

BPMN Meets DMN: Business Process and Decision Modeling

*Week 5:
Data in Business Processes*

An openHPI Course

Prof. Dr. Mathias Weske

Hasso Plattner Institute at the University of Potsdam
Business Process Technology

Week 5

Data in Business Processes

- 5.1 Organizing Process Models
- 5.2 Data and Data Flow
- 5.3 Data Execution Semantics
- 5.4 Structured Data and Sub-Processes
- 5.5 Object Lifecycle Conformance

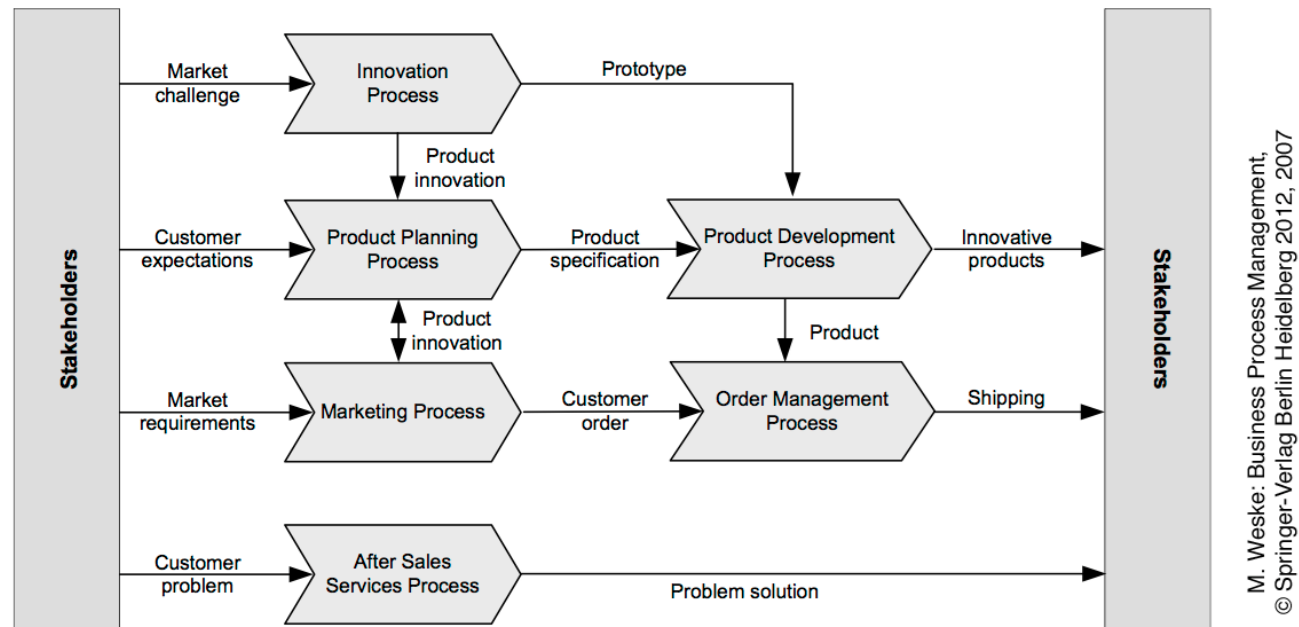
Video Clip 5.1

Organizing Process Models

BPMN Meets DMN: Business Process and Decision Modeling:
An openHPI Course by Mathias Weske

Relationships between Processes

- Observation
 - Organizational business processes are in a supplier/customer relationship



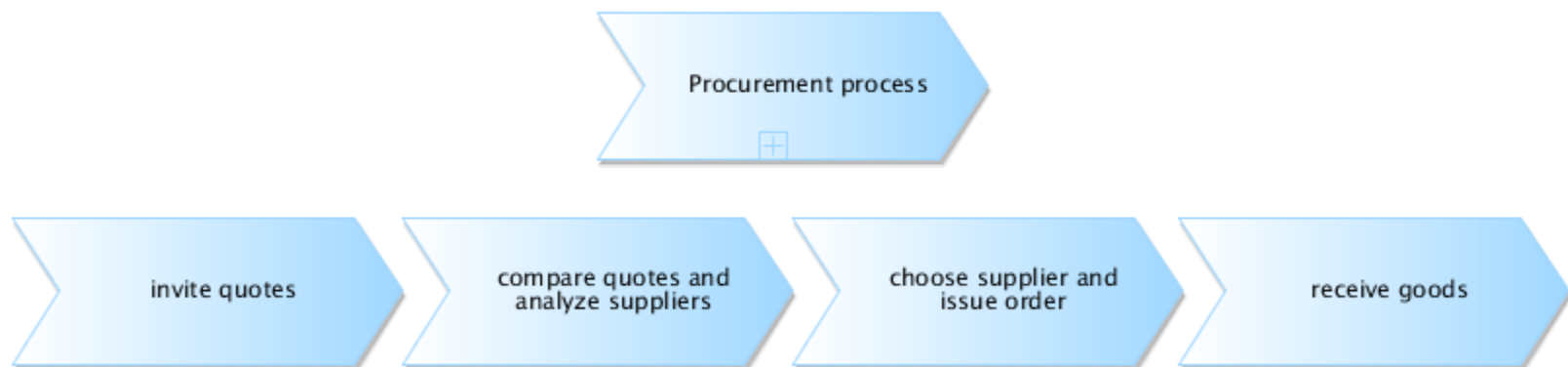
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 2.14. Process landscape relating organizational business processes with stakeholders, based on Schmelzer and Sesselmann (2010)

- This also holds for operational business processes

Relationships between Processes

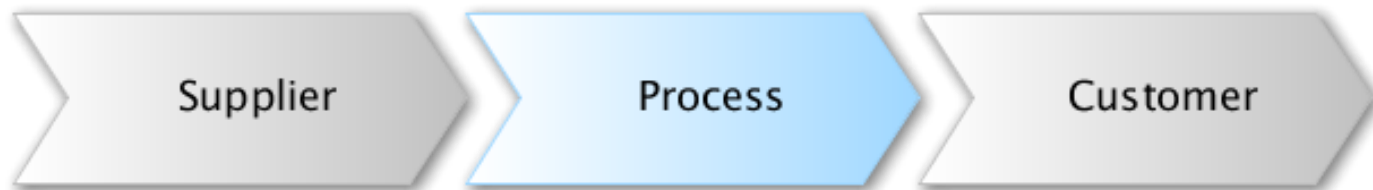
- Background
 - Organizational business processes are often decomposed in a set of operational business processes that are performed sequentially
 - These processes have supplier/customer-relationships
- Example: Procurement process
 - Sub-processes can be refined as operational business processes, e.g., using BPMN diagrams for their representation



Based on Füermann, Dammasch: Prozessmanagement. Hanser-Verlag 2008

Relationships between Processes

- Observation
 - Each business process produces results that are used by the next process in the sequence
 - These dependencies realize supplier/customer-relationships
- Consequences
 - Each business process has at least one customer and is the customer of at least one other process

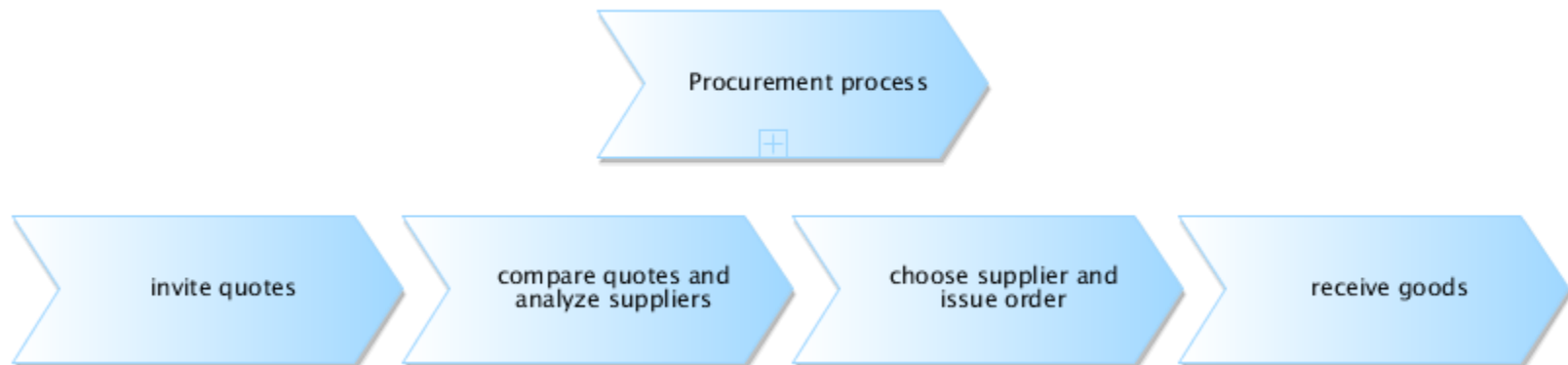


The Process consumes
input (e.g., quotes)

The Process provides
results (e.g., supplier
short list)

Example: Procurement Process

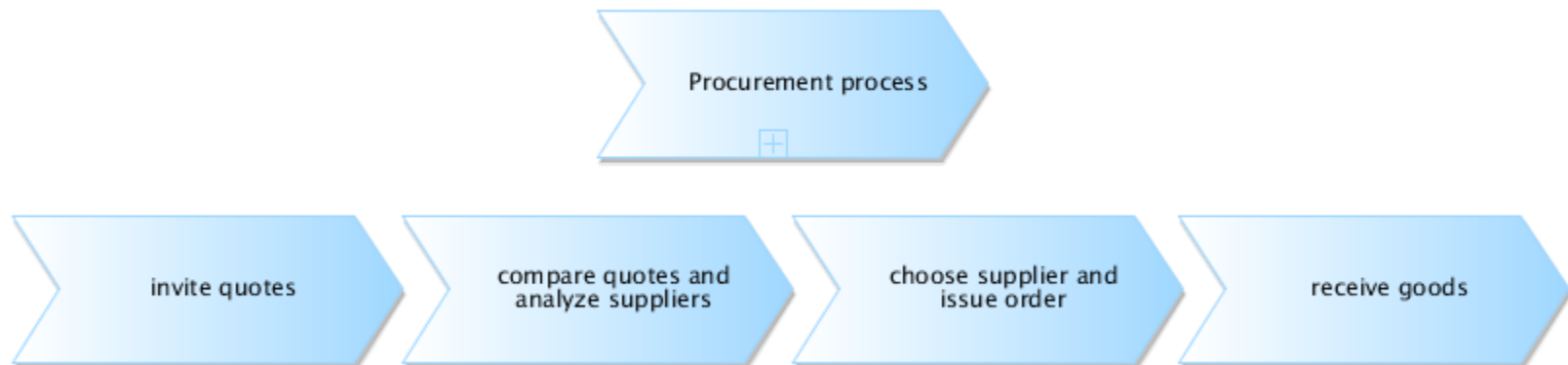
- Process interfaces
 - The results produced by one process (supplier) and consumed by another process (customer) define the process interface between them
 - Process interfaces must be well defined and measurable
- Example: result of “invite quotes” process
 - List of 3-6 quotes with information about supplier, cost and date of delivery



Based on Füermann, Dammasch: Prozessmanagement. Hanser-Verlag 2008

Example: Compare Quotes

- Distinction: Results can be provided by
 - External suppliers (different organizations, partners)
 - Internal suppliers (different departments of the same company)
- Notice
 - In BPMN, suppliers can be expressed by pools and lanes in operational business process models (see Week 4)



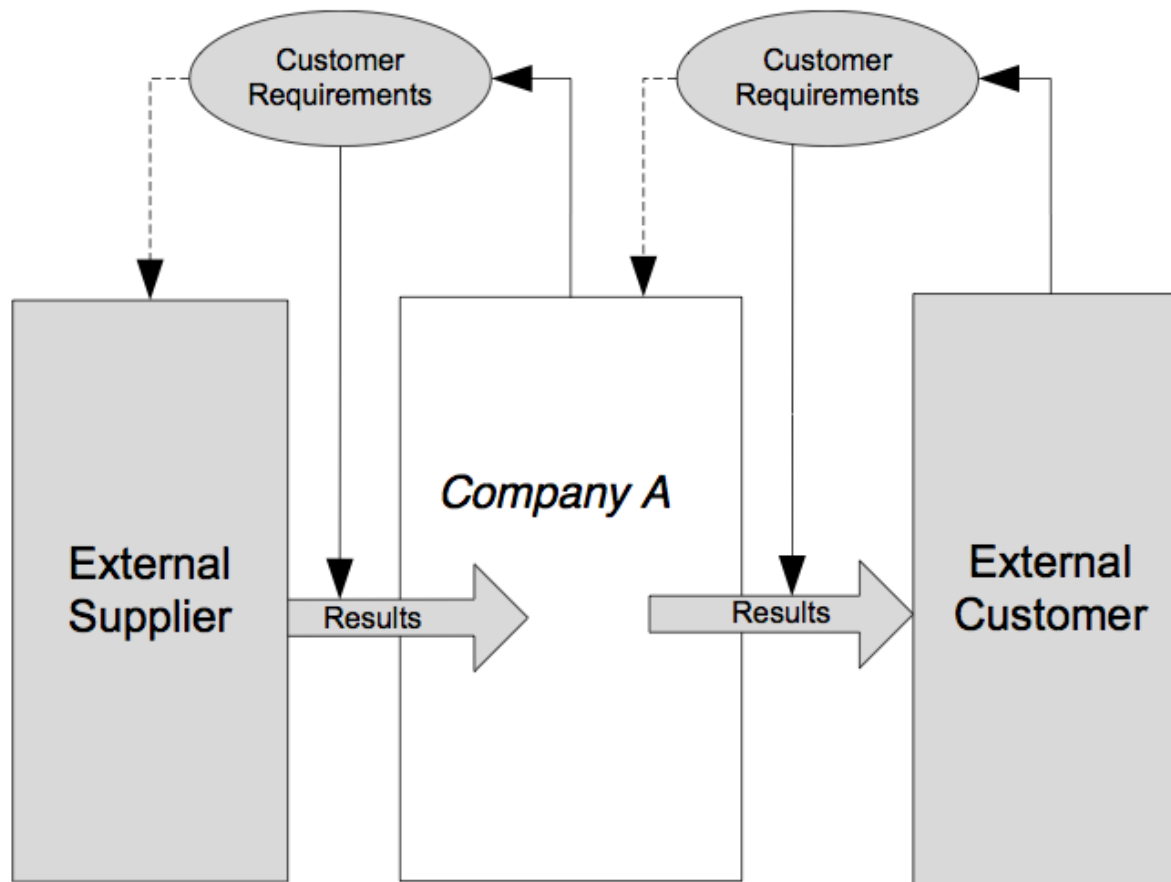
Based on Füermann, Dammasch: Prozessmanagement. Hanser-Verlag 2008

