UNICAM
Università di Camerino
1336

BPMN 2.0

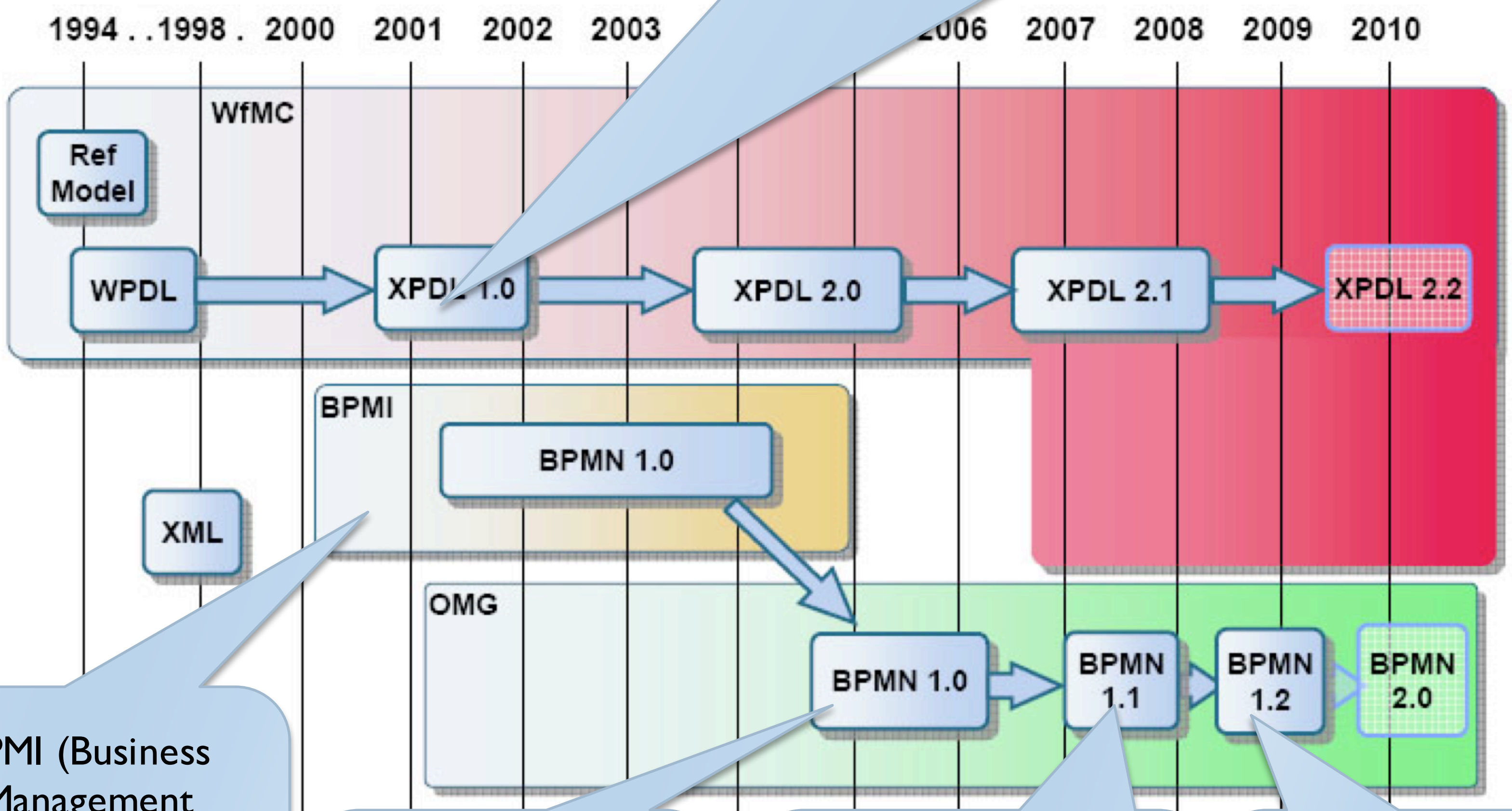
Business Process Management and Flexibility
Barbara Re, Phd

BPMN

- Business Process ~~Modelling~~ Model and Notation = BPMN
- The primary goal of BPMN is to provide a notation that is readily understandable by all business users:
  - the business analysts that create the initial drafts of the process,
  - the technical developers responsible for implementing the technology that will perform those process
  - the business people who will manage and monitor those processes.
- Thus, BPMN creates a standardized bridge for the gap between the business process design and process implementation
- We focus on it since it is the de facto standard notation to model BPs in particular within industrial contexts

History

The vendors realized immediately there was a need of a graphical representation for the language oriented towards the needs of business users. Not a notation that directly represents the precise execution language under development.



In 2001 BPMI (Business Process Management Initiative) developed BPML as an XML process execution language.

In 2004 **BPMN 1.0** was released to the public and **in 2006 it was adopted as OMG standard.**

In Feb.2008 **BPMN 1.1** was released to the public, **making the meaning of the notation more explicit.**

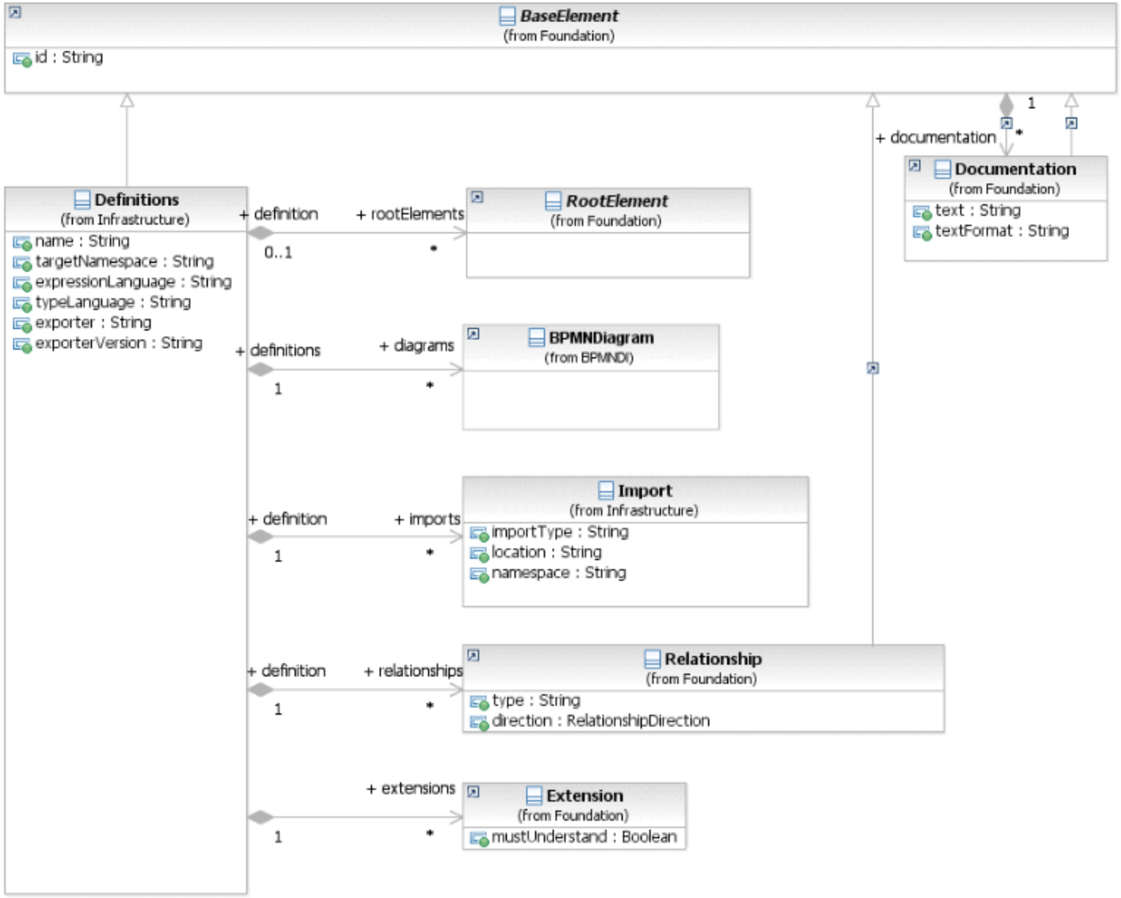
BPMN 1.2 does not include any significant graphical changes; **modifications were merely editorial.**

BPMN 1.2 vs BPMN 2.0

- **BPMN 1.2** provides a mapping from a "valid" BPMN diagram to BPEL, such that an engine can execute the process.
  - The 1.2 specification provides only **contained verbal descriptions** of the graphic notations elements and modeling rules. This leads to misleading and confusions in the translation process.
- **BPMN 2.0 beta 2** was introduced in June 2010.
  - It represents the biggest revision of BPMN since its inception.
- BPMN 2.0 received a **formal definition** in the form of a **meta-model**, that is a precise definition of the constructs and rules needed for creating specific models.

BPMN 2.0 Meta-model

- Metamodelling provides a number of benefits :
  - It formalizes the definition c models and entities.
  - It formalizes the relationshi between elements.
  - It enables interoperability.
- The new version's specification document has got **comprehensive UML class diagram** that graphically show the feature of the different BPMN constructs and their relationships.

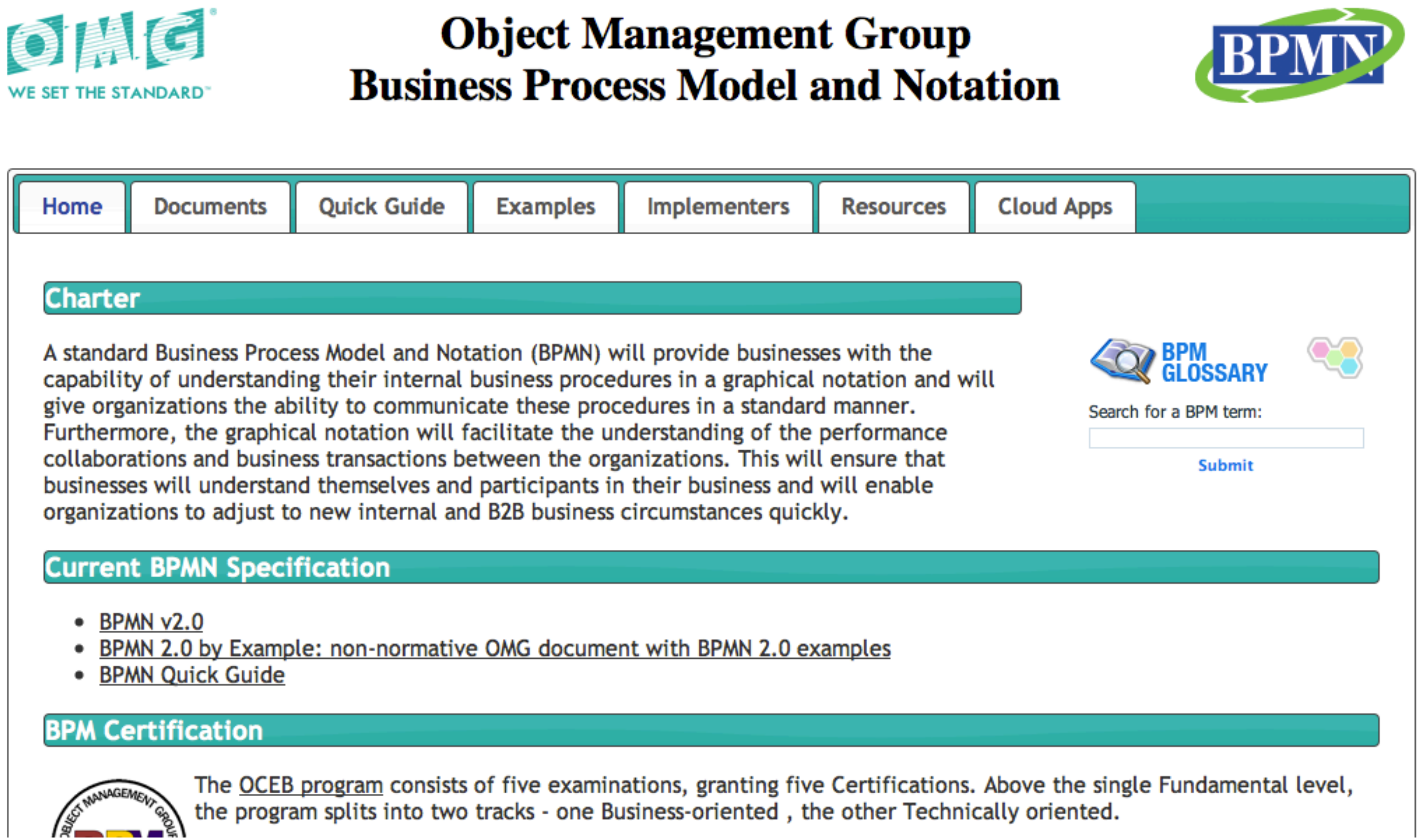


BPMN 1.2 vs BPMN 2.0

- The metamodel also has got additional language constructs that cannot be represented in the graphic models.
  - Such constructs are required by process engines to capture the necessary additional information for process execution.

- Moreover, the metamodel was the basis for the development on an **exchange format for BPMN models.**
  - Up to now, it was almost impossible to transfer BPMN models from one tool into another.
  - Some tools have got import and export interfaces for the exchange of BPMN models by means of the XPDL format, but the use of XPDL for this purpose is not widely accepted yet.
  - Moreover, XPDL has not been implemented uniformly by all vendors, so that in practice there are quite often problems with model exchange

BPMN 2.0 (http://www.bpmn.org/)



Object Management Group
Business Process Model and Notation

Charter

A standard Business Process Model and Notation (BPMN) will provide businesses with the capability of understanding their internal business procedures in a graphical notation and will give organizations the ability to communicate these procedures in a standard manner. Furthermore, the graphical notation will facilitate the understanding of the performance collaborations and business transactions between the organizations. This will ensure that businesses will understand themselves and participants in their business and will enable organizations to adjust to new internal and B2B business circumstances quickly.

