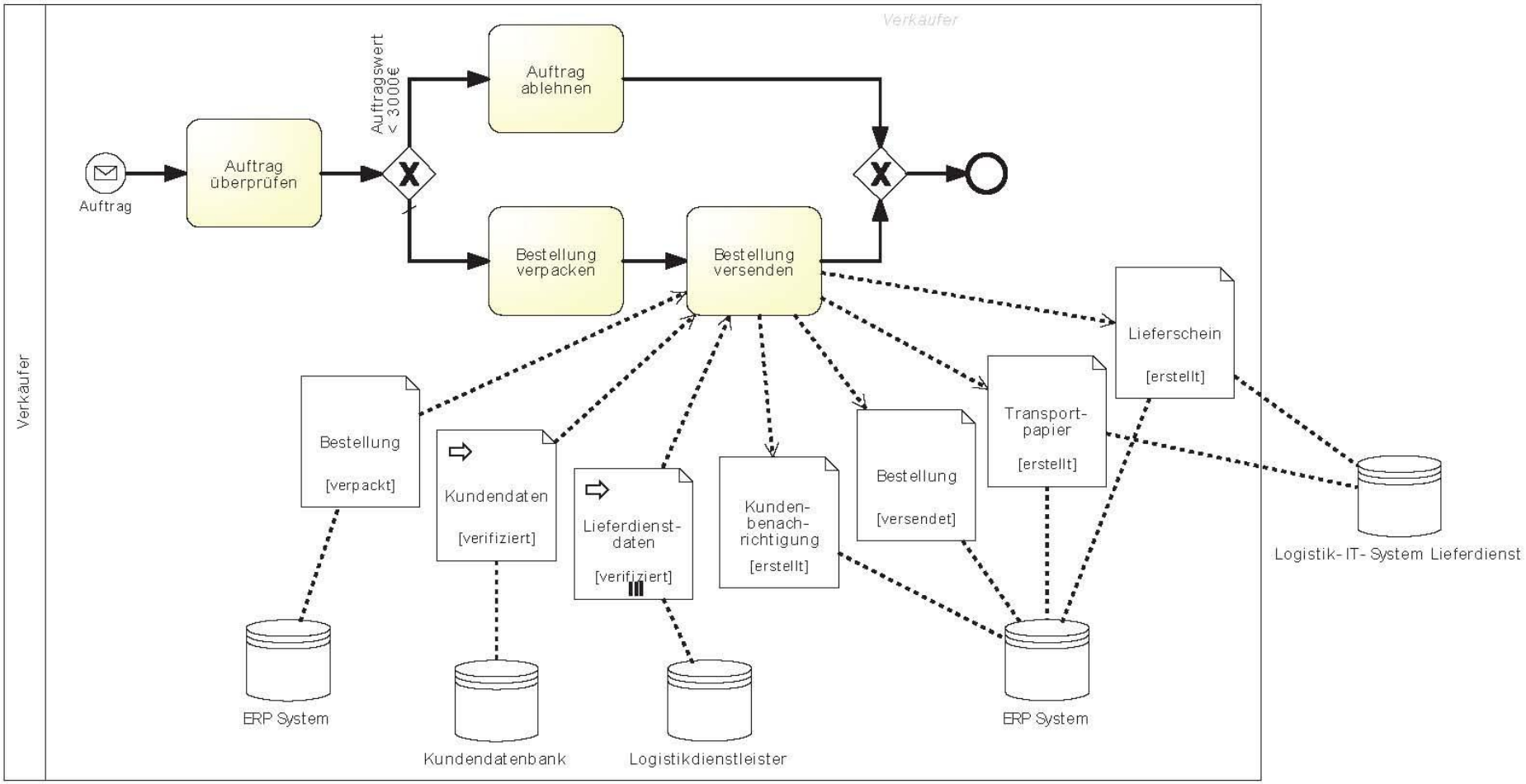


Data in Processes: IT-Systems

- Explicit data objects form an intermediate step to link IT systems
  - Data objects with states
  - IT systems in which the data are stored
- Note
  - Thus, the operational process model is enriched so as to simplify the implementation

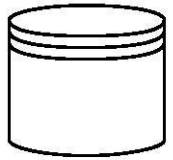


Activities, Data, States, Systems

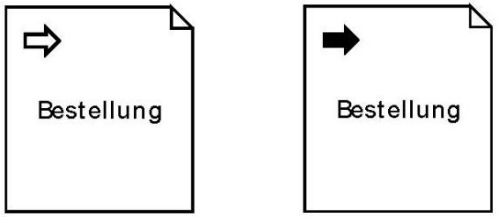


Data Modeling in BPMN

- Data object
  - *“items, physical or information items”*
  - States
  - Data object with external structure: list



- Data store
- Data from an external source for the entire process
  - Data input
  - Data output
- Messages
  - Can refer to data objects identifiers



Data Modeling in BPMN

- Associations
  - Read
  - Write
- Not considered in BPMN
  - Data structures, behavior of objects
    - Using the corresponding UML diagram types
  - *Expression language*
    - Standard XPath and XQuery

Data Modeling in BPMN