Process Interfaces

- Principle
 - Processes are conducted to produce some result
 - Results are always created for a customer process (internal or external)
 - The quality of produced results is only determined by the requirements of the customer process

*The outcome of a process shall be exactly
what is actually required by the customer process.*

External View: Requirements and Results



M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 8.1. Supplier-customer relationships between companies

Overall View: Requirements and Results

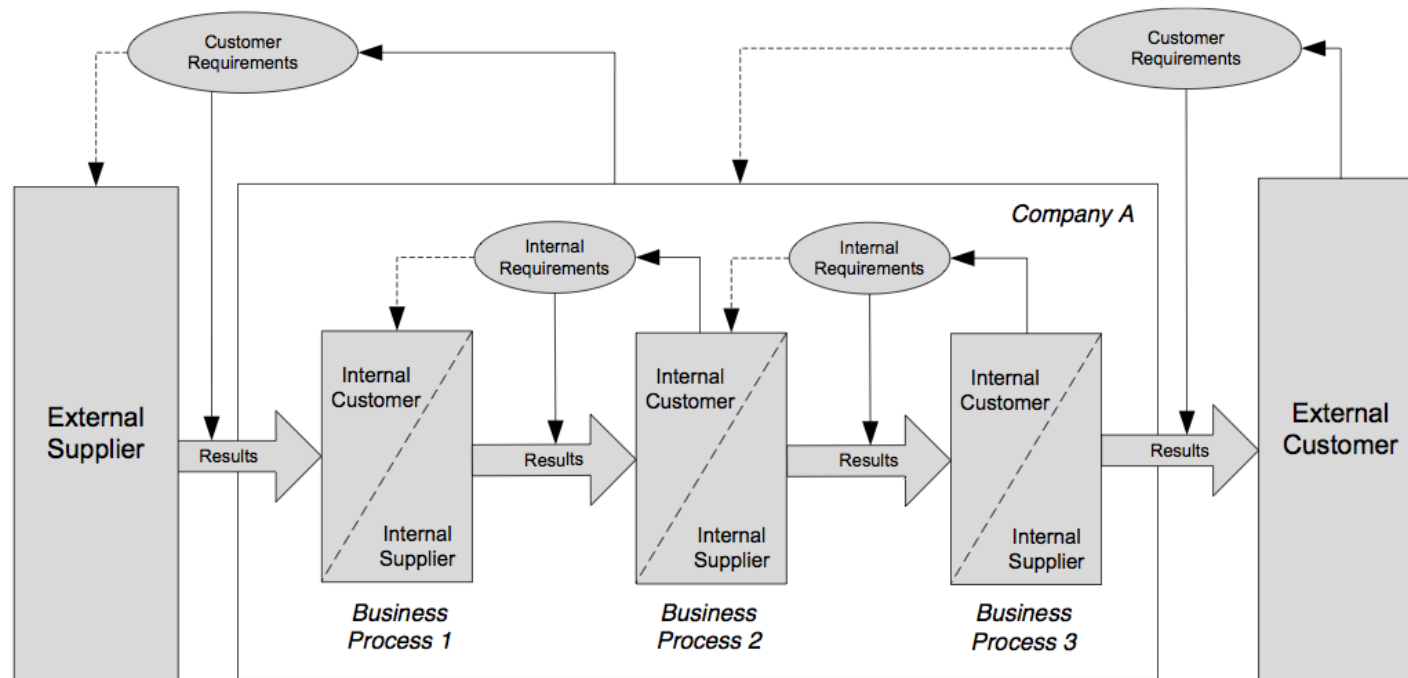


Fig. 8.2. Supplier-customer relationships between processes, based on Füermann and Dammasch (2008)

- Each process ...
 - ... is the customer of a preceding process
 - ... receives input, as required by the process
 - ... produces results as output and thereby creates value

Video Clip 5.1

Organizing Process Models

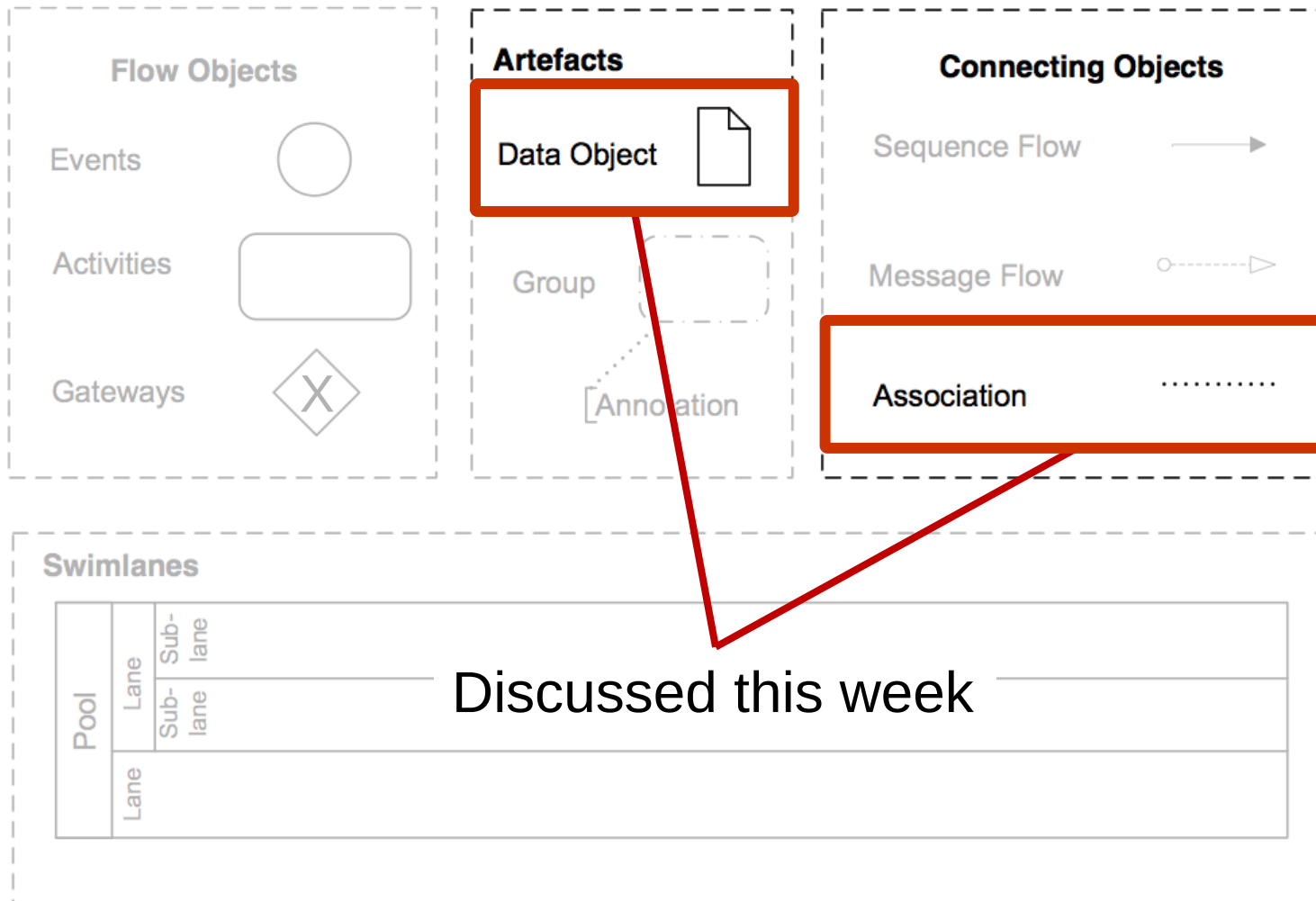
- Processes are conducted to provide results
- Processes are in a supplier/customer relationship
 - Supplier creates output
 - Customer uses it as input
- Requirements always defined by customer process
- Supplier/customer relationship exists also between processes of different companies

Video Clip 5.2

Data and Data Flow

BPMN Meets DMN: Business Process and Decision Modeling:
An openHPI Course by Mathias Weske

Notation Categories and their Elements

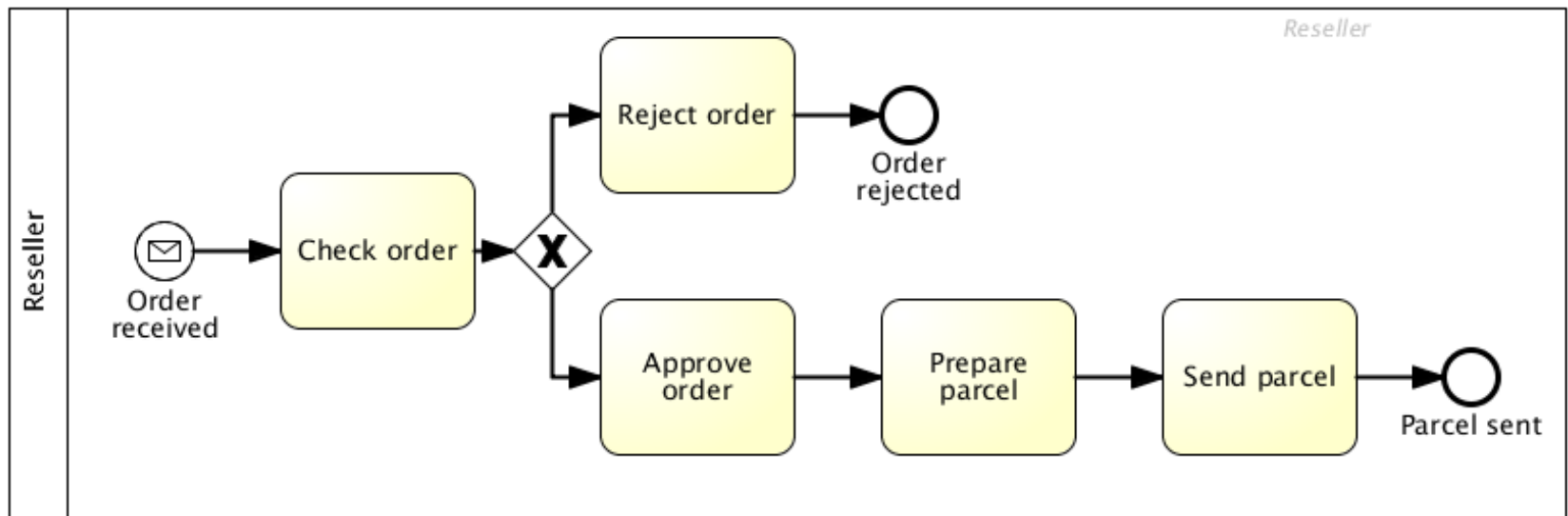


M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Fig. 4.76. BPMN: categories of elements

Motivation for Data in Processes

- Traditionally, the focus of business process modeling is on activities and control flow



- However
 - In process models, data objects appear implicitly as labels of activities and events

Motivation for Data in Processes

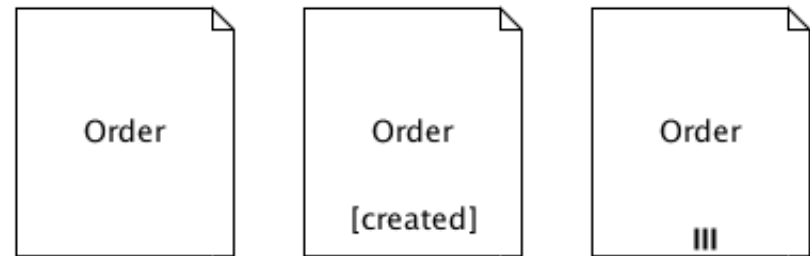
- Execution of business processes
 - Data is created, manipulated and used during the execution of a business process
 - Data serves as input, is created as output, and drives decisions
 - Representing data objects in process models helps bridging the gap between business and IT, during process implementation
- What will be covered
 - Modeling of data objects
 - Dependencies between data objects
 - Possible states of a data object during process execution
 - Relationships between data and process activities

Modeling of Data in BPMN

- Notational elements for representing data

- Data objects

- „items, physical, or information items“
- Data objects with state
- Data objects with external structure: lists

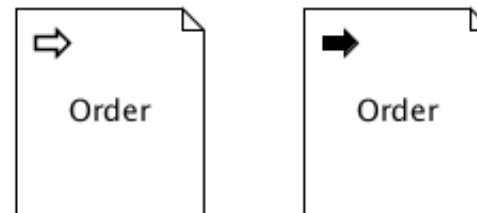


- Data storage



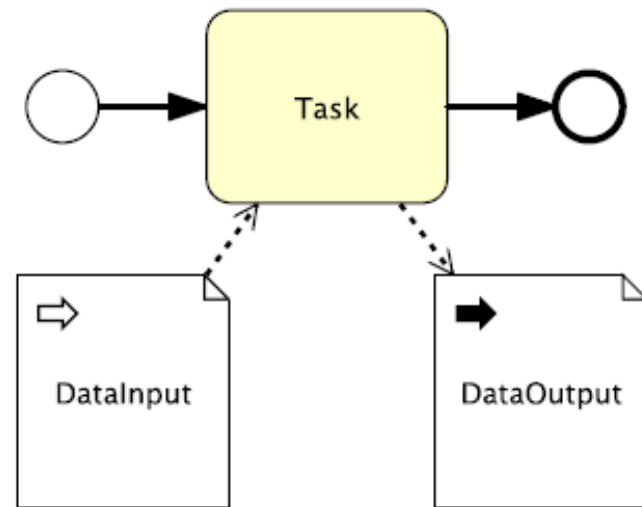
- Data from external source for the entire process