Search for a BPM term:

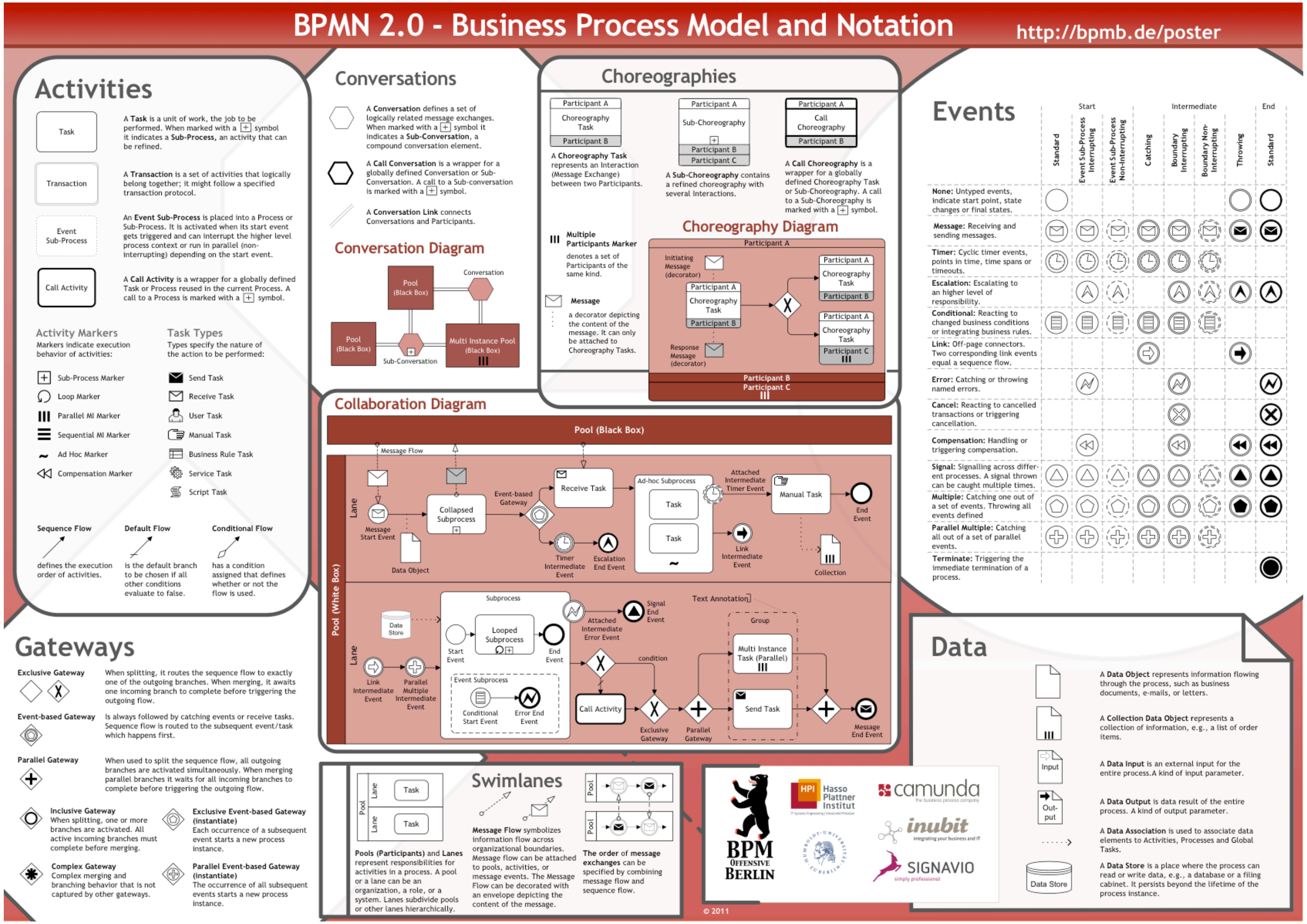
Current BPMN Specification

- BPMN v2.0
- BPMN 2.0 by Example: non-normative OMG document with BPMN 2.0 examples
- BPMN Quick Guide

BPM Certification

The OCEB program consists of five examinations, granting five Certifications. Above the single Fundamental level, the program splits into two tracks - one Business-oriented , the other Technically oriented.

BPMN 2.0 manifesto



BPMN 2.0 - Business Process Model and Notation

http://bpmb.de/poster

Activities

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.

A **Transaction** is a set of activities that logically belong together; it might follow a specified transaction protocol.

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.

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
- III 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.

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.

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

Exclusive Event-based Gateway (instantiate)
Each occurrence of a subsequent event starts a new process instance.

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

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.

III **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

Collaboration Diagram

Events

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

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.

BPM OFFENSIVE BERLIN

HPI Hasso Plattner Institut

camunda the business process company

inubit integrating your business and IT

SIGNAVIO simply professional

© 2011

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.

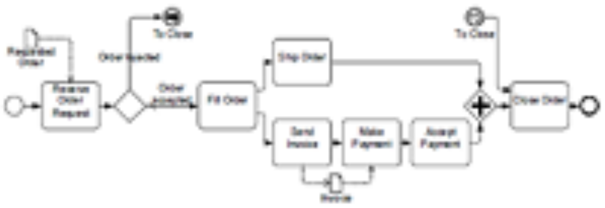
A **Data Store** is a place where the process can read or write data, e.g., a database or a filing cabinet. It persists beyond the lifetime of the process instance.

BPMN 2.0 - models

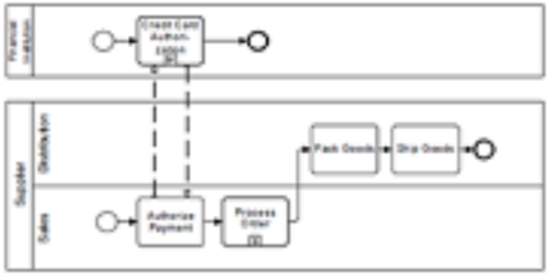
- **Processes**, including:
  - Private (internal) **Business Processes**
  - Public **Processes**
- **Choreographies**
- **Collaborations**, which can include **Processes** and/or **Choreographies**
  - A view of **Conversations**

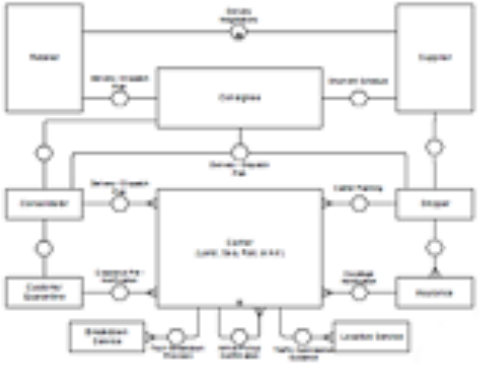
BPMN 2.0 model types

Process

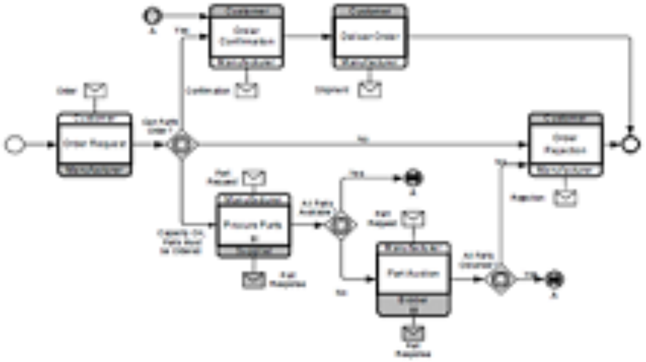


Collaboration





Conversation

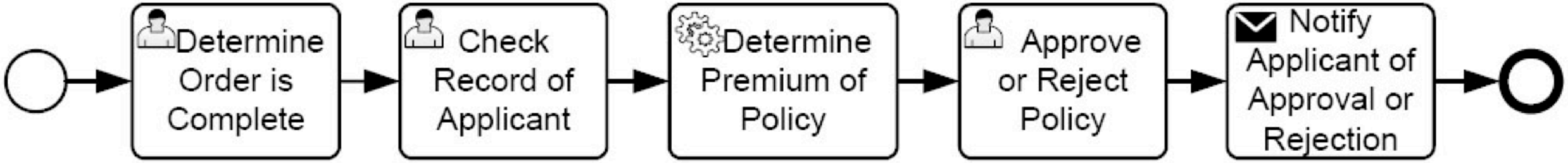


Choreography

Process: Private Business Process (executable or not)

Private **Business Processes** are those internal to a specific organization.

An Example:


Determine Order is Complete
Check Record of Applicant
Determine Premium of Policy
Approve or Reject Policy
Notify Applicant of Approval or Rejection

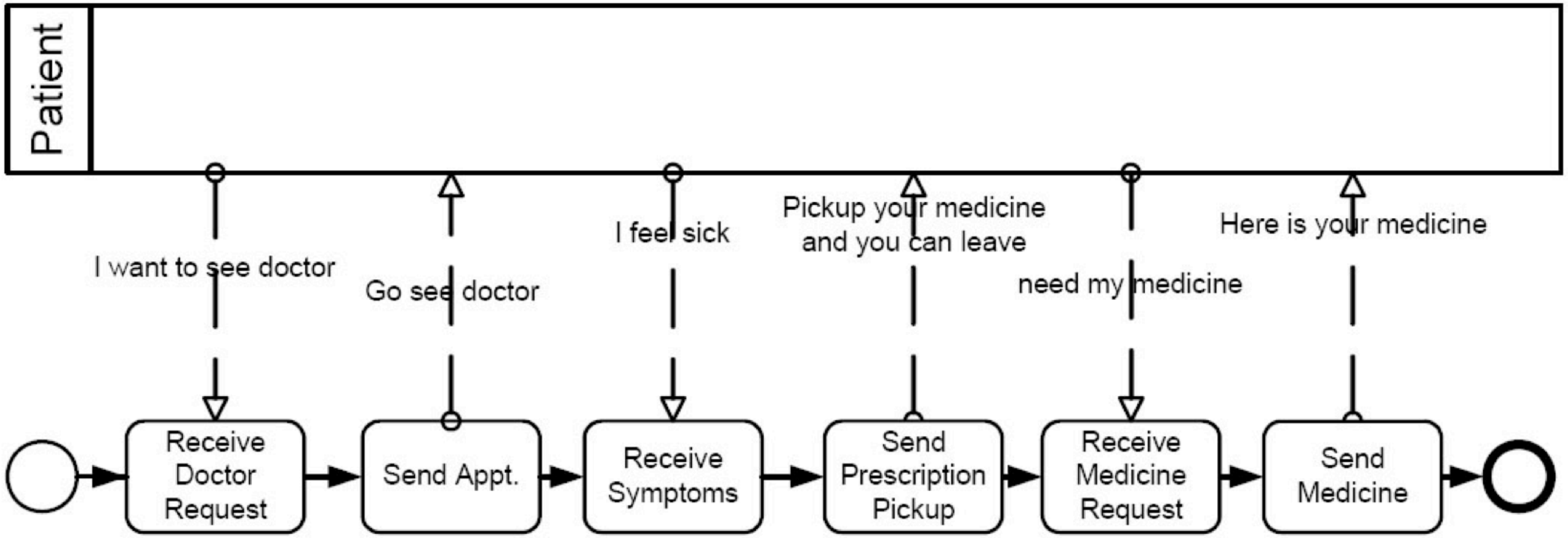
- An *executable* **Process** is a **Process** that has been modeled for the purpose of being executed according to the semantics

- A non-executable **Process** is a *private* **Process** that has been modeled for the purpose of documenting **Process** behavior at a modeler-defined level of detail. Thus, information needed for execution, such as formal condition Expressions are typically not included in a *non-executable* **Process**.

Process: Public Business Process

A *public* **Process** represents the interactions between a *private* **Business Process** and another **Process** or *Participant*

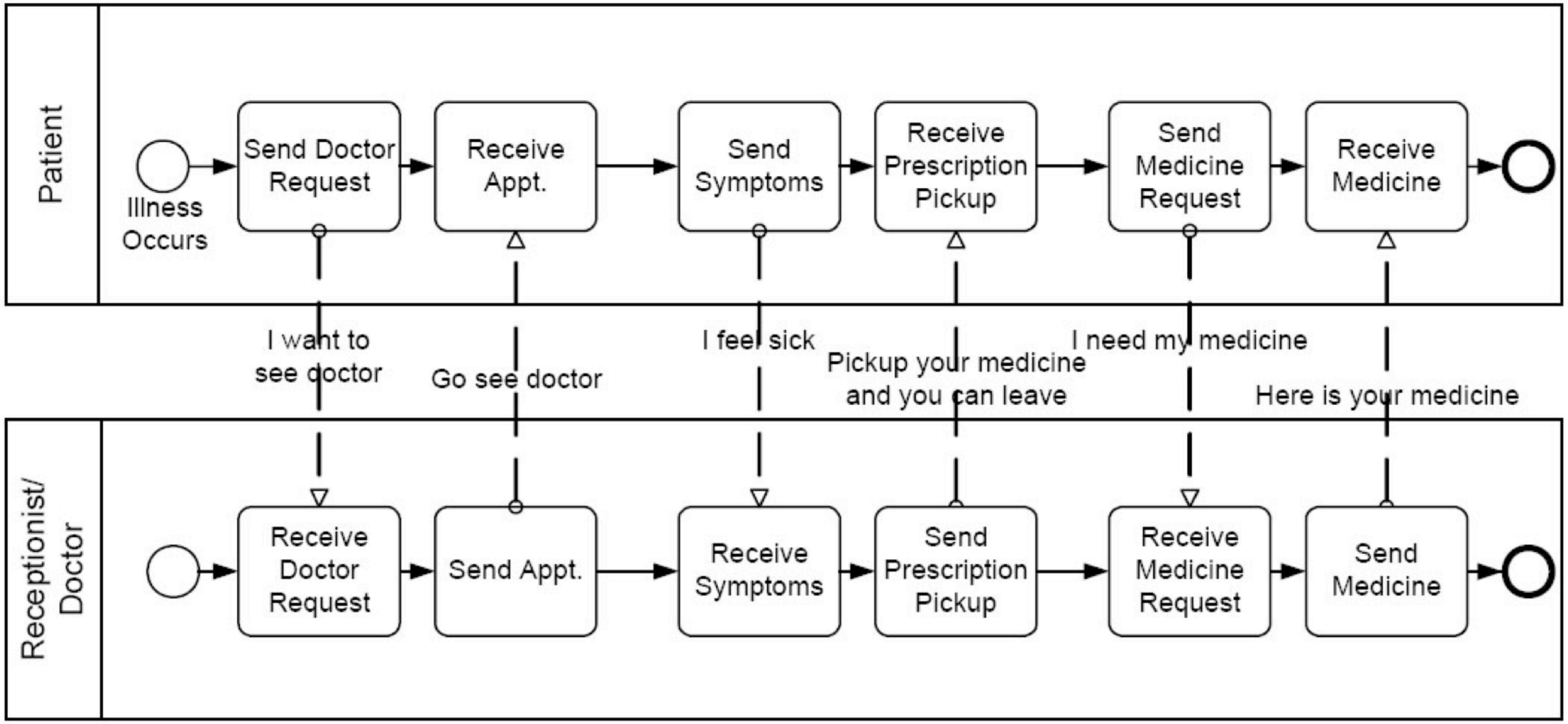
An Example:



Collaboration Diagram

A **Collaboration** depicts the interactions between two or more business entities

An Example:

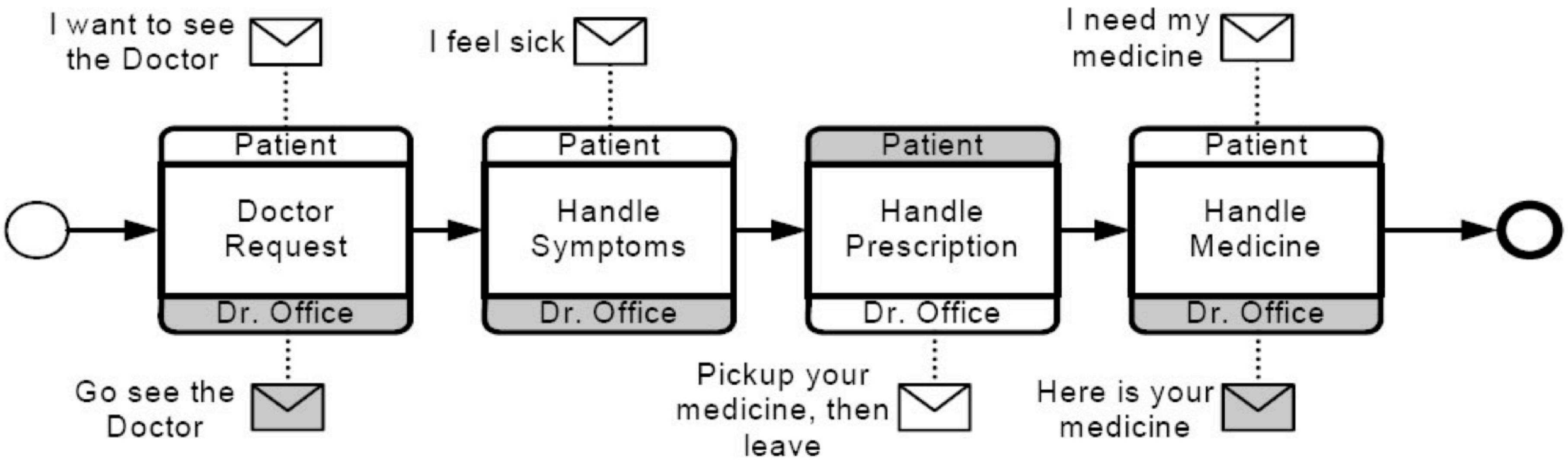


Choreography

- A self-contained **Choreography** is a definition of the expected behavior, basically a procedural contract, between interacting *Participants*.

- While a normal **Process** exists within a **Pool**, a **Choreography** exists between **Pools** (or *Participants*).

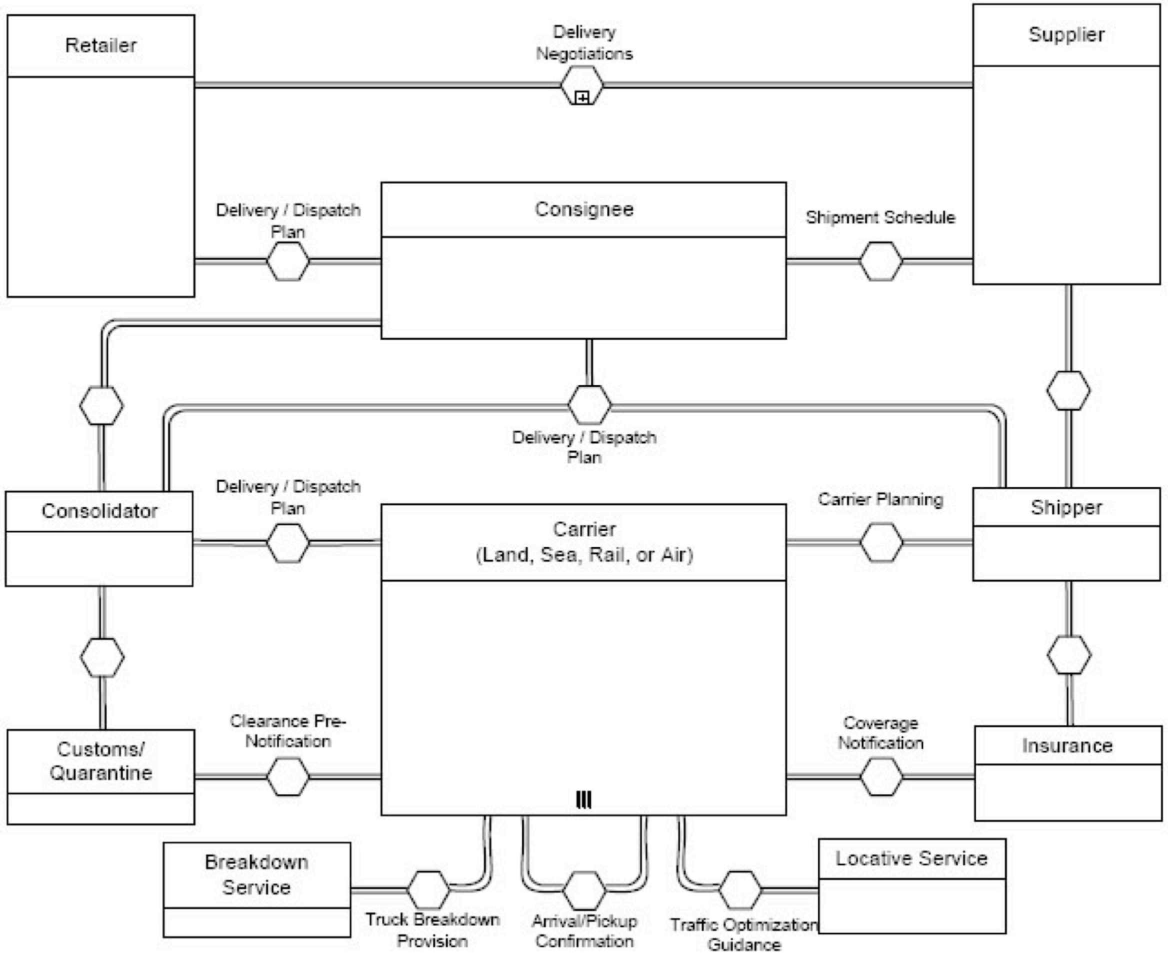
An Example:



Conversation Diagram

- The **Conversation** diagram is a particular usage of and an informal description of a **Collaboration** diagram.

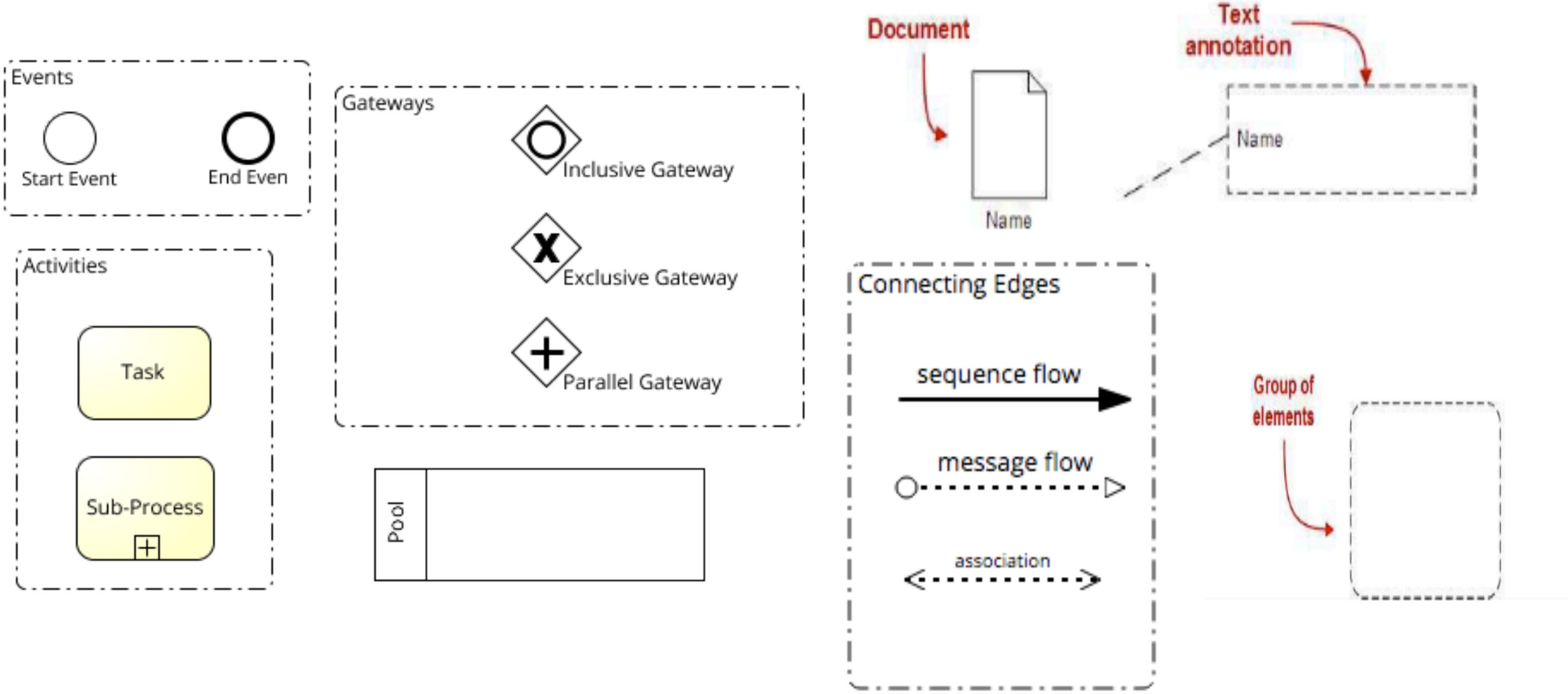
- A **Conversation** is the logical relation of **Message** exchanges.



Process / Collaboration Diagram

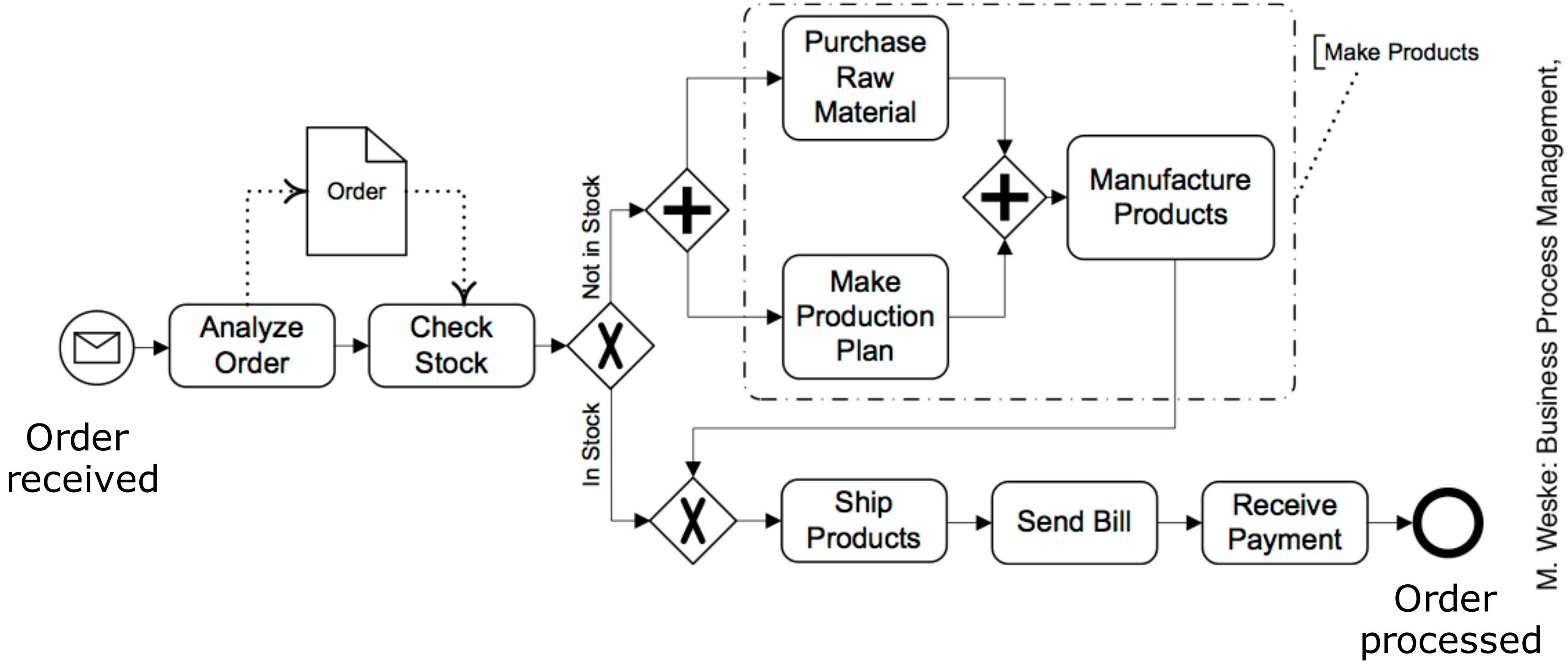
Business Process Management and Flexibility
Barbara Re, Phd

Basic Elements



Used for simple business process modelling, understood by majority of business process related stakeholders