- Data input
- Data output



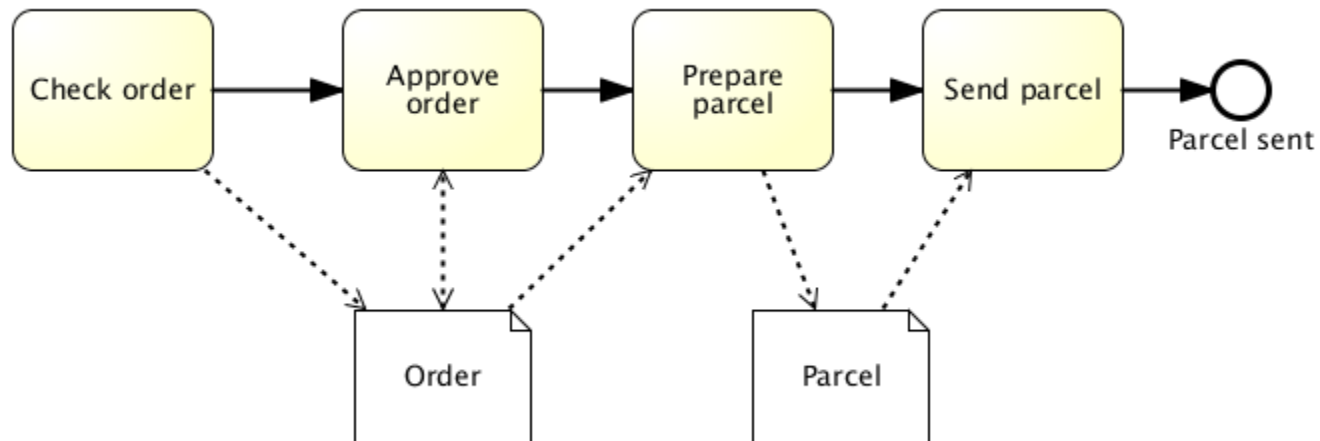
Data Input / Data Output

- Data Input / Data Output represent input / output data to the entire process (or sub-process)
- If applied to a business process
 - Data object existed before the start of the process
 - Data object exists after the termination of the process
- Can also be applied to sub-processes



Representing Data Objects

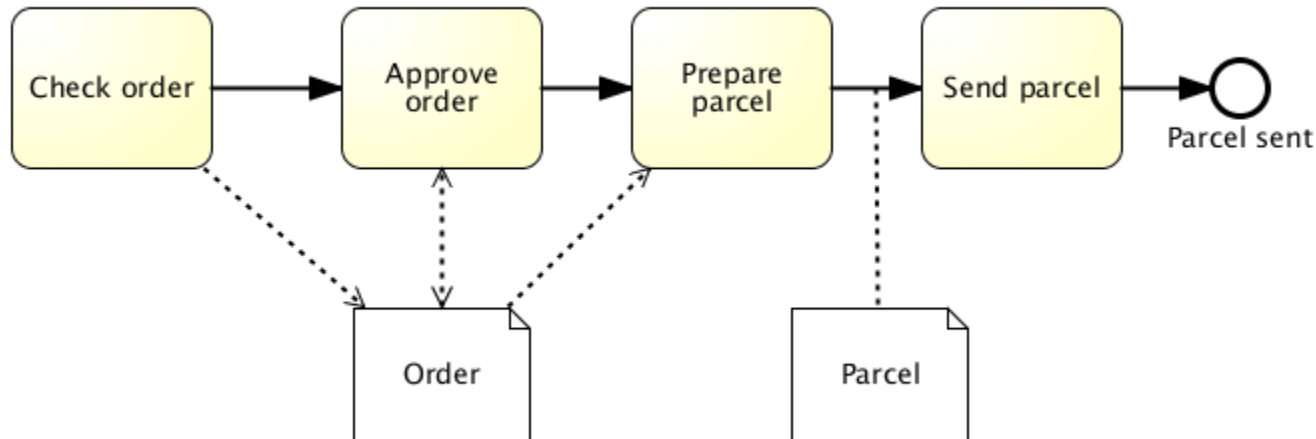
- Dashed edges indicate the reading and writing of data objects by activities



- Notice
 - The term „data object“ might be misleading
 - We represent the type level, not the level of individual data objects

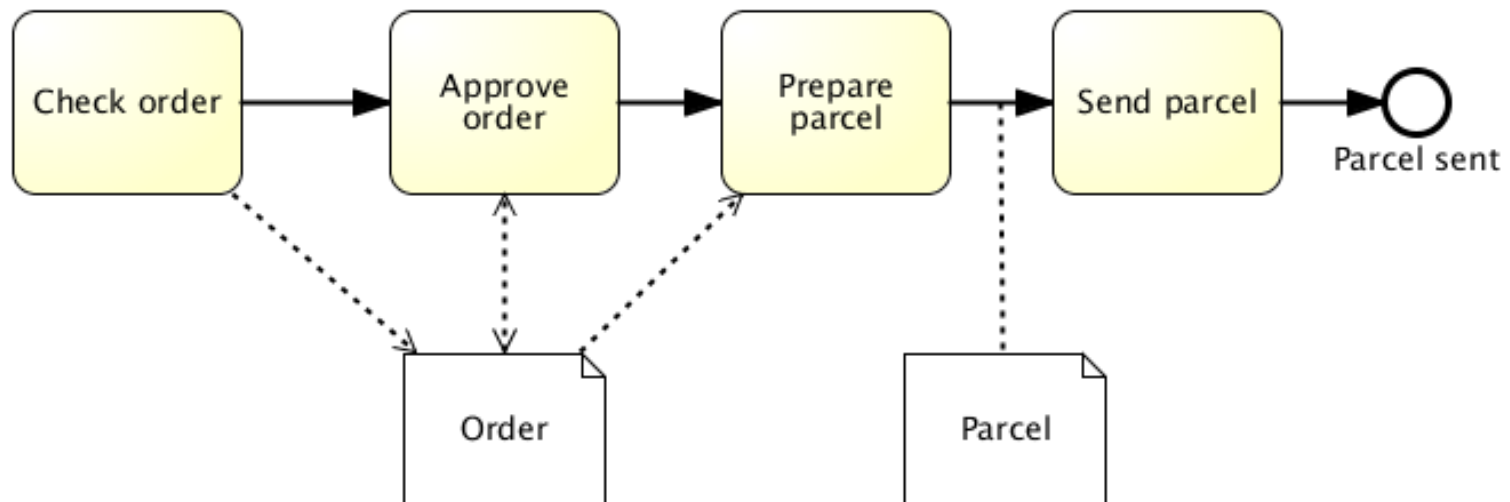
Representing Data Objects

- Data flow between activities related by sequence flow can be expressed in a condensed form



Representing Data Objects

- The structure of data objects is not represented in BPMN
 - The structure of data objects can be modeled, e.g., in UML class diagrams or entity-relationship diagrams

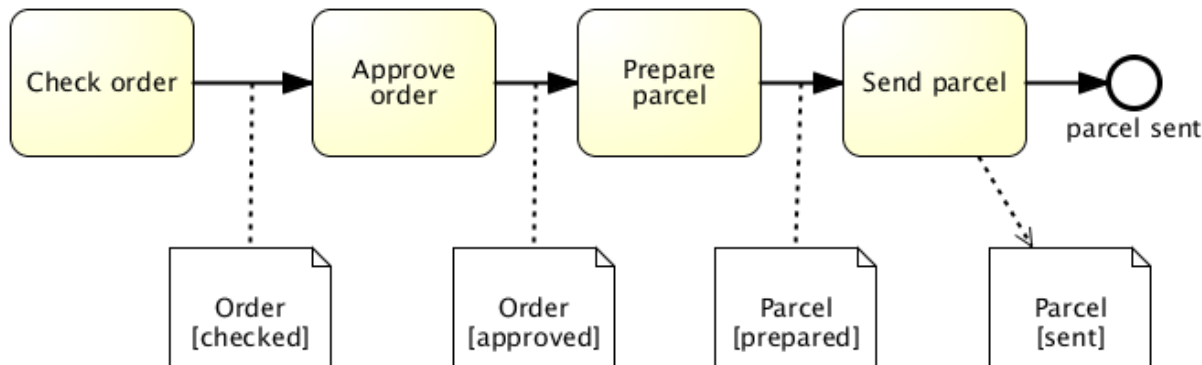
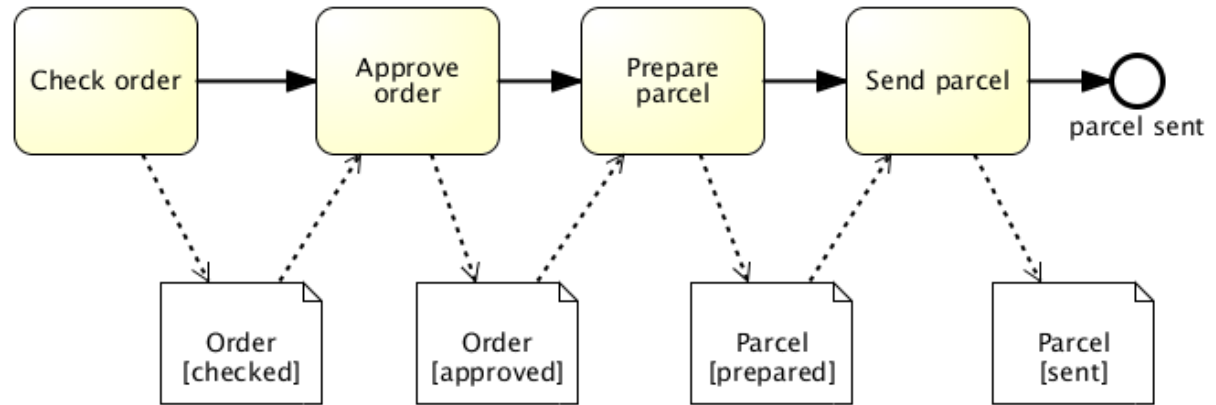


Concurrent Access

- Notice
 - BPMN does not provide means to express transactional properties
 - Managing concurrent access to data needs to be done by IT systems, e.g., database systems

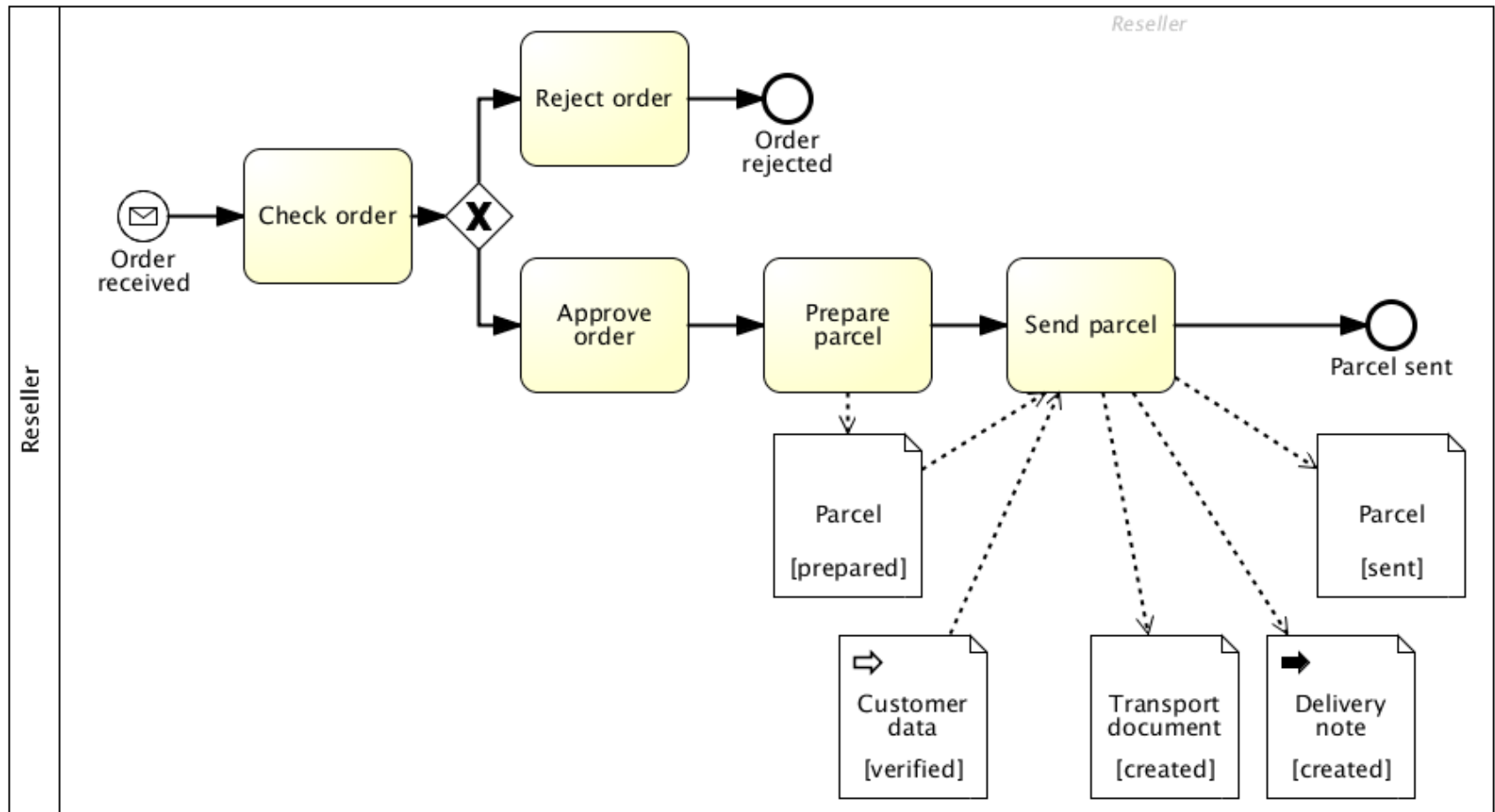
Representing Data Object States

- Data objects can have a state, i.e., a representation of its current value
 - State transitions are triggered by activities, writing data objects