Simple BP model in BPMN

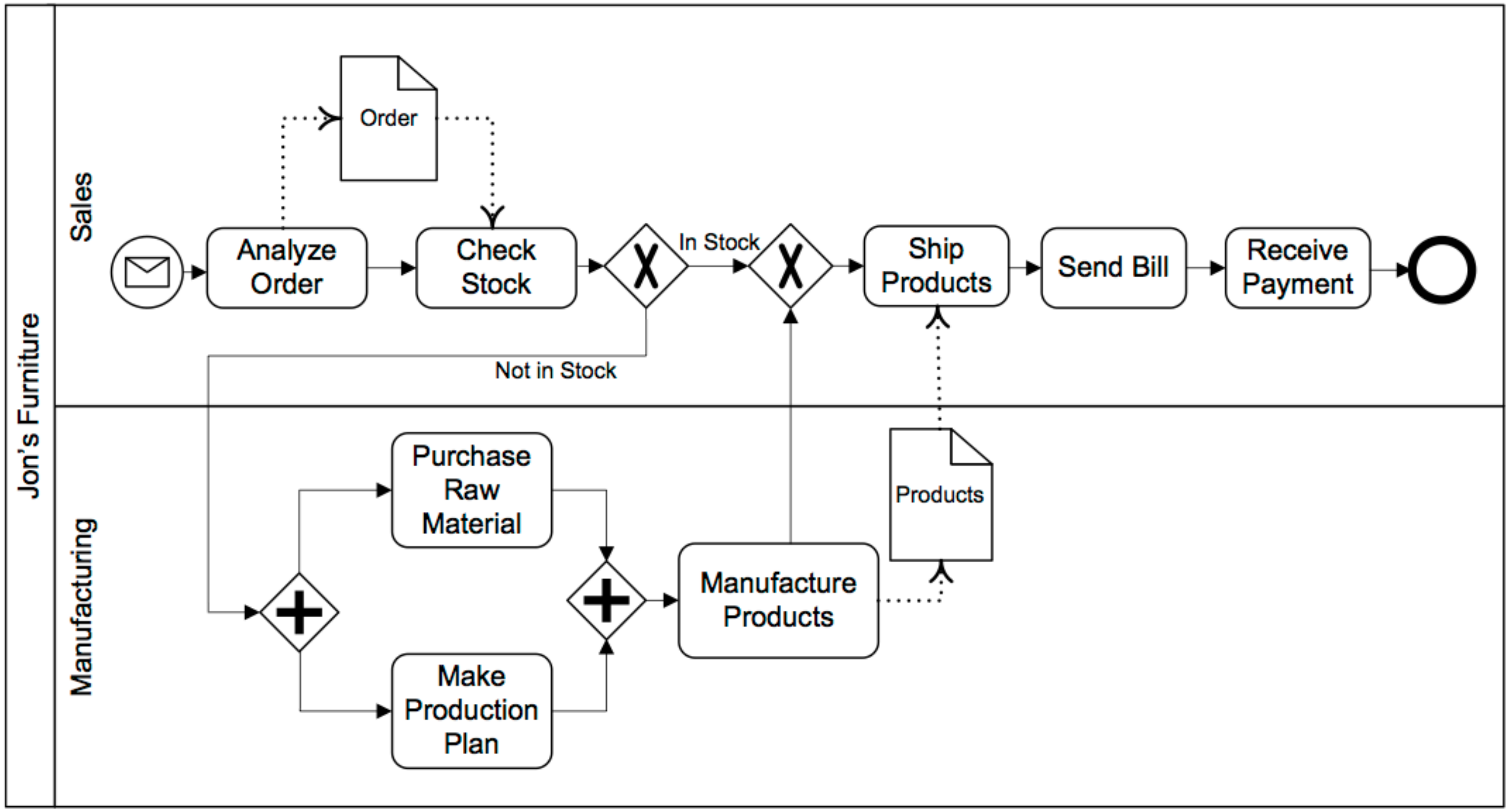


M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Resources

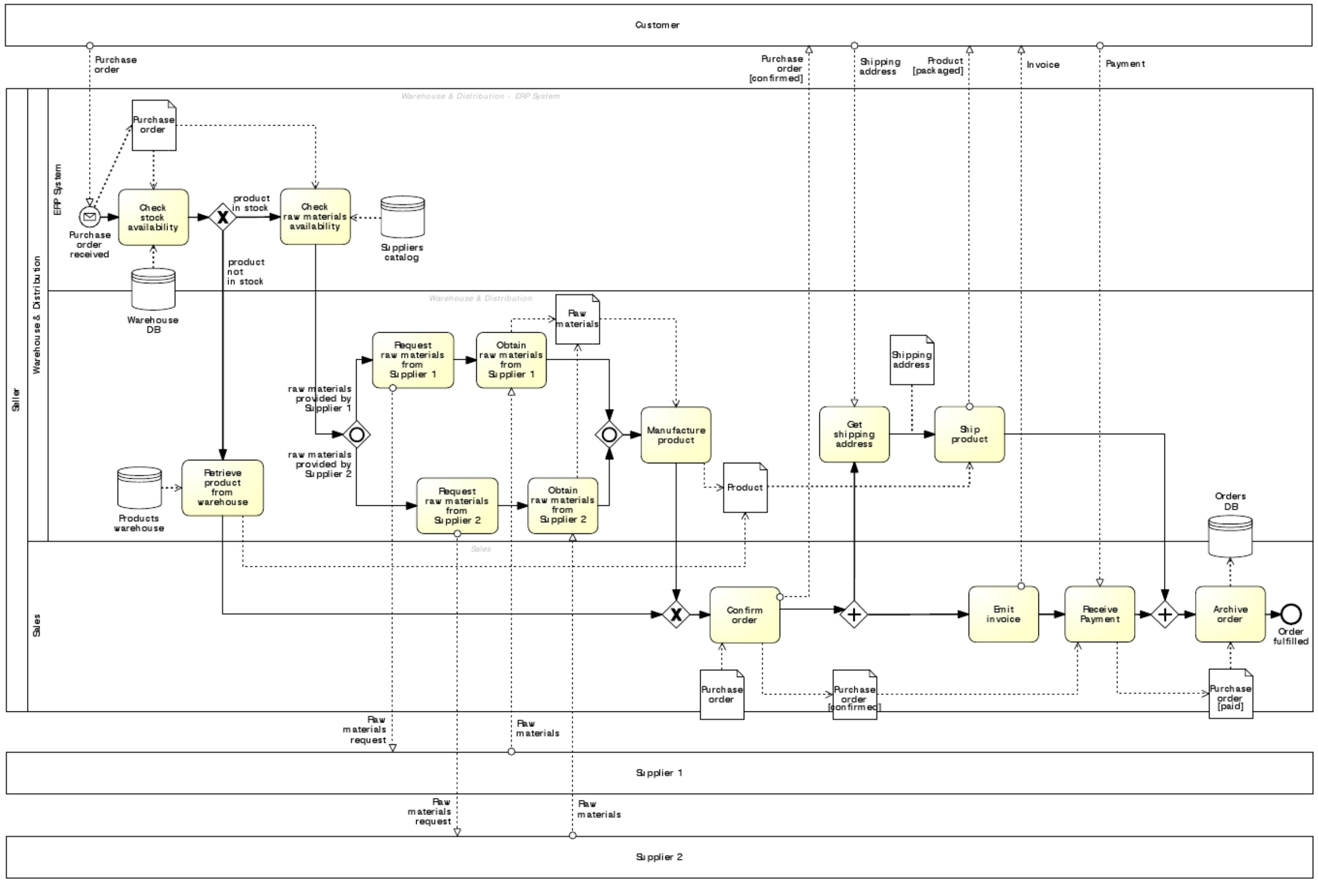
- BPMN provides two constructs to model resource aspects:
  - **Pools** are generally used to model resource classes
  - **Lanes** are used to partition a pool into sub-classes or single resources. Lanes can be nested within each other in multiple levels.
- There are no constraints as to what specific resource type a **pool** or a **lane** should model
  - We would typically use:
    - A pool to model a *business party* like a whole organization
    - A lane to model a department, unit, team or software system/equipment within that organization

Simple BP model with resources information



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

Message Flow

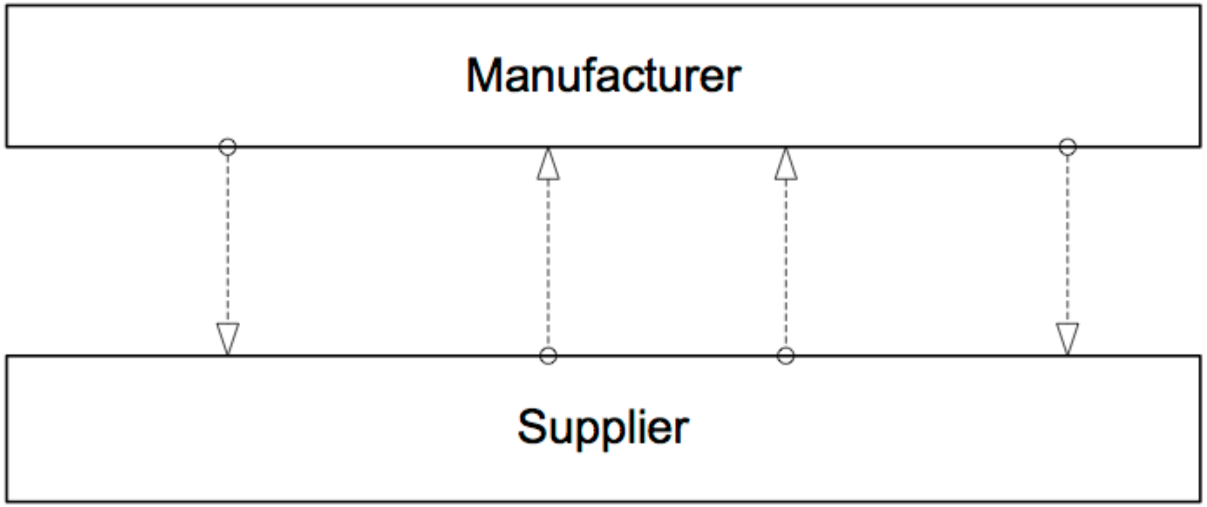


- **A message flow** represents the flow of information between two separate resource classes (pools)

Black box or white box?

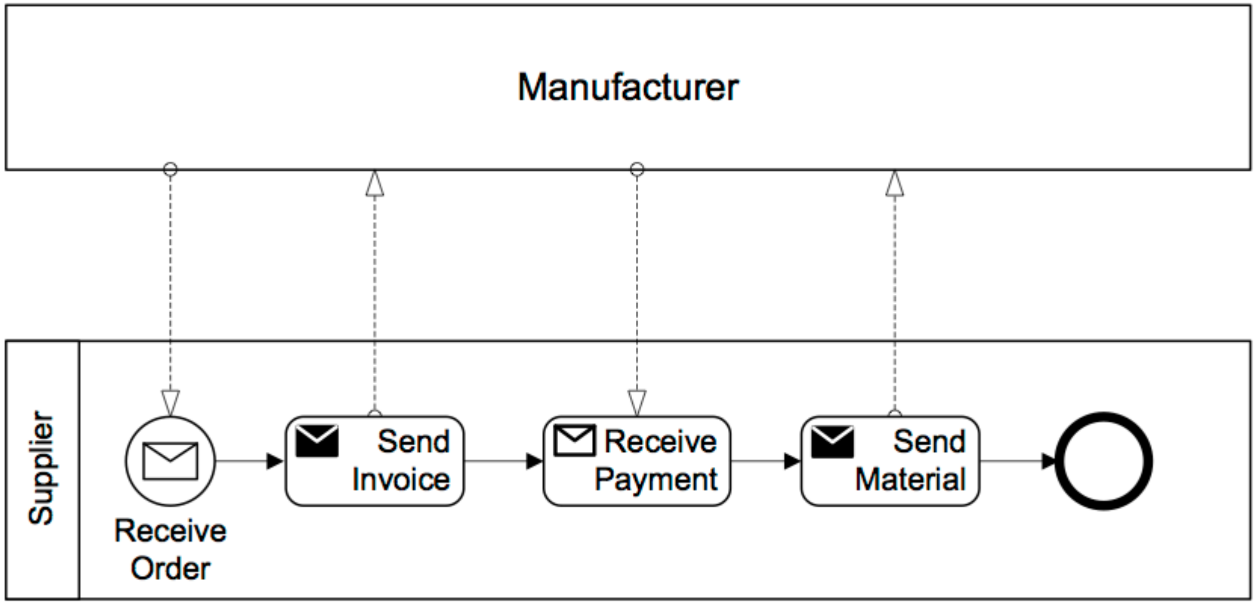
- Modeling a pool as a white box or as a black box is a matter of relevance
- When working on a collaboration diagram, an organization may decide whether or not to expose their internal behavior depending on the requirements of the project at hand

BP collaboration – message flow



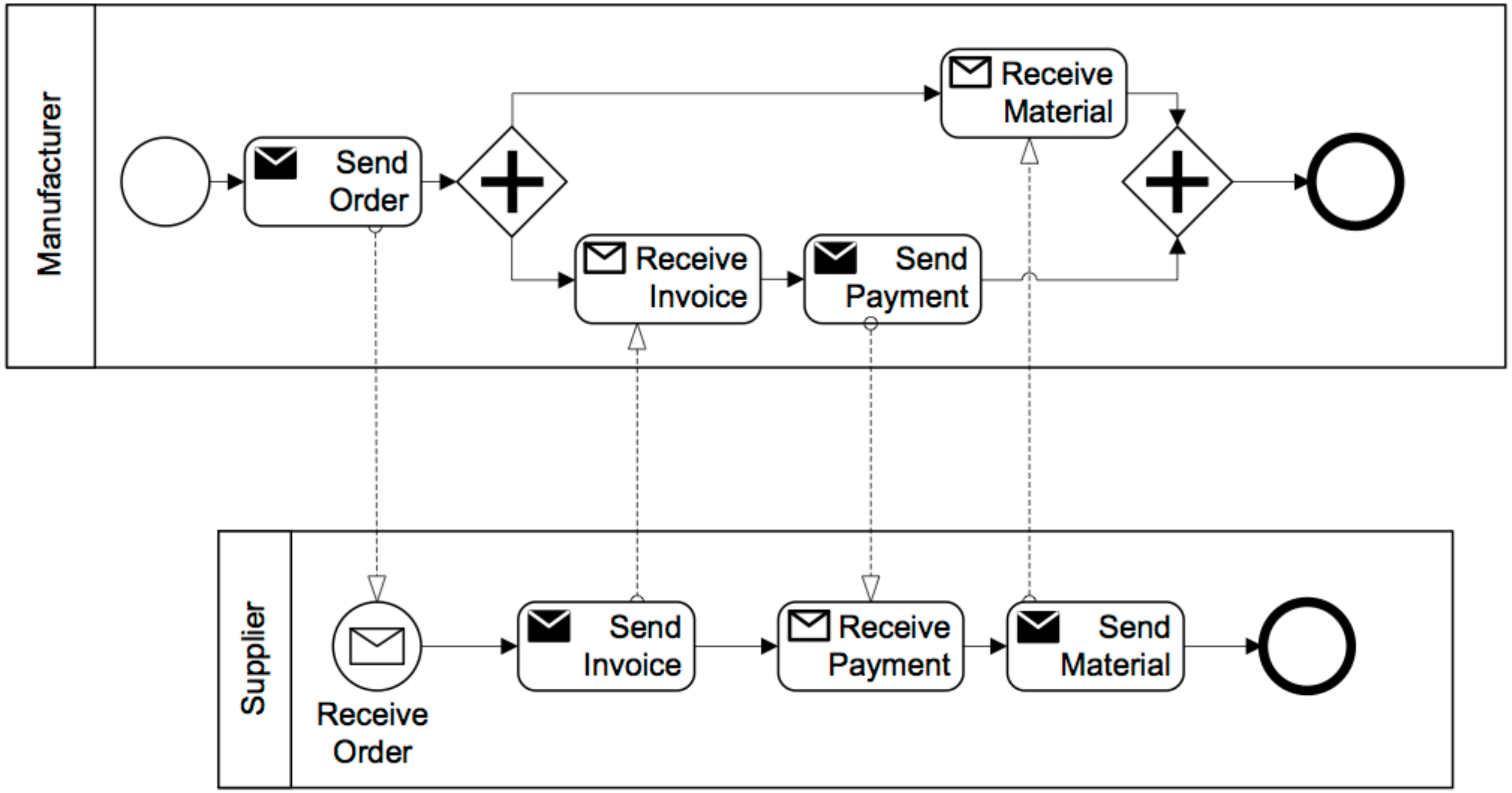
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

BP collaboration with one public process (supplier)



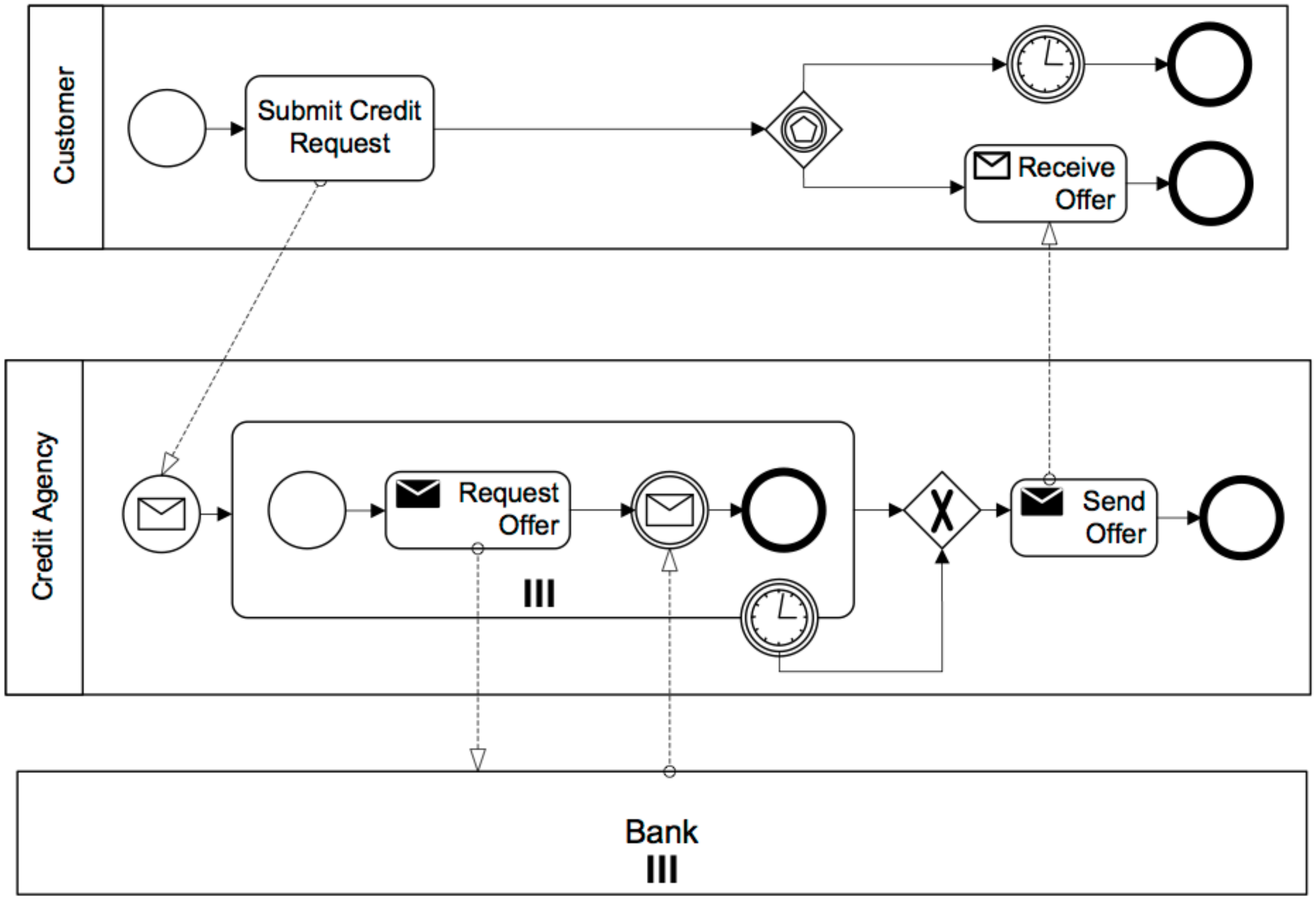
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

BP collaborating with two public process



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

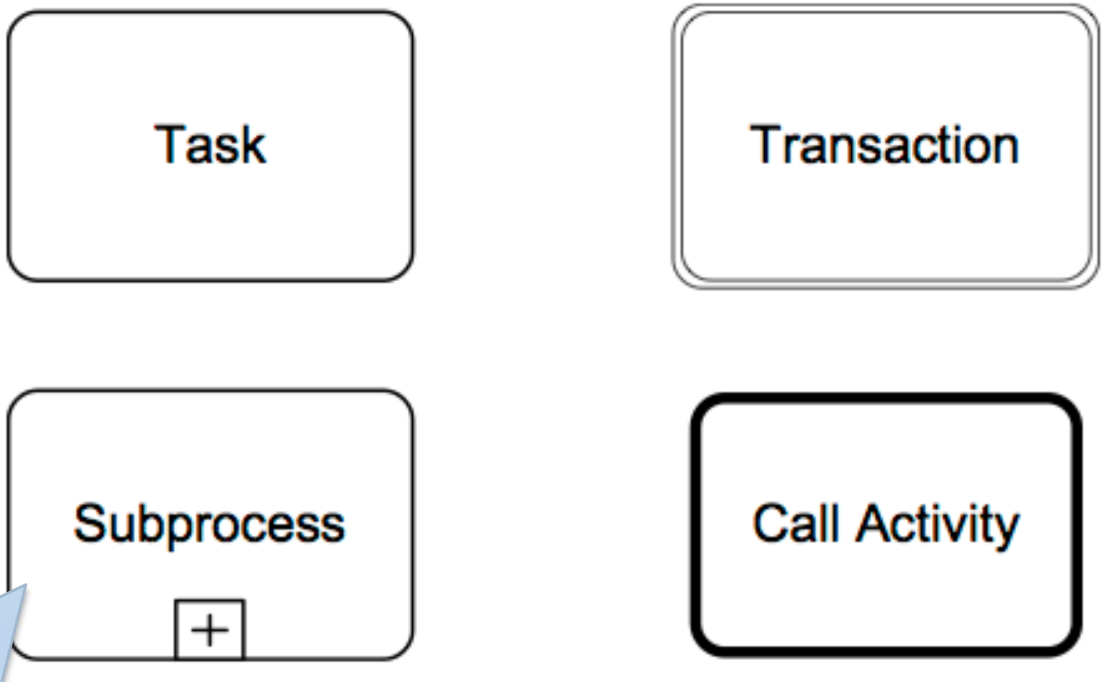
Multiple instance pool



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

Activity types in the BPMN

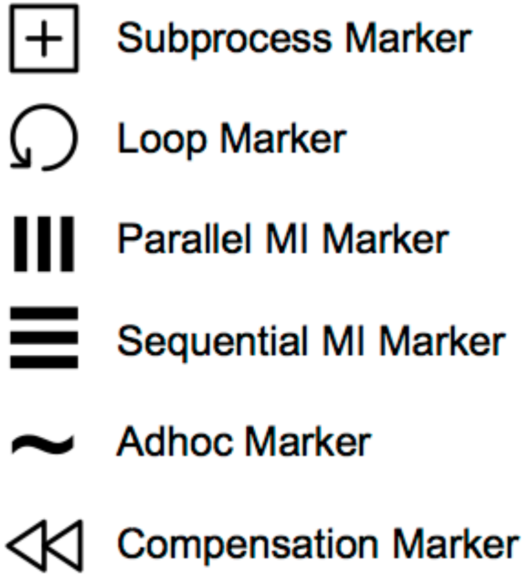
An atomic activity, also called *task*, is an activity capturing a unit of work that cannot be further broken down



A sub-process represents a self-contained, composite activity that can be broken down into smaller units of work

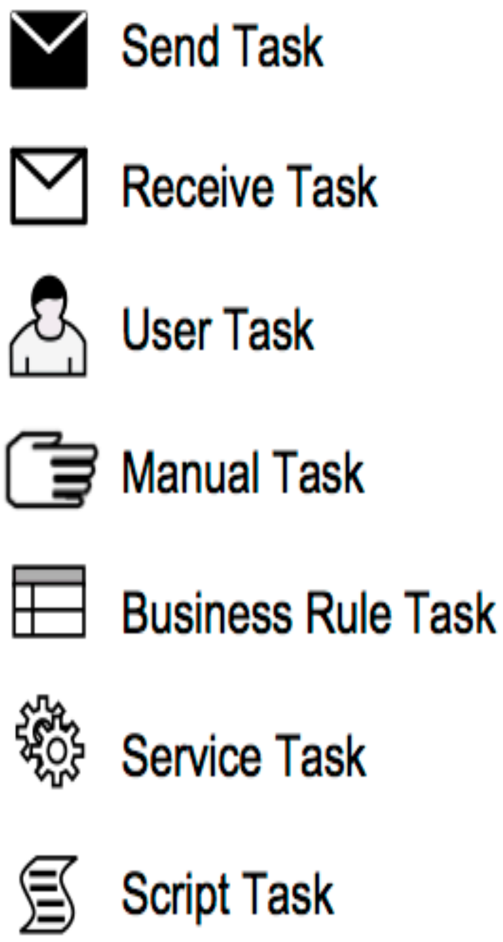
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Activity Markers



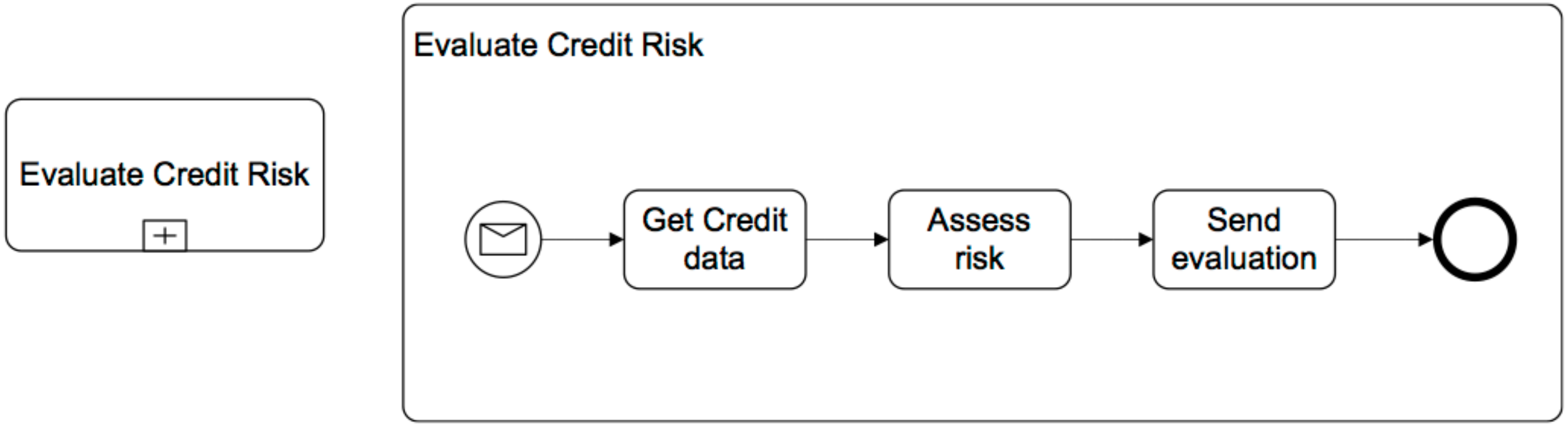
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Task types



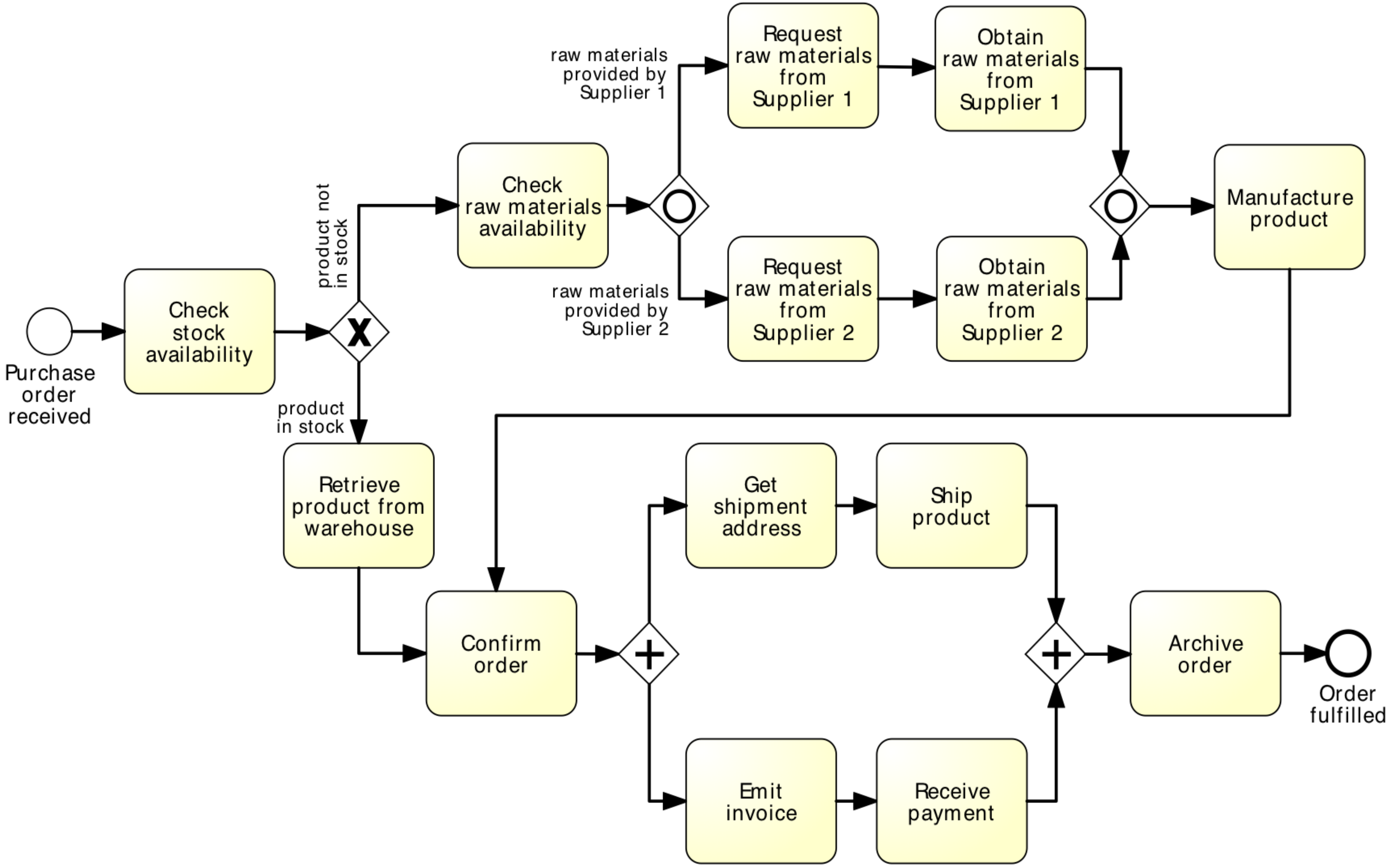
- A send task is a task that send a message. Once the message is sent, the task is completes
- A receive task is a task that waits to receive a message. Once the message arrives, the task completes
- User tasks represent traditional workflow tasks that involve user interaction
- Manual activities are performed without the support of software systems.
- A Business Rule task is used to synchronously execute one or more rules.
- A service task is a task that is implemented by a piece of software, either using a Web services interface or an application programming interface to a software system.
- A script task is a task that uses some scripting language expression in order to be performed. Script tasks are used to represent simple functionality, for which no dedicated software system is required.