Example: Data Objects

- The *order fulfillment* process acts on several data objects



Video Clip 5.2

Data and Data Flow

- Data drives the execution of business processes
- Data objects in BPMN
 - Used to represent information items and physical items
- Data objects have states and state transitions
- Data objects can be
 - Input/output of activities, i.e., they are local to a process instance
 - Input/output of processes, i.e., they exist beyond the process instance

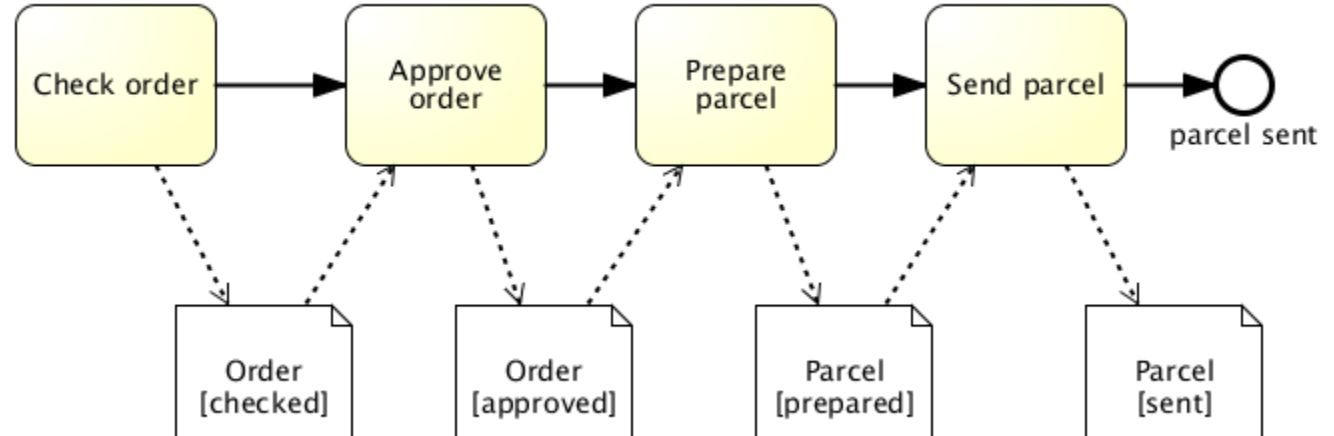
Video Clip 5.3

Data Execution Semantics

BPMN Meets DMN: Business Process and Decision Modeling:
An openHPI Course by Mathias Weske

Execution Semantics

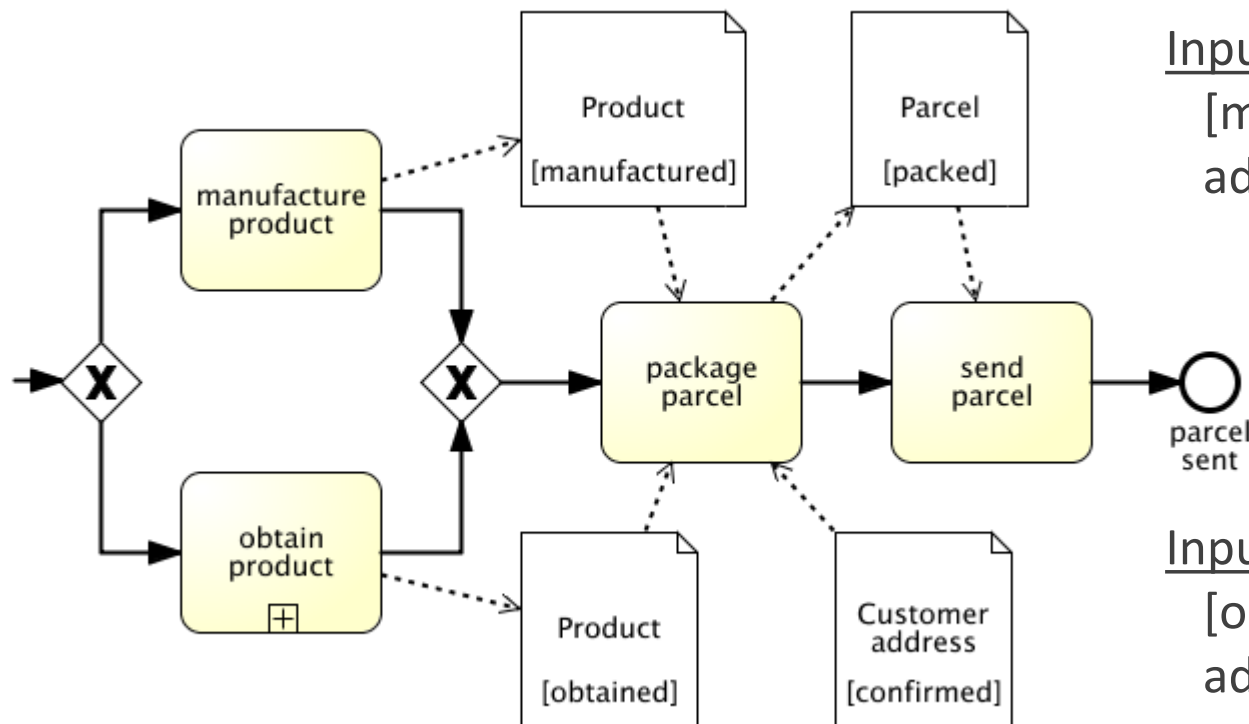
- So far, only sequence flow decides if and when an activity can be enabled
- Data influences execution semantics of processes as well
 - An activity is enabled only if the data that the activity reads is available and in the required state



- In case of multiple data objects to read, additional definitions are required

Input Sets

- If an activity has multiple incoming read edges, input sets are used to refine the enabling of that activity
- Example: “package parcel”



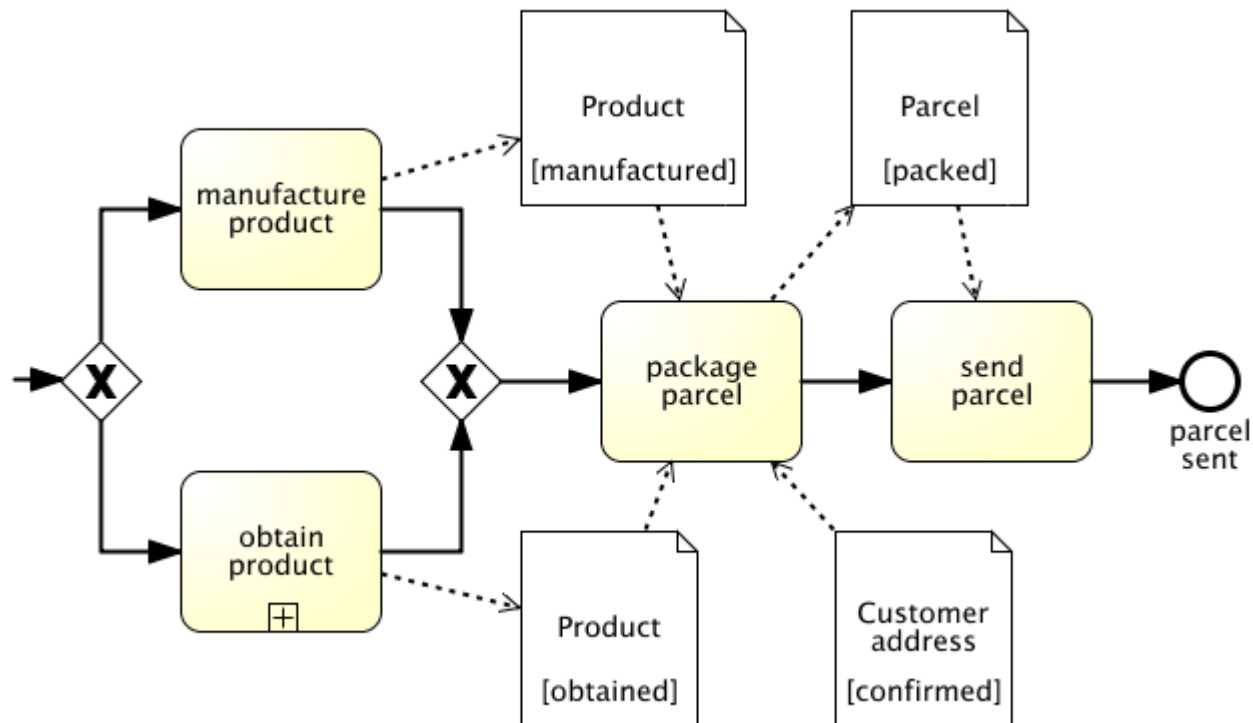
InputSets:

InputSet1 = {Product
[manufactured], Customer
address [confirmed]}

InputSet2 = {Product
[obtained] }, Customer
address [confirmed]}

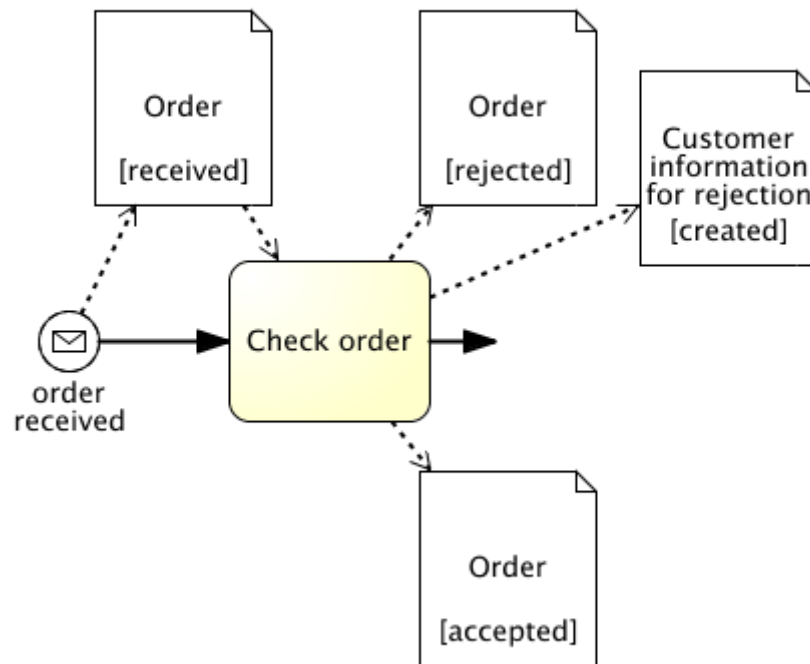
Input Sets

- More concrete
 - If sequence flow is signaled, the input sets of the activity are evaluated
 - If an input set is available, the activity is enabled



Output Sets

- If an activity has multiple outgoing write edges, output sets can refine the output of that activity



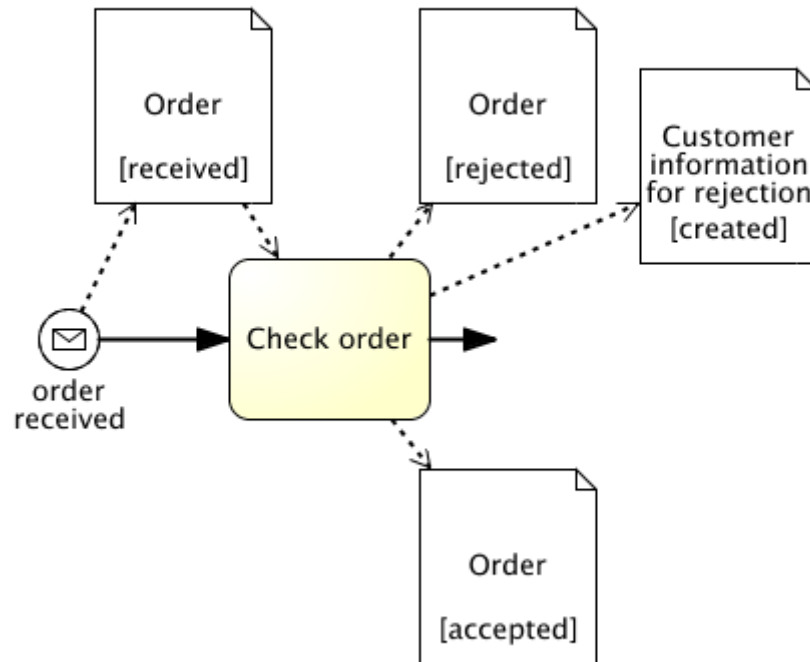
OutputSets:

OutputSet1 = {Order [rejected],
Customer information for
rejection[created]},

OutputSet2 = {Order[accepted]}

Output Sets

- More concrete
 - Output sets refer to a collection of data objects with states, which can be available after the execution of the activity
 - After activity termination one of these output sets is available



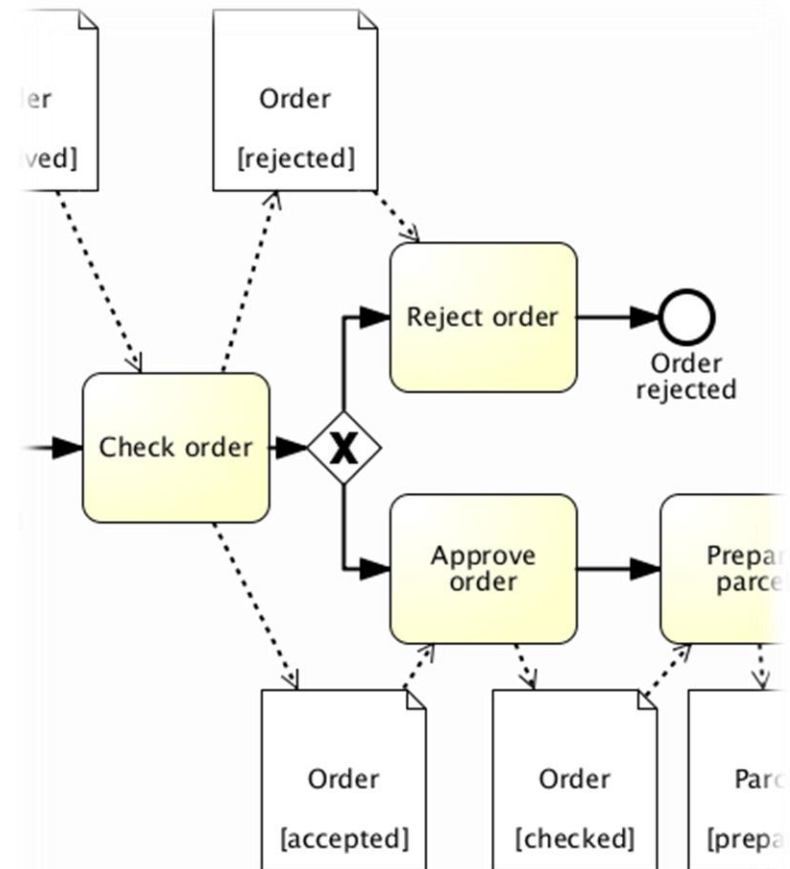
OutputSets:

OutputSet1 = {Order [rejected],
Customer information for
rejection[created]},

OutputSet2 = {Order[accepted]}

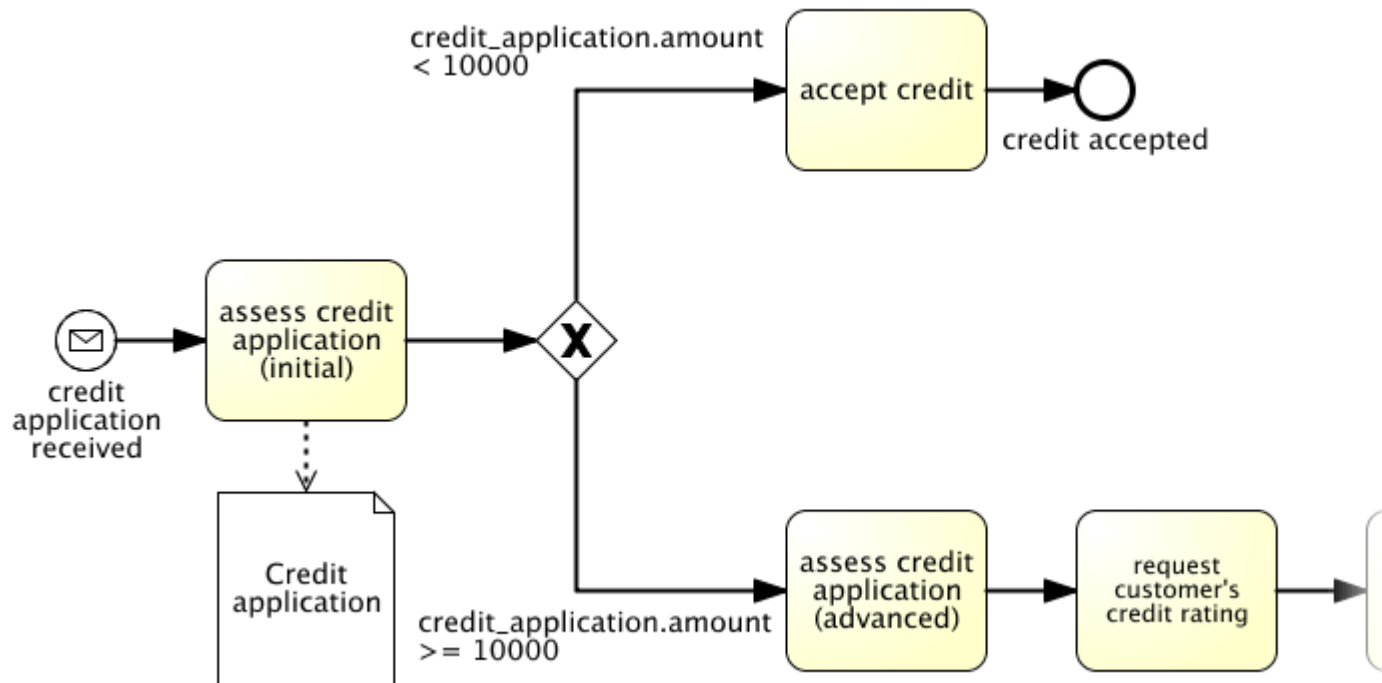
Data Objects and Decisions

- Decisions of XOR gateways can use data objects (data-based exclusive gateway)
- Example
 - Decision is based on the state of the *Order* data object
 - *Check order* has two output sets



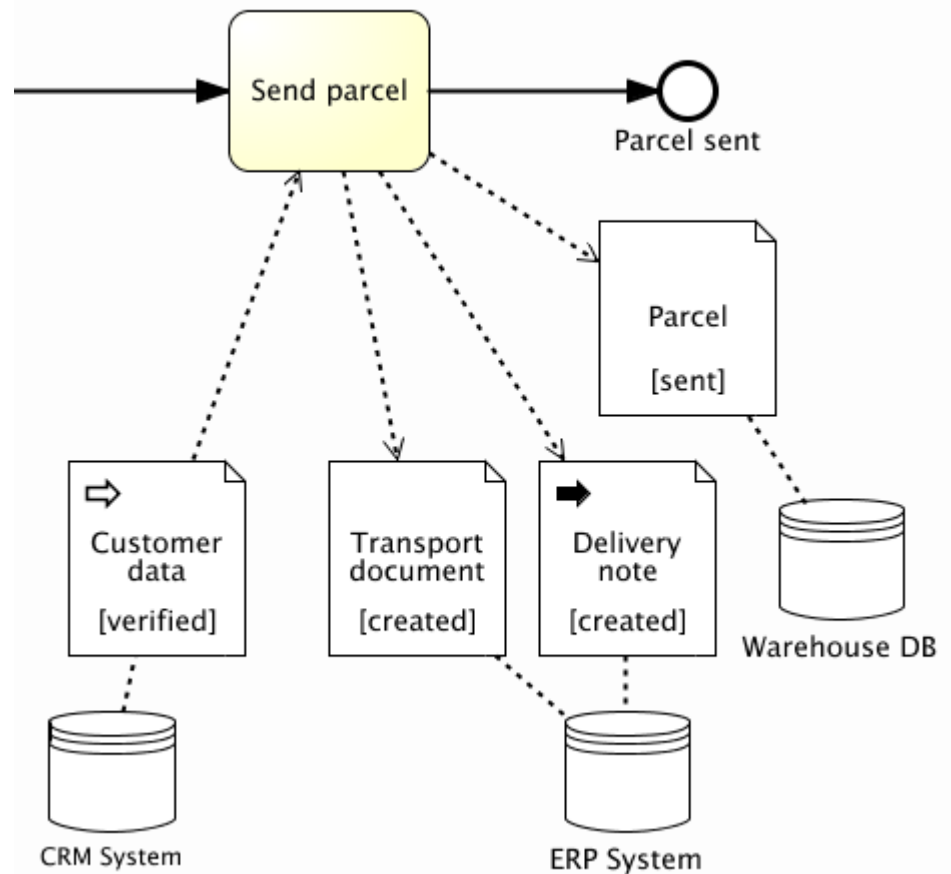
Data Objects and Decisions

- Decisions might also use values of data objects
 - Notice that the conditions are not completely formalized
 - However, the goal of the decision is expressed well in the process model



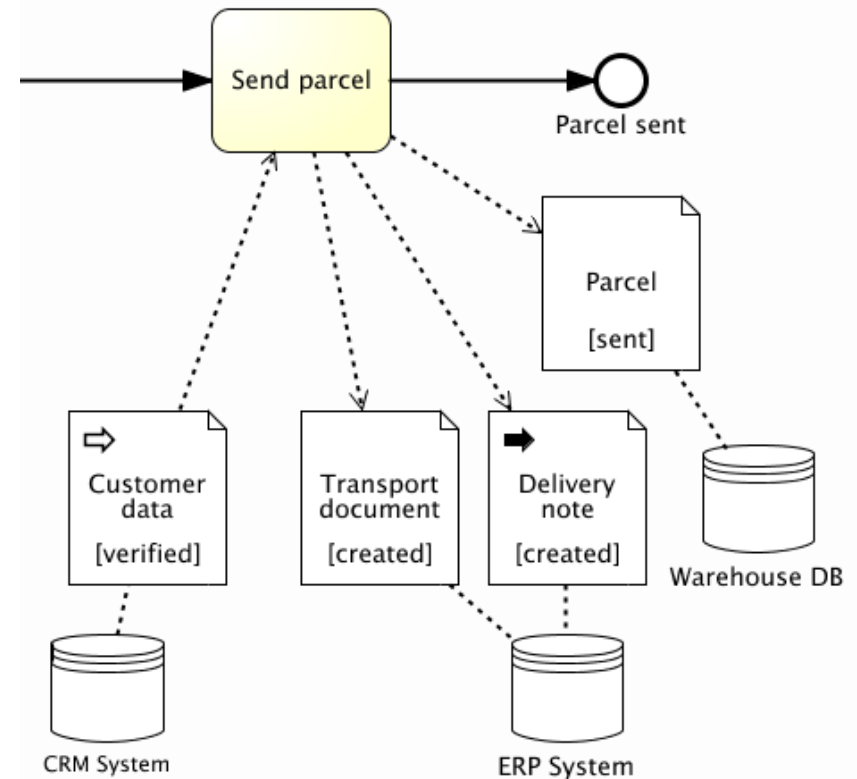
Data in Processes: Data Storage

- Data storage describes locations where data is stored and retrieved from

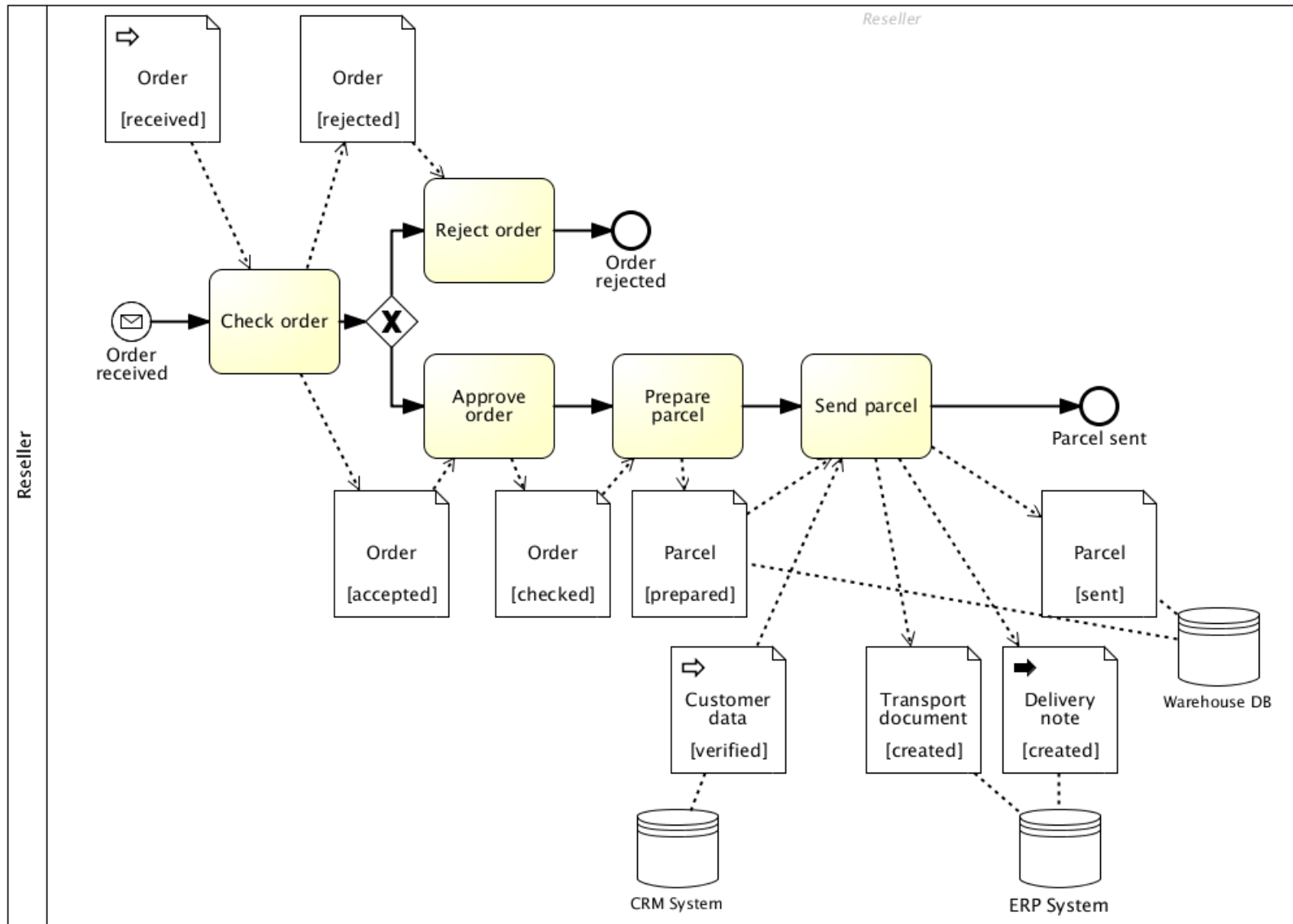


Persistent and Transient Data

- Persistent
 - Data objects are available after the process instance has terminated
- Transient
 - Data objects are created during the process instance and are not available after its termination
- In BPMN, all data objects are transient, unless they are
 - associated with a data storage
 - data input or data output of a business process



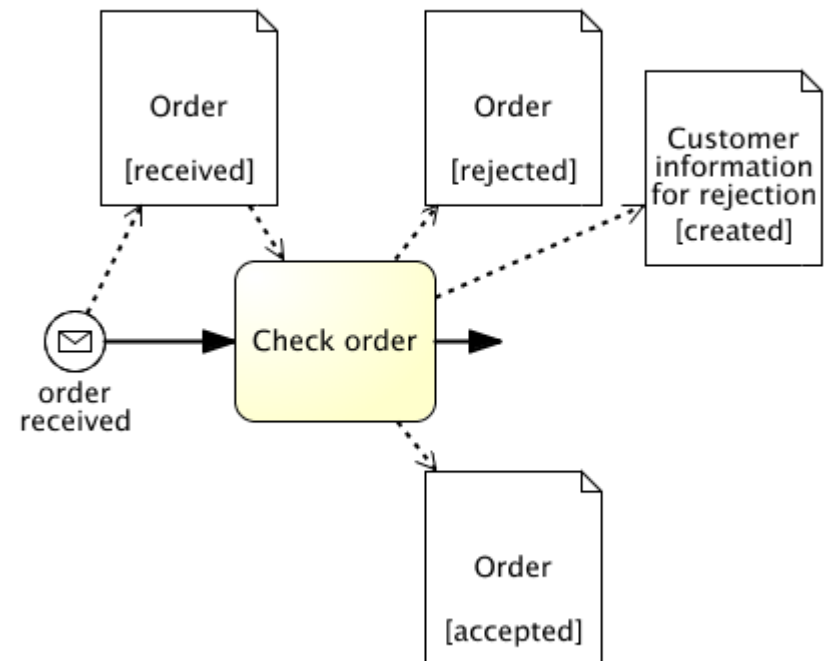
Activities, Data Objects, States, Systems



Video Clip 5.3

Data Execution Semantics

- Data objects influence processes
- Data objects might be required for executing activities
 - Input sets and output sets for detailed definition
 - Data objects impact decisions
- Data stores provide persistent storage
- Data input / data output
 - can be used to reflect supplier/customer relationships



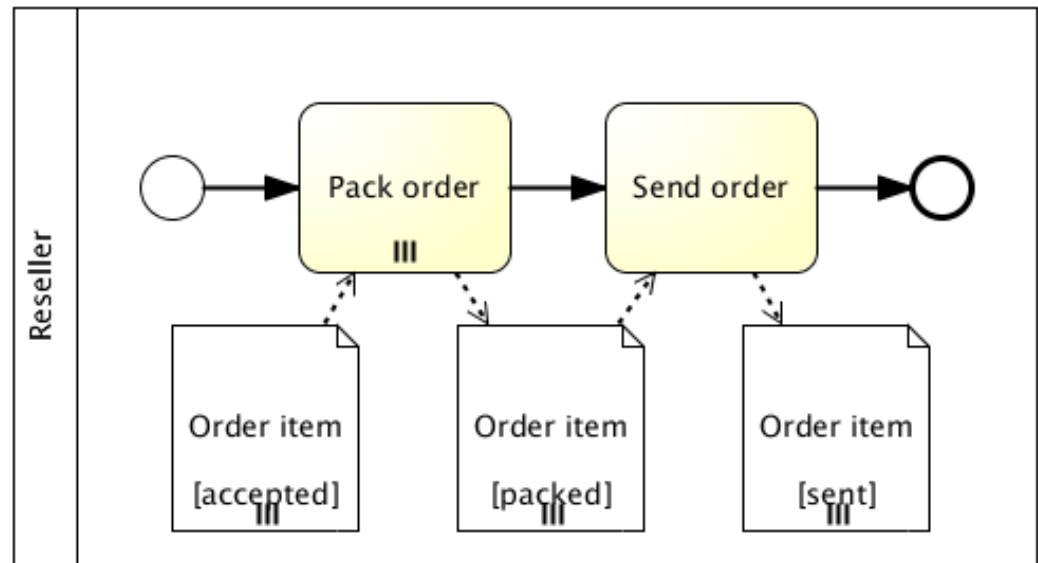
Video Clip 5.4

Data Objects and Sub-Processes

BPMN Meets DMN: Business Process and Decision Modeling:
An openHPI Course by Mathias Weske

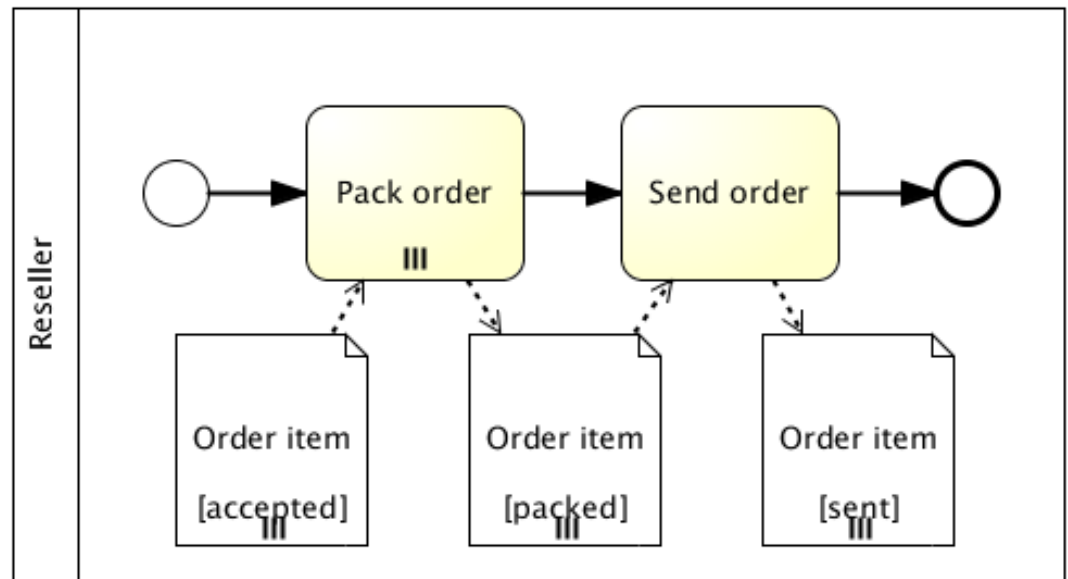
Processing of Data Object Collections

- Idea
 - Data objects defined as collection can be processed by multiple instances (MI) activities
- Input
 - Each data object of a collection is processed by one activity instance
 - Parallel or sequential execution possible
- Example
 - *Pack order*
 - Parallel execution



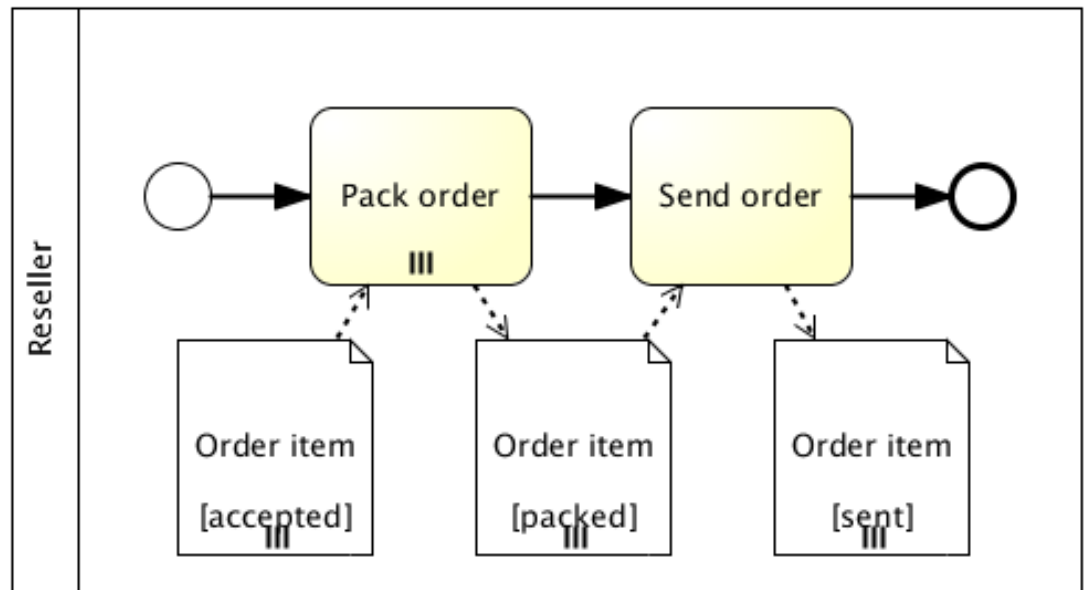
Processing of Data Object Collections

- Output
 - Each activity instance creates a data object for an output collection
- Example
 - The state of each *Order item* is changed from *accepted* to *packed* by the *Pack order* MI activity



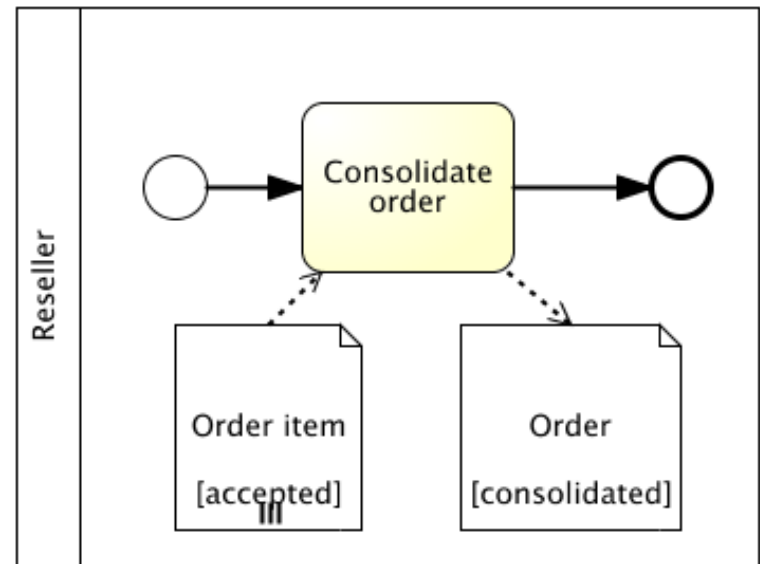
Processing of Data Object Collections

- Input, output of data object collections
 - Activities can also process collections of data objects
- Example
 - The *Send order* activity reads the *Order item* data collection in state *packed* and changes the state of each order item to *sent*



Processing of Data Object Collections

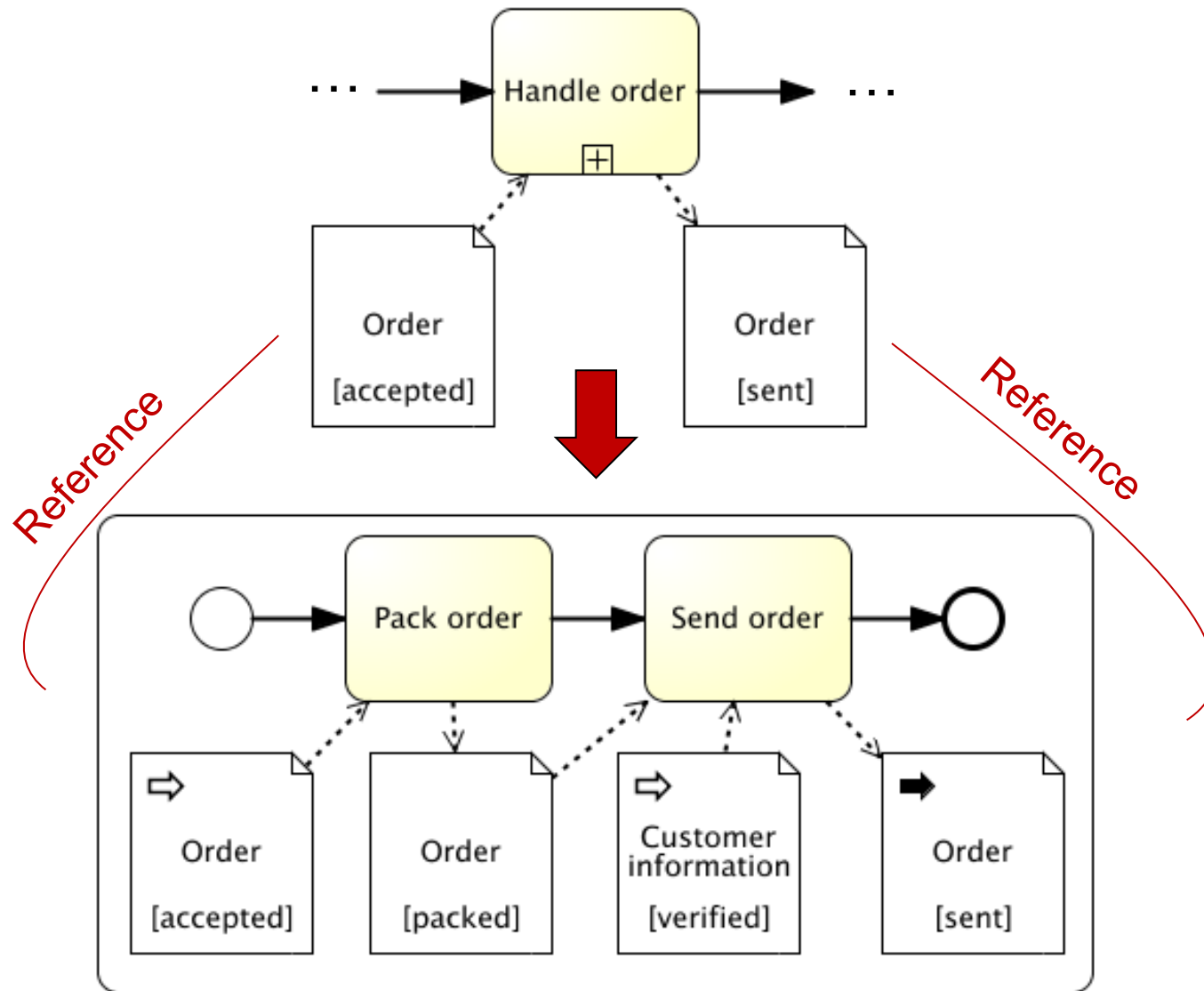
- Input / output consolidation
 - An activity can read a data object collection and create just one output data object
- Example: Consolidate order
 - After termination of the activity instance, a new data object *Order* in state *consolidated* is created