Collapsed and expanded sub-process

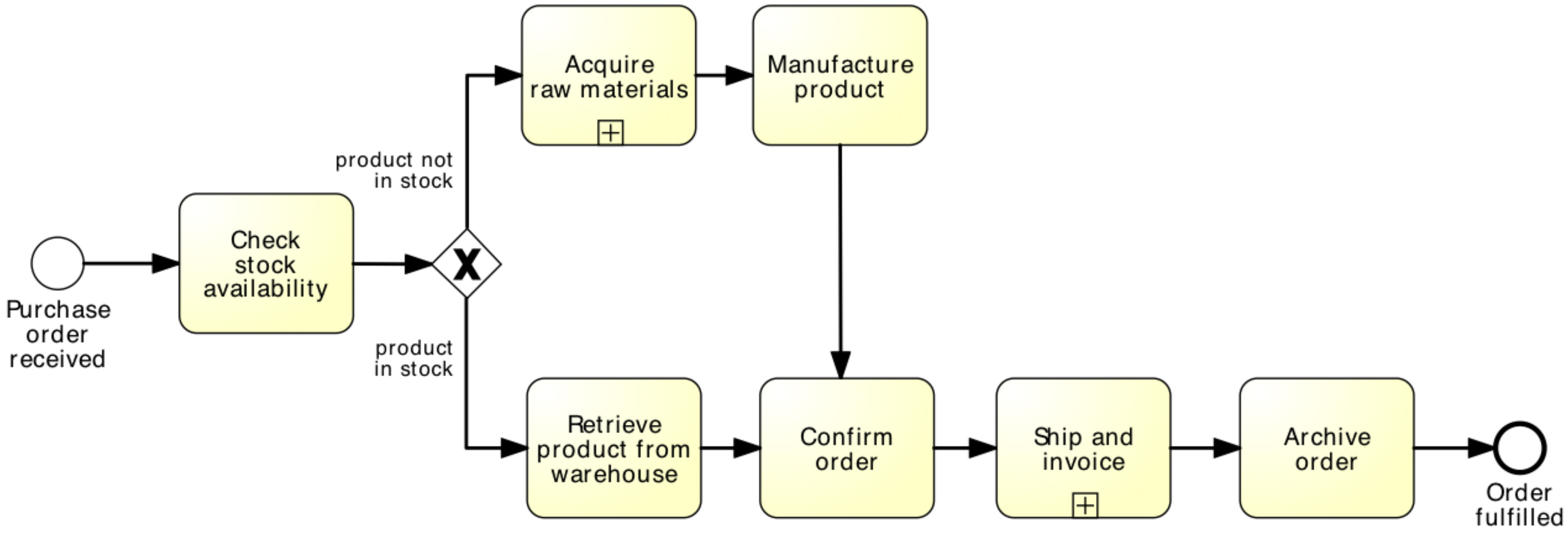


M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

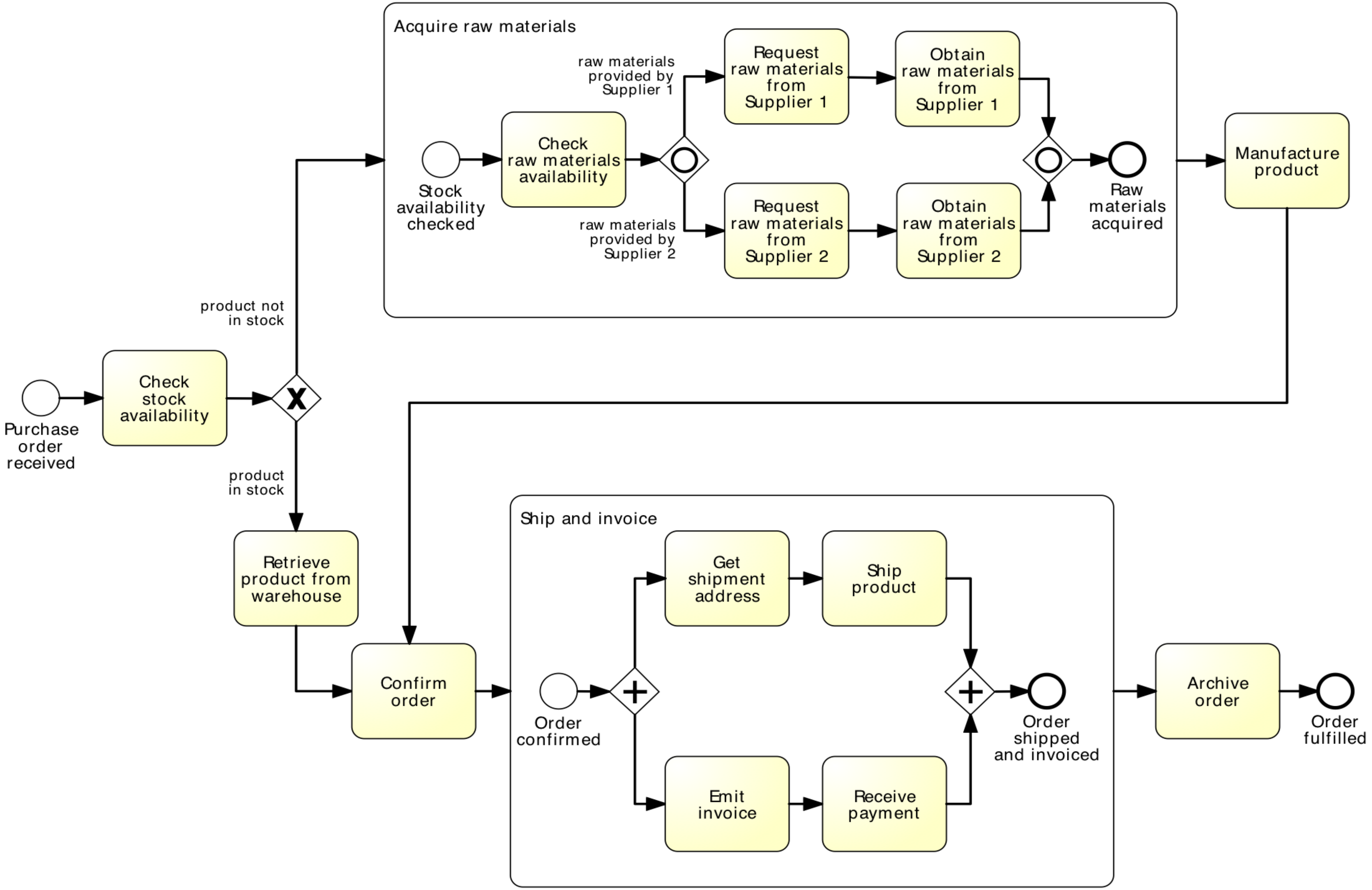
Anything wrong with this model?


Purchase order received
Check stock availability
product not in stock
product in stock
Check raw materials availability
raw materials provided by Supplier 1
raw materials provided by Supplier 2
Request raw materials from Supplier 1
Obtain raw materials from Supplier 1
Request raw materials from Supplier 2
Obtain raw materials from Supplier 2
Manufacture product
Retrieve product from warehouse
Confirm order
Get shipment address
Ship product
Emit invoice
Receive payment
Archive order
Order fulfilled

Is this better?



Expanded…


Acquire raw materials
Stock availability checked
Check raw materials availability
raw materials provided by Supplier 1
Request raw materials from Supplier 1
Obtain raw materials from Supplier 1
raw materials provided by Supplier 2
Request raw materials from Supplier 2
Obtain raw materials from Supplier 2
Raw materials acquired
Manufacture product
Purchase order received
Check stock availability
product not in stock
product in stock
Retrieve product from warehouse
Confirm order
Ship and invoice
Order confirmed
Get shipment address
Ship product
Emit invoice
Receive payment
Order shipped and invoiced
Archive order
Order fulfilled

Use of Sub-processes

1. Break down large models into smaller ones, making them easier to understand and maintain

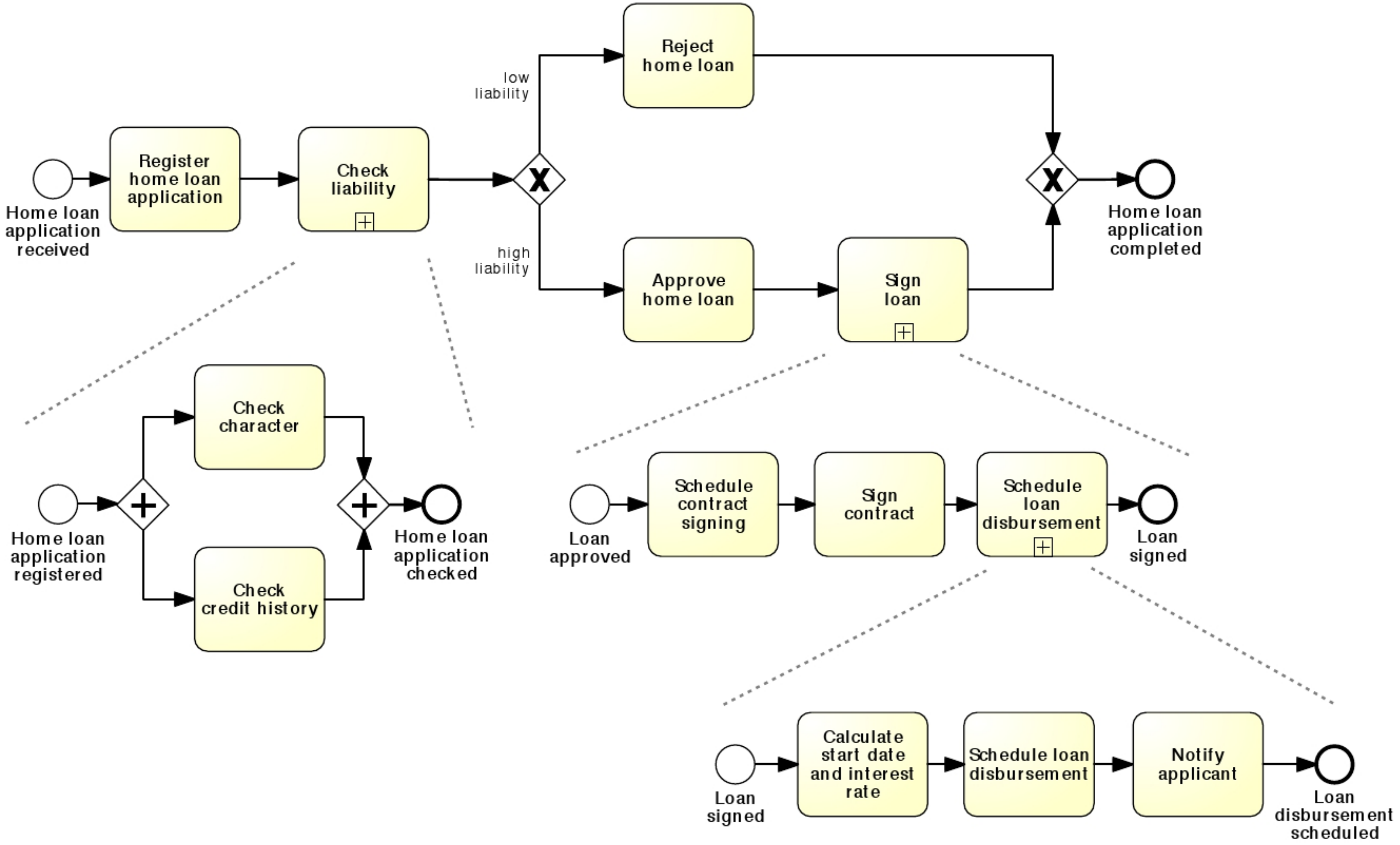
   → process hierarchies: we can nest sub-processes in multiple levels, so as to decompose a process model hierarchically

2. Share common fragments across multiple processes

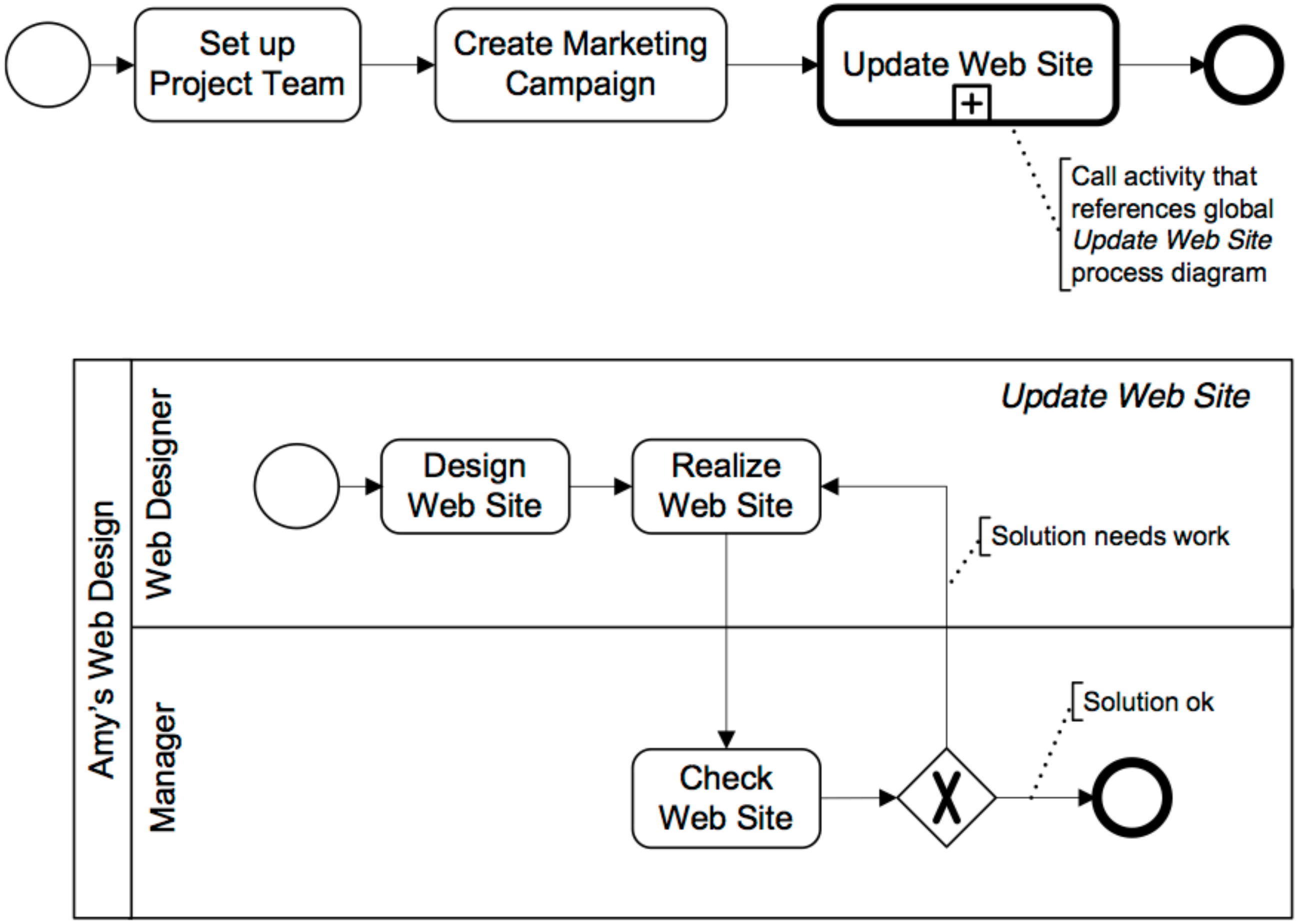
   → shared sub-processes

3. Identify parts of a process that should be repeated

Process hierarchies


Home loan application received
Register home loan application
Check liability
low liability
high liability
Reject home loan
Approve home loan
Sign loan
Home loan application completed
Home loan application registered
Check character
Check credit history
Home loan application checked
Loan approved
Schedule contract signing
Sign contract
Schedule loan disbursement
Loan signed
Loan signed
Calculate start date and interest rate
Schedule loan disbursement
Notify applicant
Loan disbursement scheduled

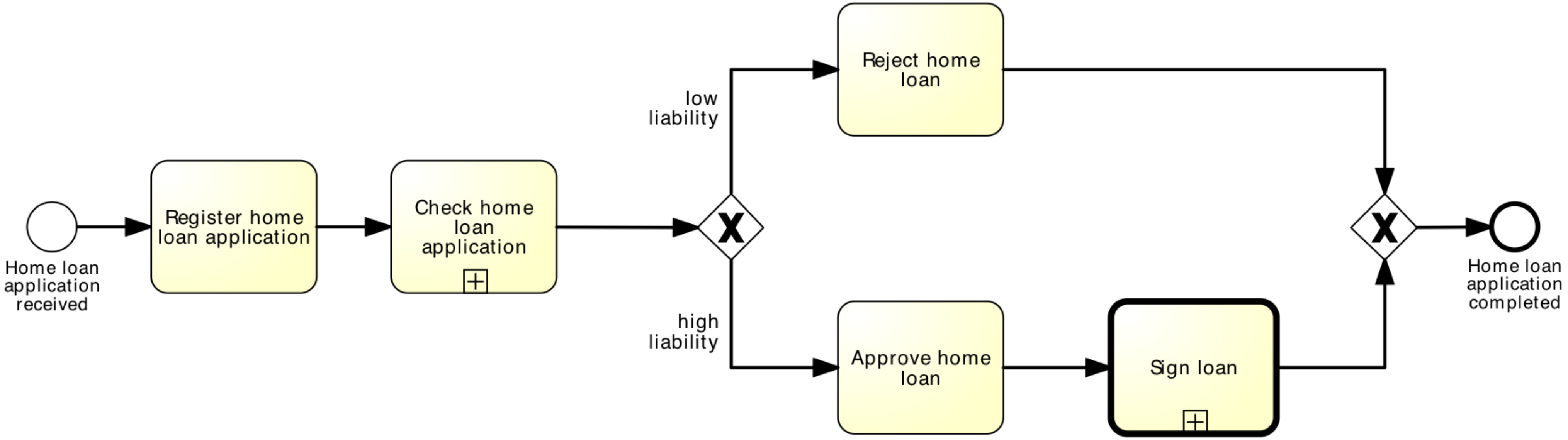
Share common fragments across multiple processes: Call Activity