Data in Processes and Sub-Processes

- Transfer of the data objects between a sub-process activity and its child process
- Approach
 - Manual assignment of a data object on the process level to a data object on the sub-process level
 - Use of references
- Attention
 - Assignment of, e.g., *Order* to *Invoice* is possible
 - No consistency check, modeler has the responsibility to preserve consistency
- Notice
 - Data input and data output can also be applied to sub-processes

Context of Data Objects



Video Clip 5.4

Data Objects and Sub-Processes

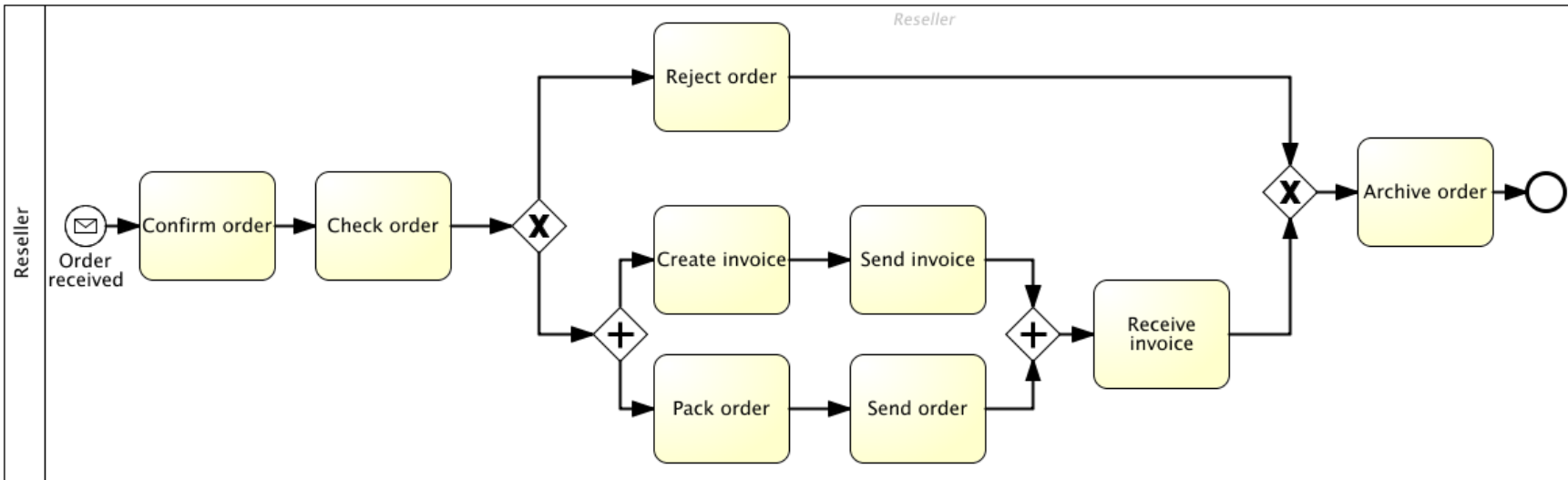
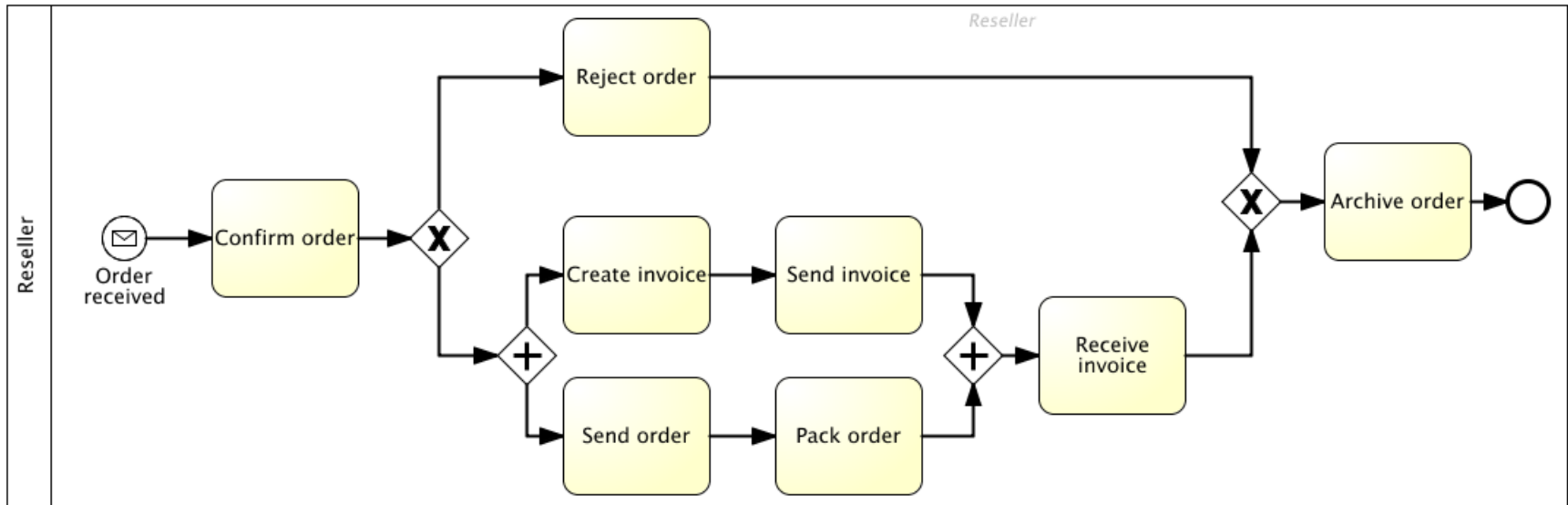
- Data object collections can be processed by activities
- Multiple instance activities
 - process each element of the collection in one activity instance
- Single instance activities
 - process the collection as a whole
- Data can be transferred to sub-processes

Video Clip 5.5

Object Lifecycle Conformance

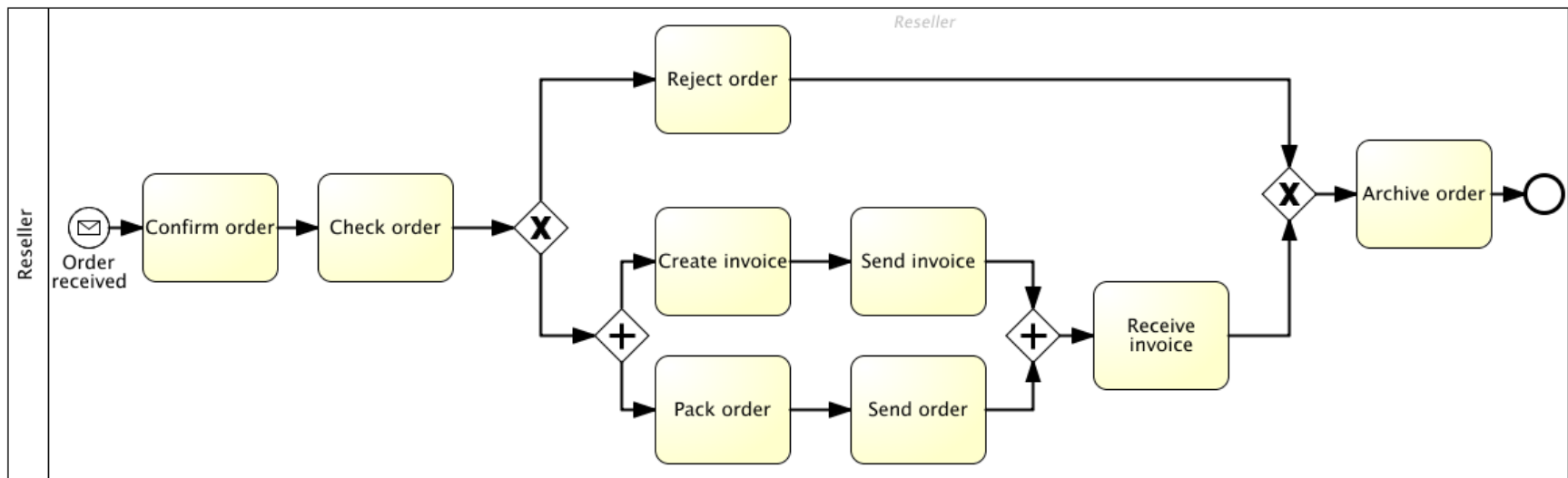
BPMN Meets DMN: Business Process and Decision Modeling:
An openHPI Course by Mathias Weske

Motivation



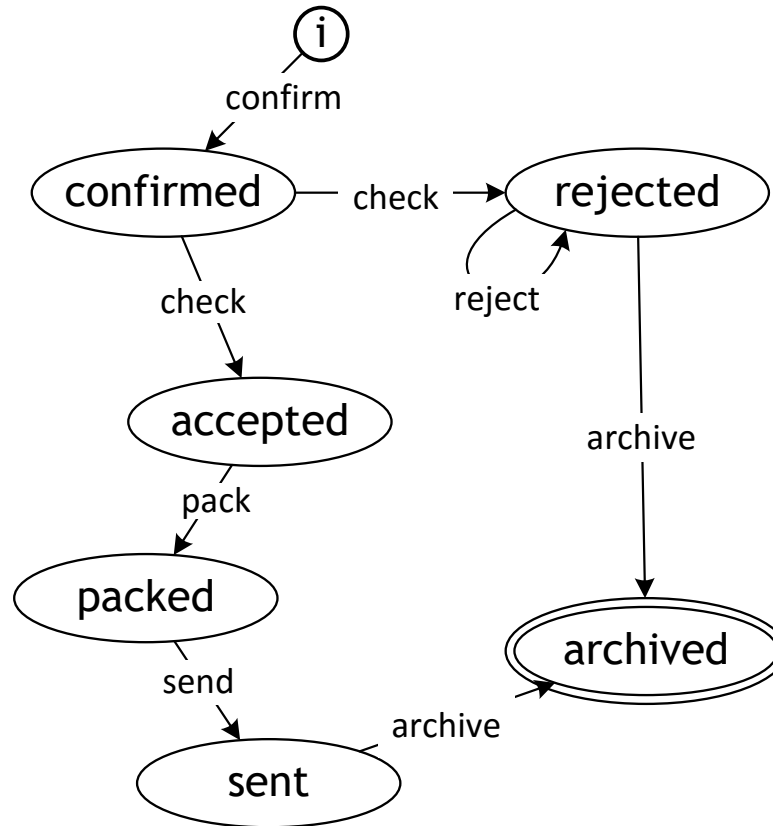
Motivation

- Data objects change their state during process executions
- These state transitions need to comply to the data object behavior
 - Can be specified using data object lifecycles, specified as state transition diagrams
 - We assume that data object lifecycles are available



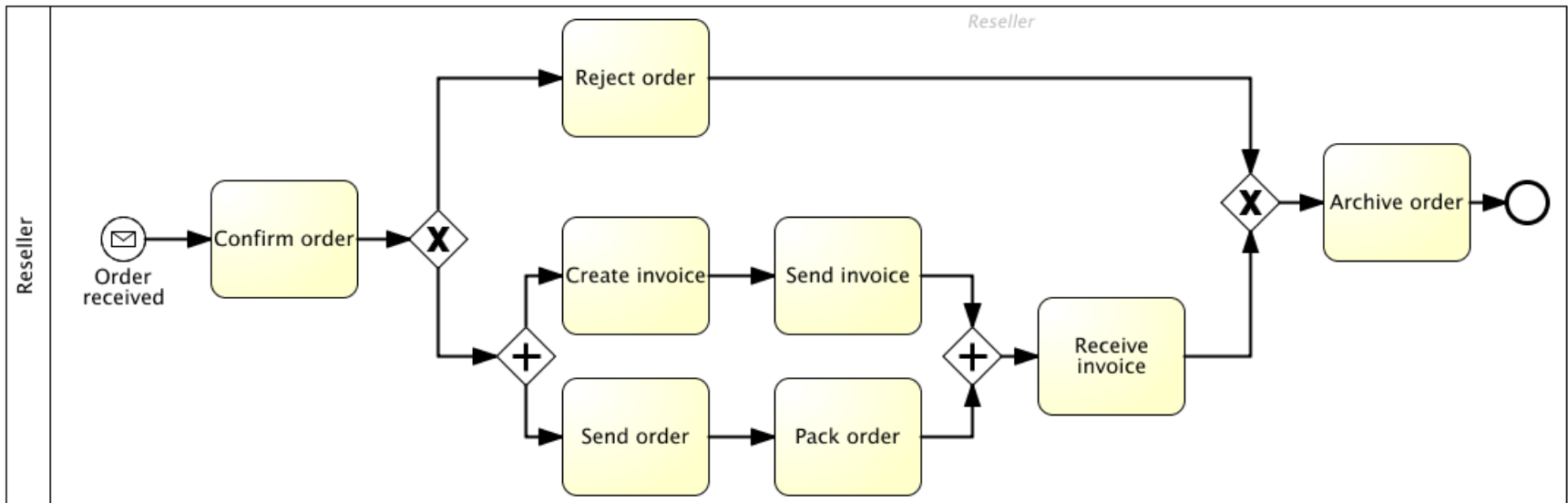
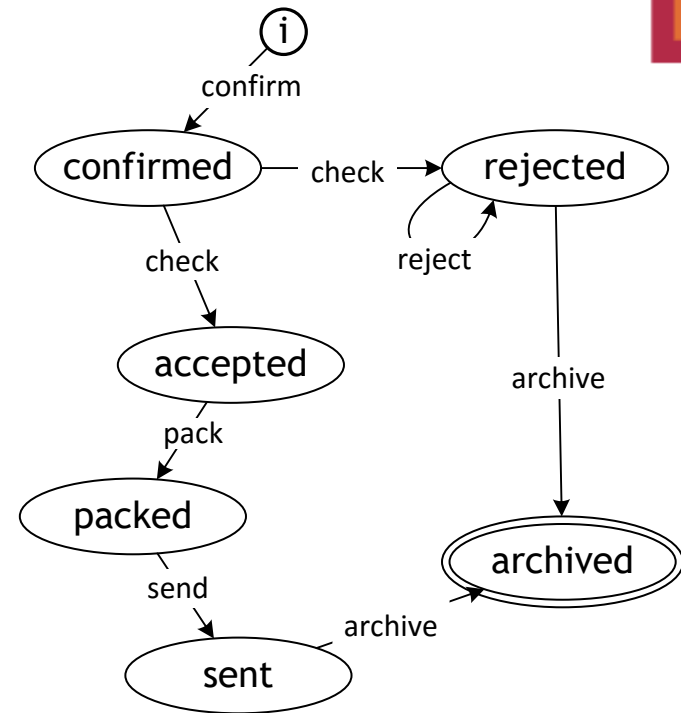
Motivation

- Example: object lifecycle of *Order*



Conformance Example

- We can now find inconsistencies
 - Missing *Check order* activity
 - Wrong sequence of *Send order* and *Pack order* activities



Object Lifecycle Conformance

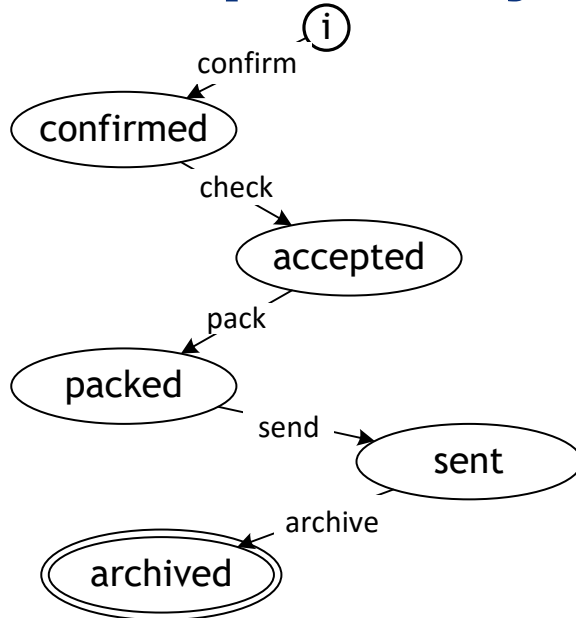
- Idea
 - Define property for checking consistency between data objects and processes
- *Definition*
 - *A process conforms to a data object lifecycle, if the process includes only state transitions which are defined by the data object lifecycle.*

Object Lifecycle Conformance

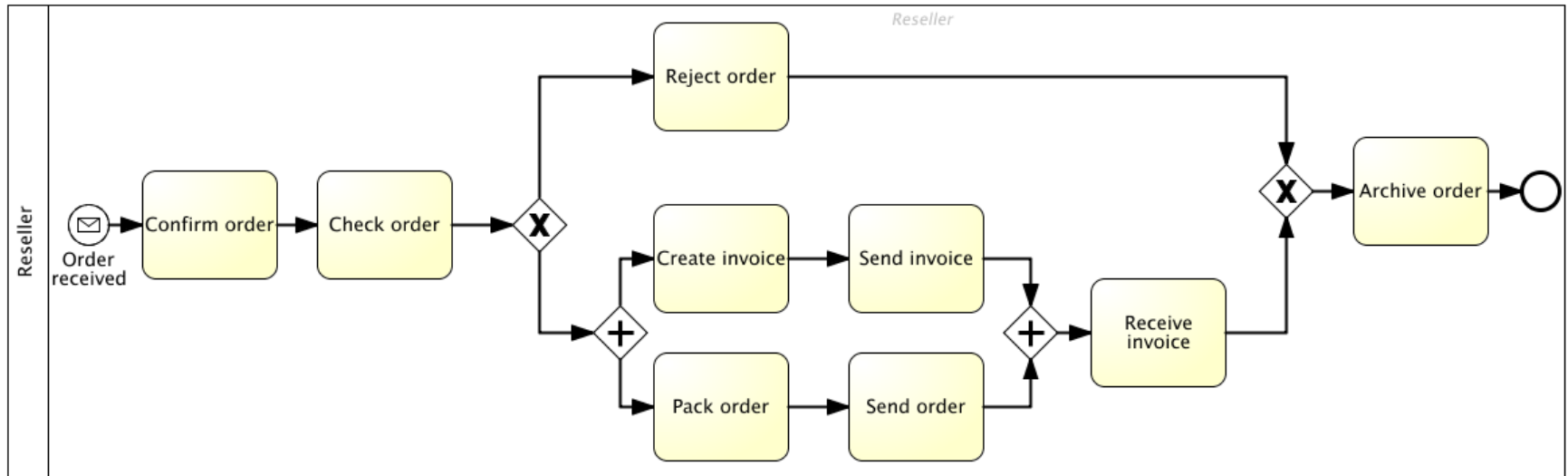
Process P fulfills Object Lifecycle Conformance for data object O , if the following holds

- For each state transition of object O in process P by some process activity, there exists a corresponding state transition in the object lifecycle of O*
- Initial state and final state of the object O in process P are also part of the object lifecycle of O*

Example, Object Lifecycle 1

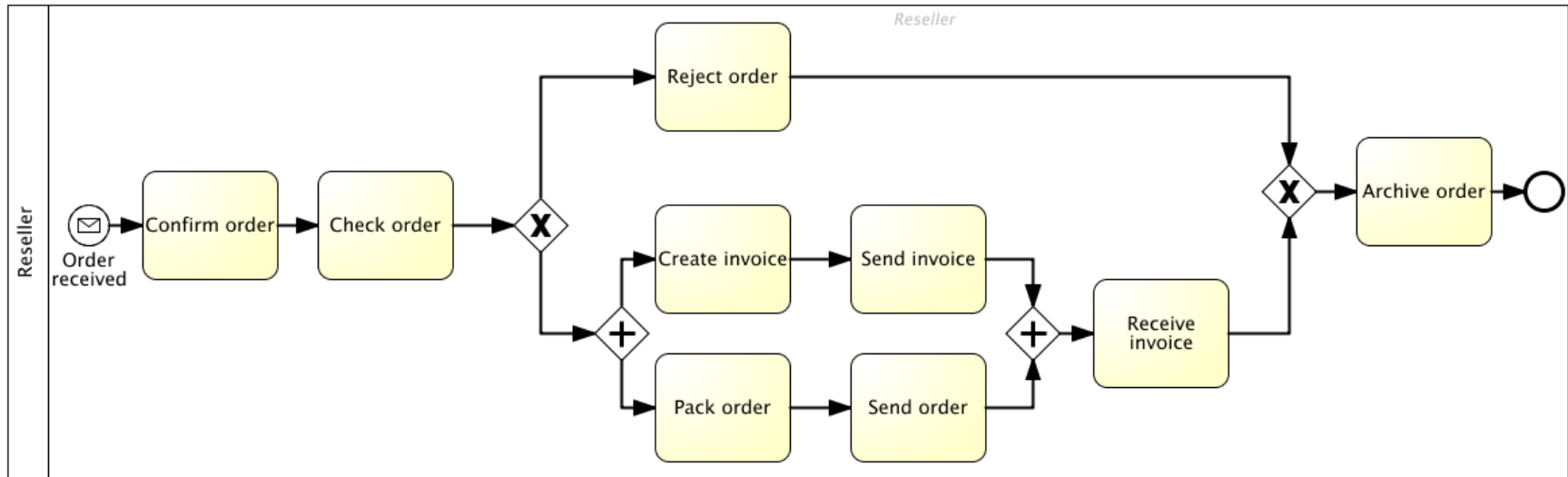
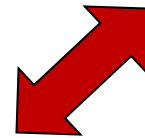
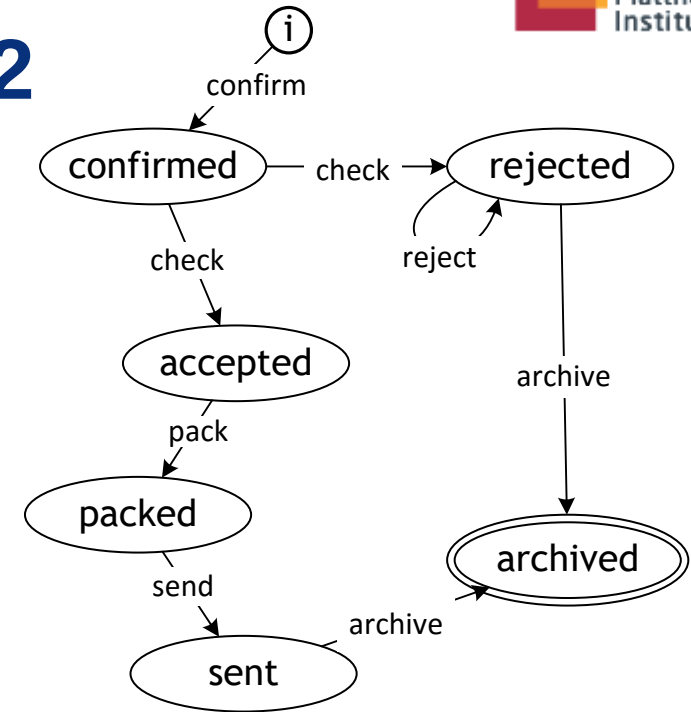


- Object lifecycle conformance is not fulfilled
 - State transition *reject* of data object *Order* can not be found in the object life cycle

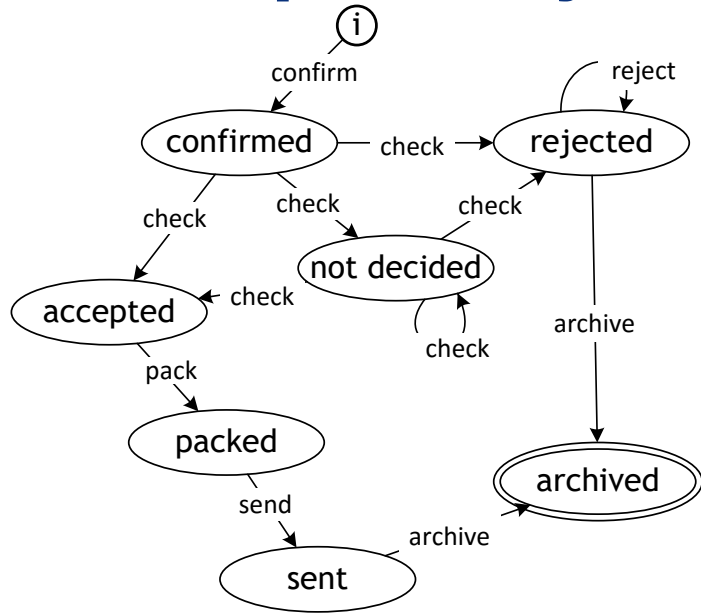


Example, Object Lifecycle 2

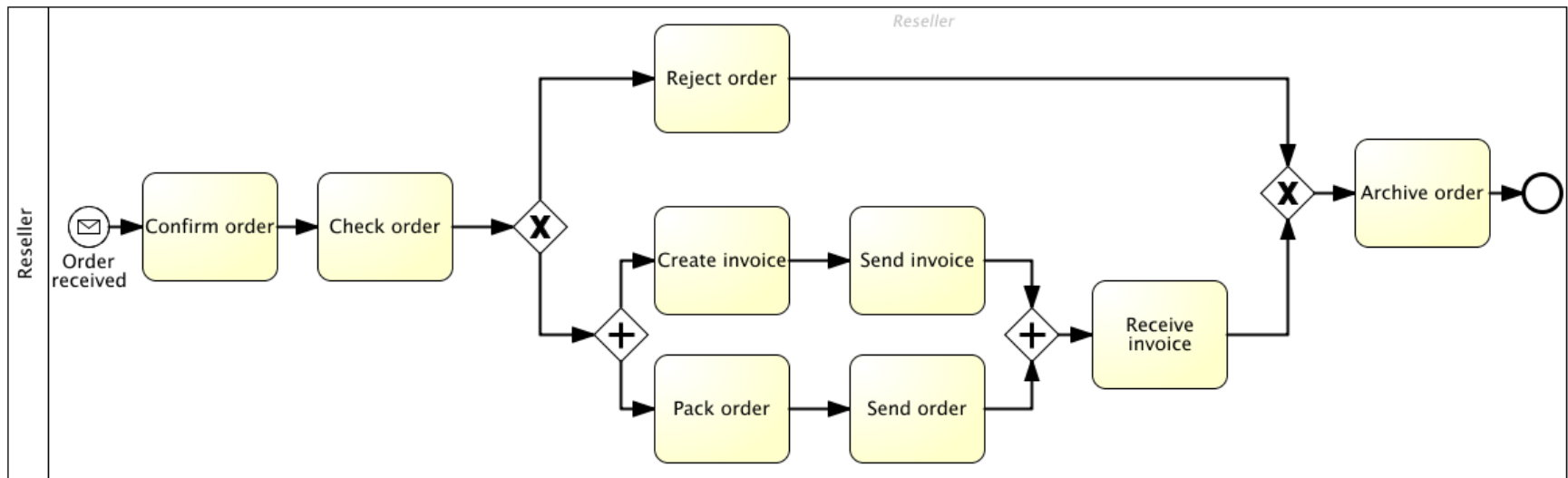
- Object lifecycle conformance is fulfilled
 - All state transitions can be also found in the object life cycle of the data object Order



Example, Object Lifecycle 3



- Object lifecycle conformance is fulfilled
 - The process requires only allowed state transitions.
 - Other processes may require the additional behavior of the data object



Usage Scenarios

- With the help of object lifecycle conformance, modeling errors and inconsistencies can be detected
 - In the process model: Missing/wrong activities or wrong sequence of activities
 - For the data object: Errors in the object lifecycle
- Extension of operational business processes for developing implemented process models

Video Clip 5.5

Object Lifecycle Conformance

- Processes and data object lifecycle must be consistent
- Object lifecycle conformance requires that no new states or state transitions are introduced by the process model
- However, not all states (and state transitions) need to be visited by business processes.

Summary Week 5

- Organizing process models
 - Input and output between process models
- Data objects, states and data flow
 - Data objects, states, collections, and persistent data stores
- Data execution semantics
 - Data objects change process execution semantics
- Multiple instances activities
 - Data object collections can be processed by MI activities
- Object Lifecycle Conformance
 - Make sure processes and data object states are in line