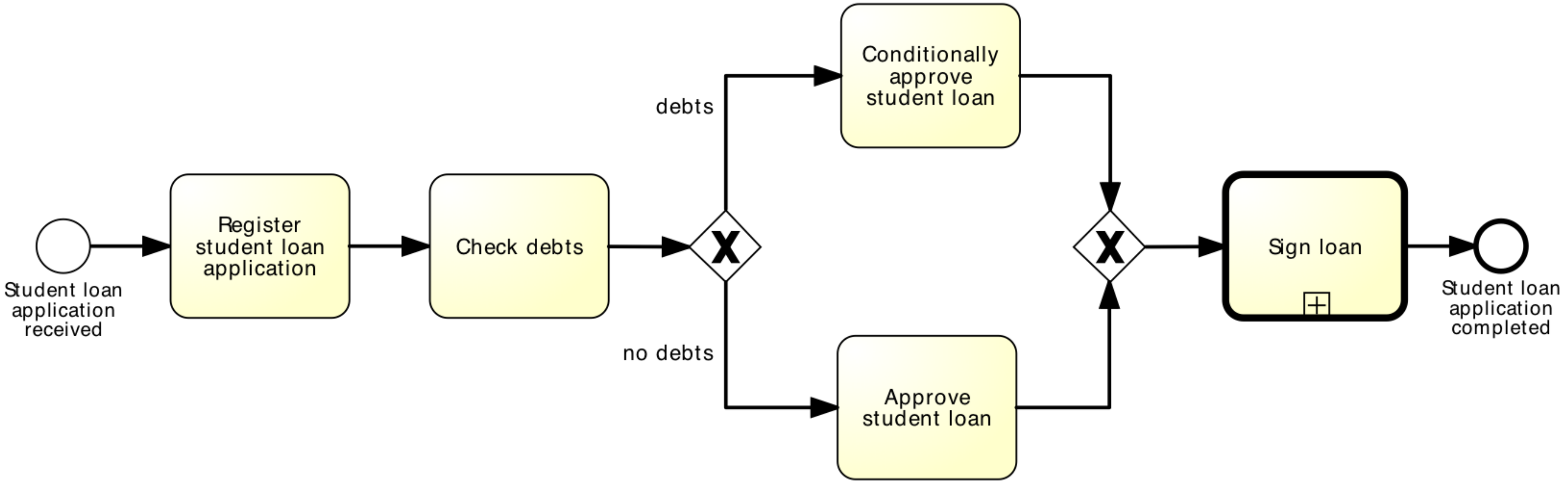


M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

The call activity refers to a process model that can be invoked by more than one process
In the same process model collection

Share common fragments across multiple processes: example

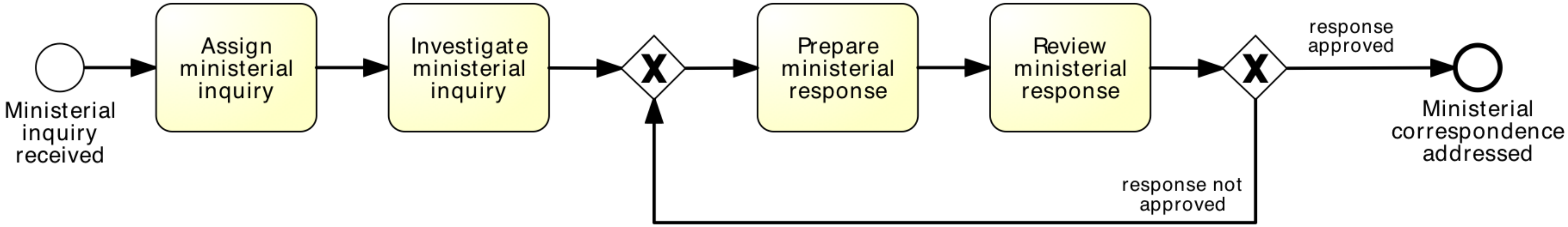

Home loan application received
Register home loan application
Check home loan application
low liability
high liability
Reject home loan
Approve home loan
Sign loan
Home loan application completed


Student loan application received
Register student loan application
Check debts
debts
no debts
Conditionally approve student loan
Approve student loan
Sign loan
Student loan application completed

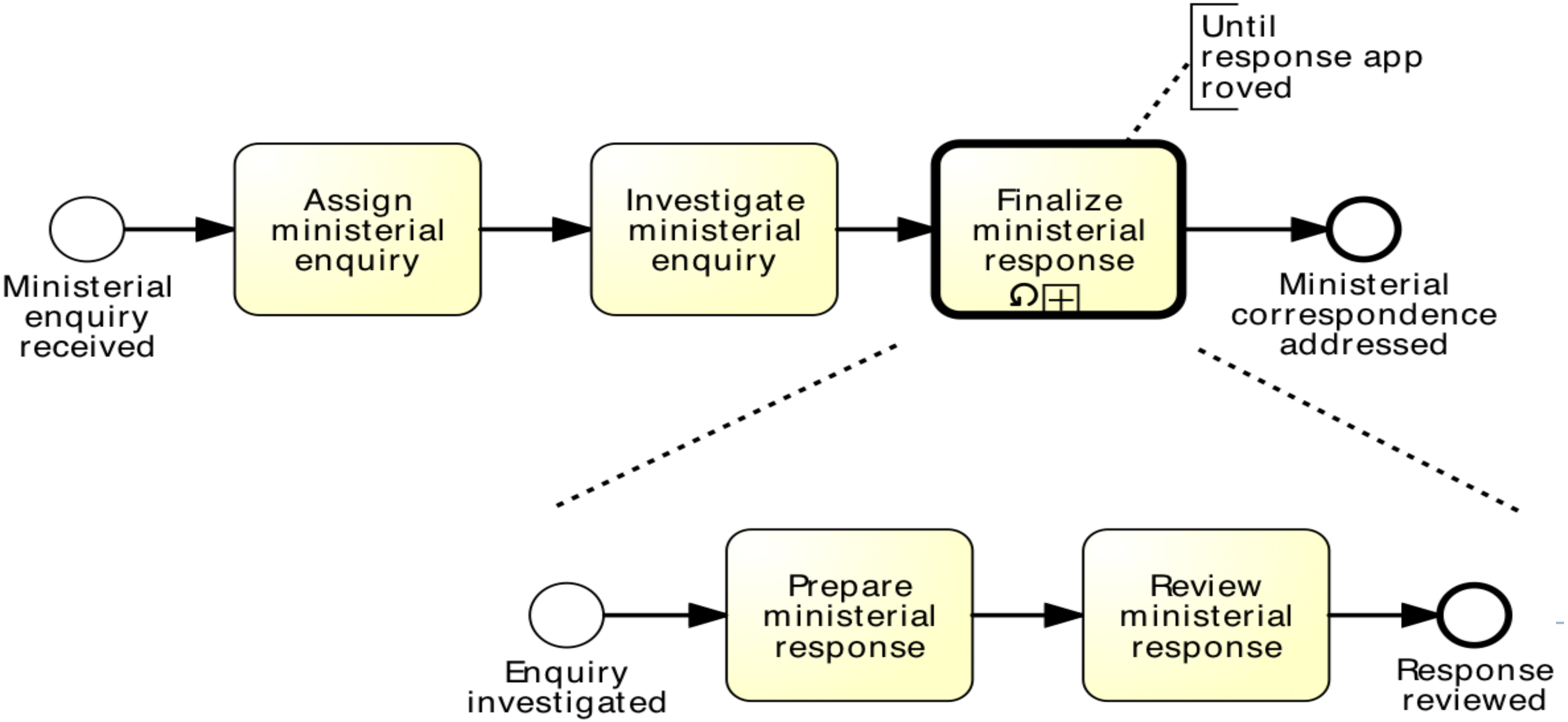
Call Activity: reusability and maintanance

- Our default choice should be to define sub-processes as global process models so as to **maximize their reusability** within our process model collection
  - Supporting processes such as payment, invoicing, HR, printing, are good candidates for being defined as global process models, since they are typically shared by various business processes within an organization
- Besides reusability, another advantage of using global process models is that **any change made to these models will be automatically propagated to all process models** that invoke them

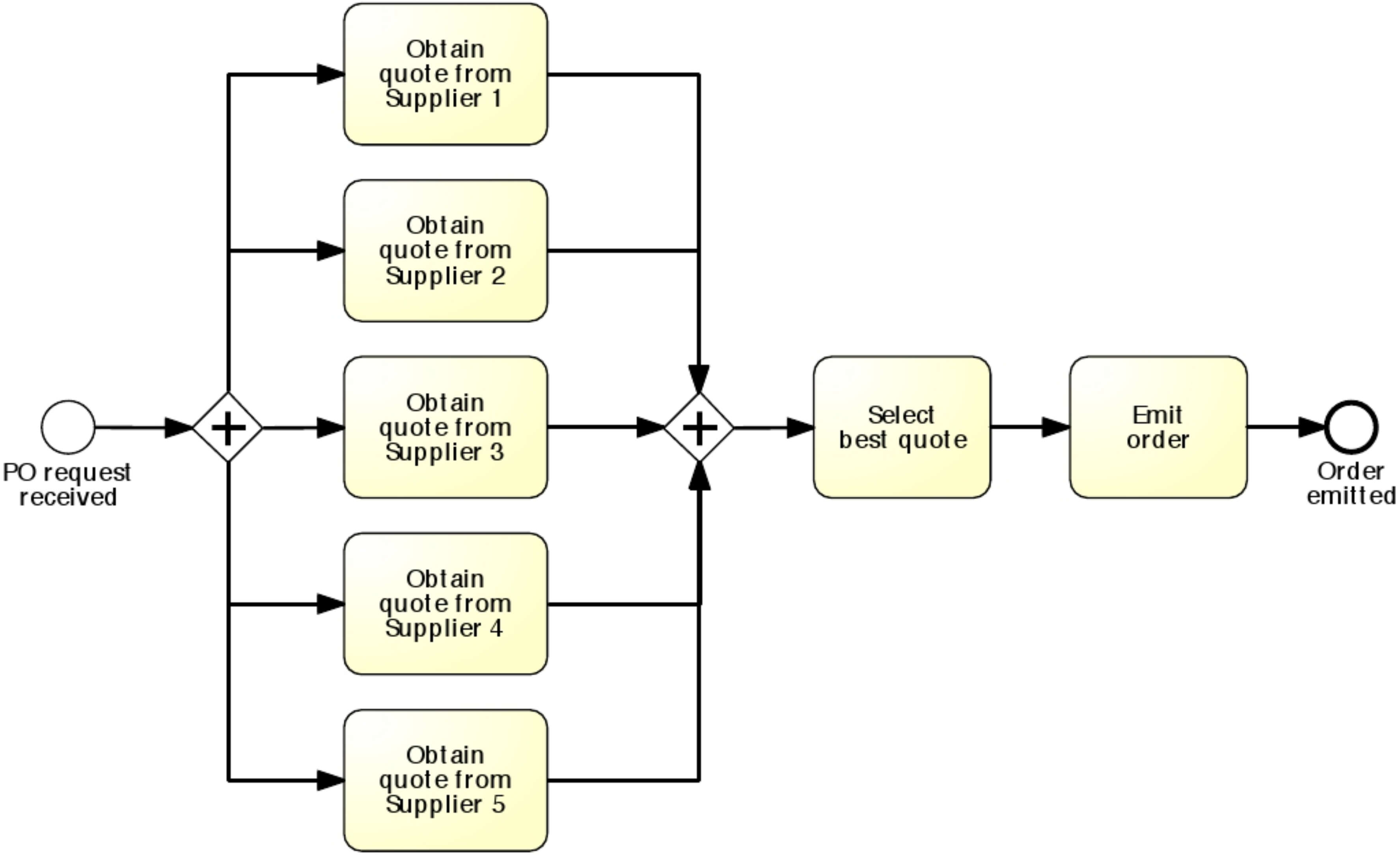
Identify parts of a process that should be repeated


Ministerial inquiry received
Assign ministerial inquiry
Investigate ministerial inquiry
Prepare ministerial response
Review ministerial response
response approved
Ministerial correspondence addressed
response not approved

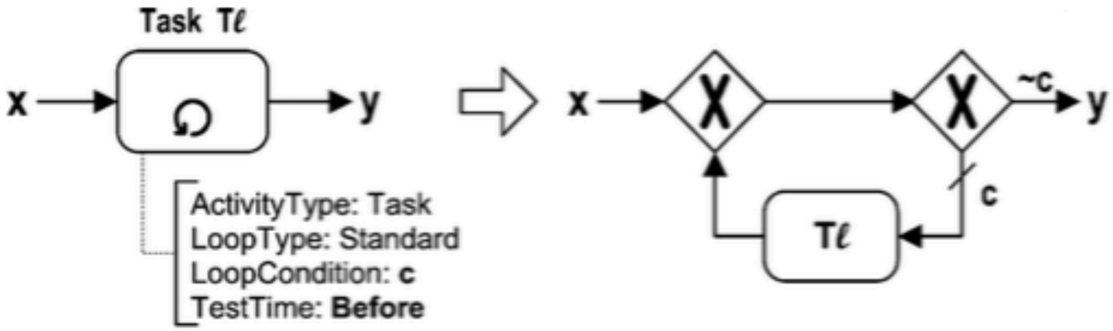
equivalent to:


Until response app roved
Ministerial enquiry received
Assign ministerial enquiry
Investigate ministerial enquiry
Finalize ministerial response
Ministerial correspondence addressed
Enquiry investigated
Prepare ministerial response
Review ministerial response
Response reviewed

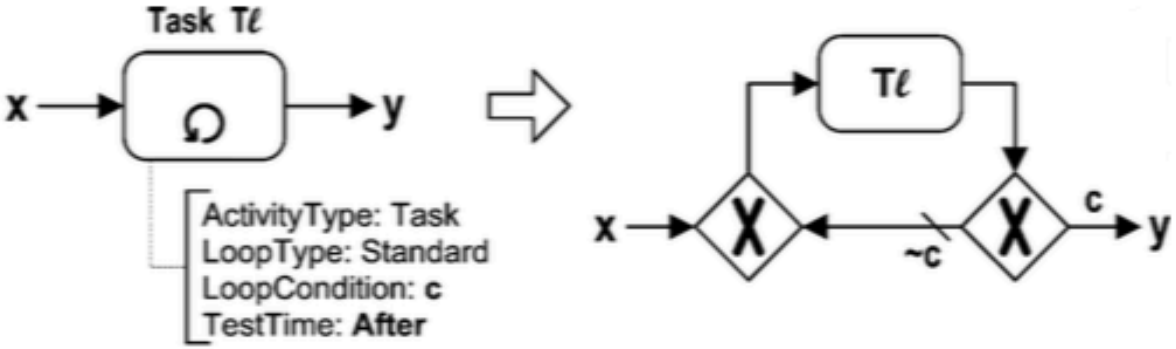
Identify parts of a process that should be executed multiple time in parallel


PO request received
Obtain quote from Supplier 1
Obtain quote from Supplier 2
Obtain quote from Supplier 3
Obtain quote from Supplier 4
Obtain quote from Supplier 5
Select best quote
Emit order
Order emitted

More On the Loop

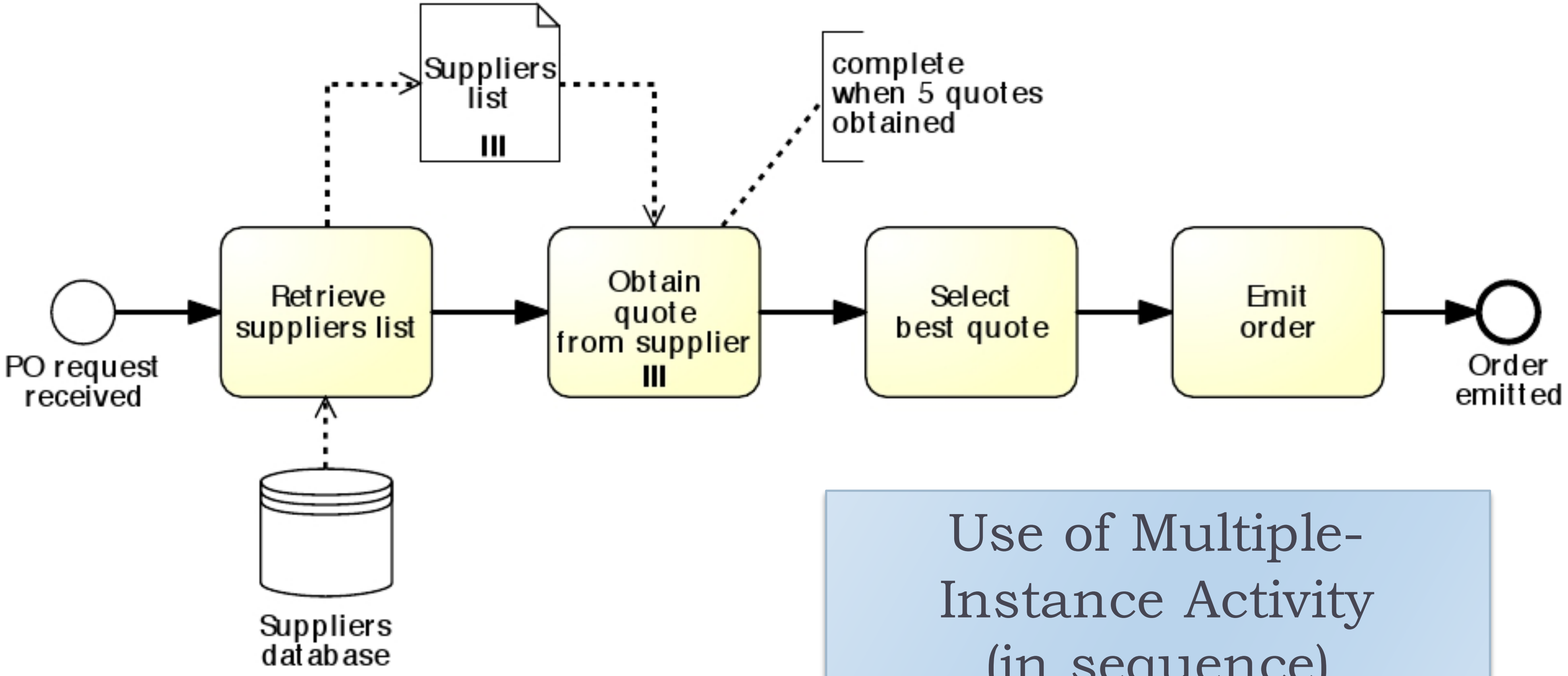


(a) “while-do” loop



(b) “do-until” loop

Use of Multiple-Instance Activity (in paralle)



Use of Multiple-Instance Activity (in sequence)

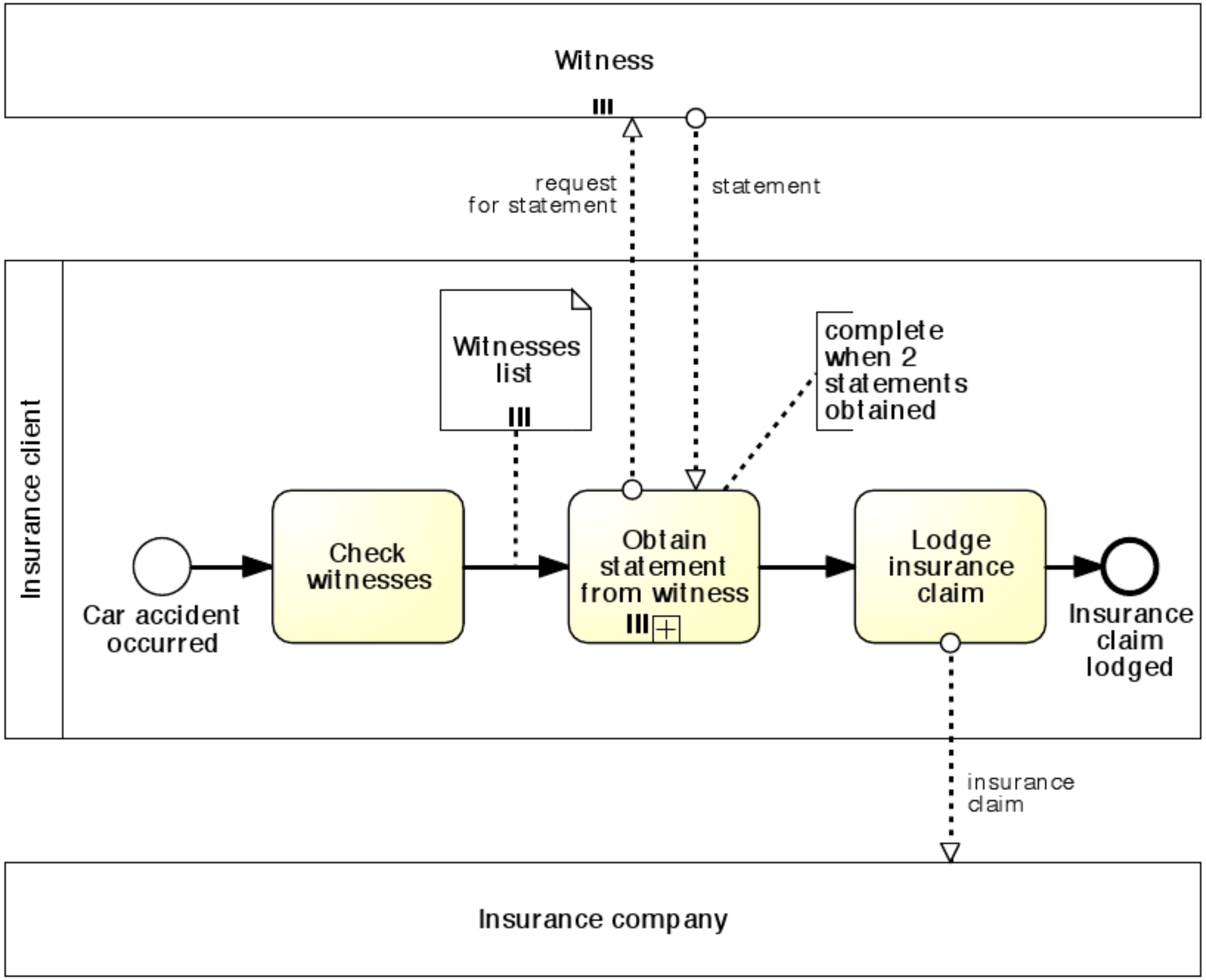
☰ Sequential MI Marker

Exercise - Model the following process fragment

After a car accident, a statement is sought from two witnesses out of the five that were present, in order to lodge the insurance claim.

As soon as the first two statements are received, the claim can be lodged with the insurance company without waiting for the other statements.

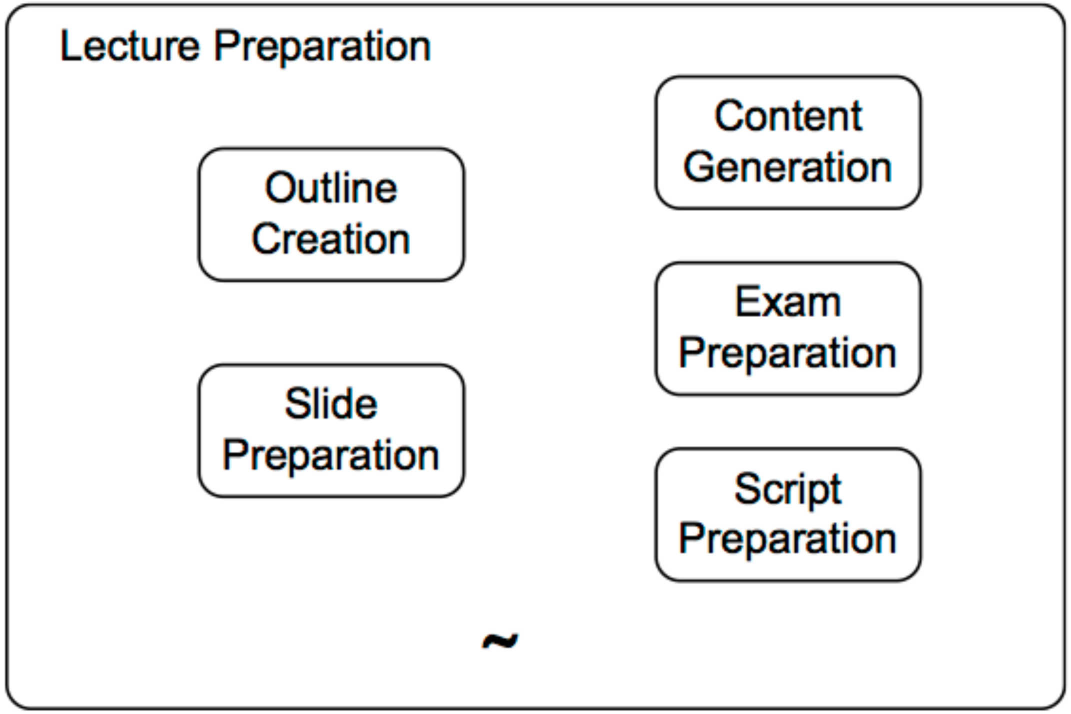
Solution


Witness
request for statement
statement
Insurance client
Witnesses list
complete when 2 statements obtained
Car accident occurred
Check witnesses
Obtain statement from witness
Lodge insurance claim
Insurance claim lodged
insurance claim
Insurance company

Uncontrolled Repetition: Sample adhoc process

Sometimes we may need to model that one or more activities can be repeated a number of times, without a specific order, until a condition is met.

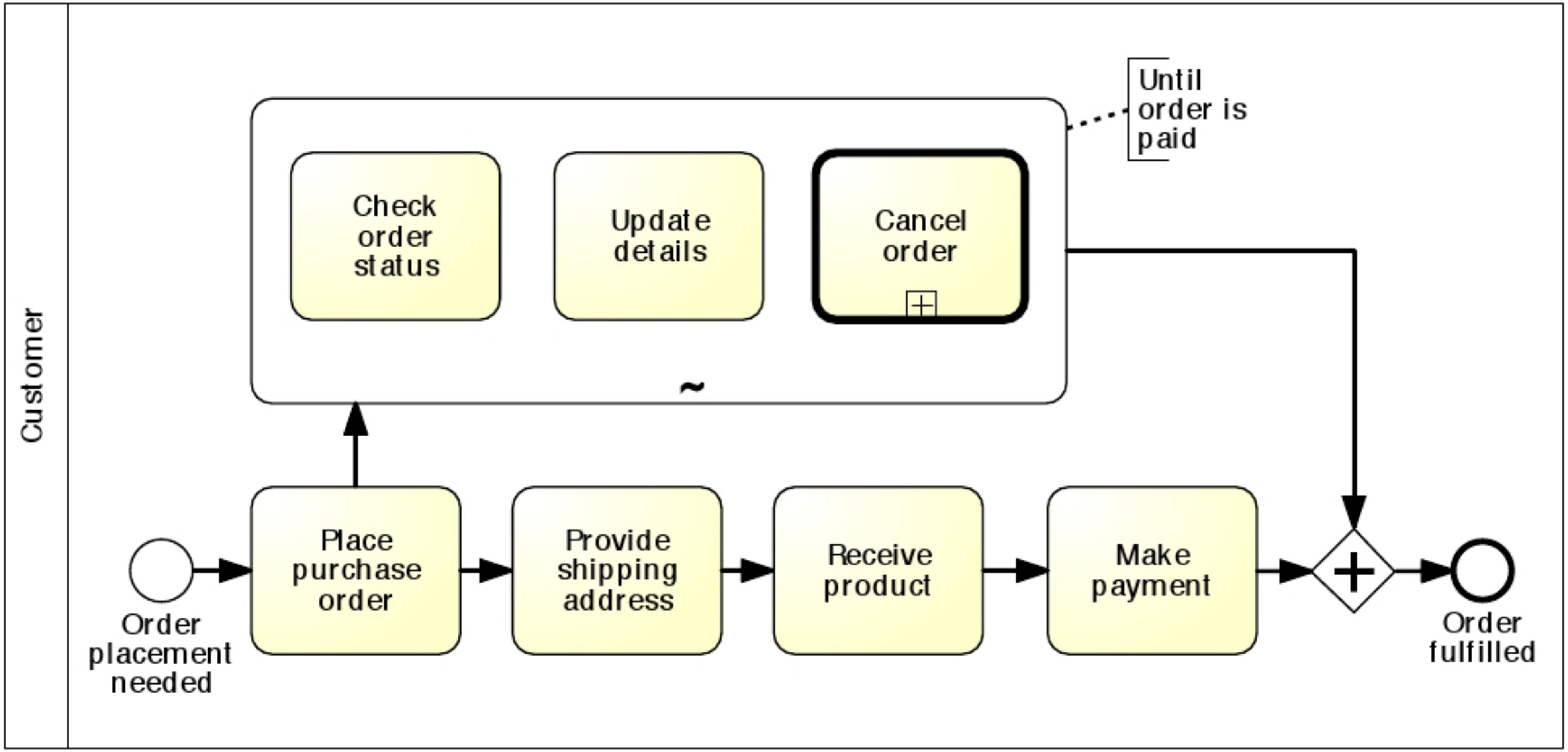
These activities are **uncontrolled**, in the sense that they may be repeated multiple times with no specific order, or not occur at all, until a condition is met—in our case the order being fulfilled.



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

A **partial order may be established** among the activities of an ad-hoc sub-process via the sequence flow. However, **we cannot represent start and end events** in an ad-hoc sub-process.

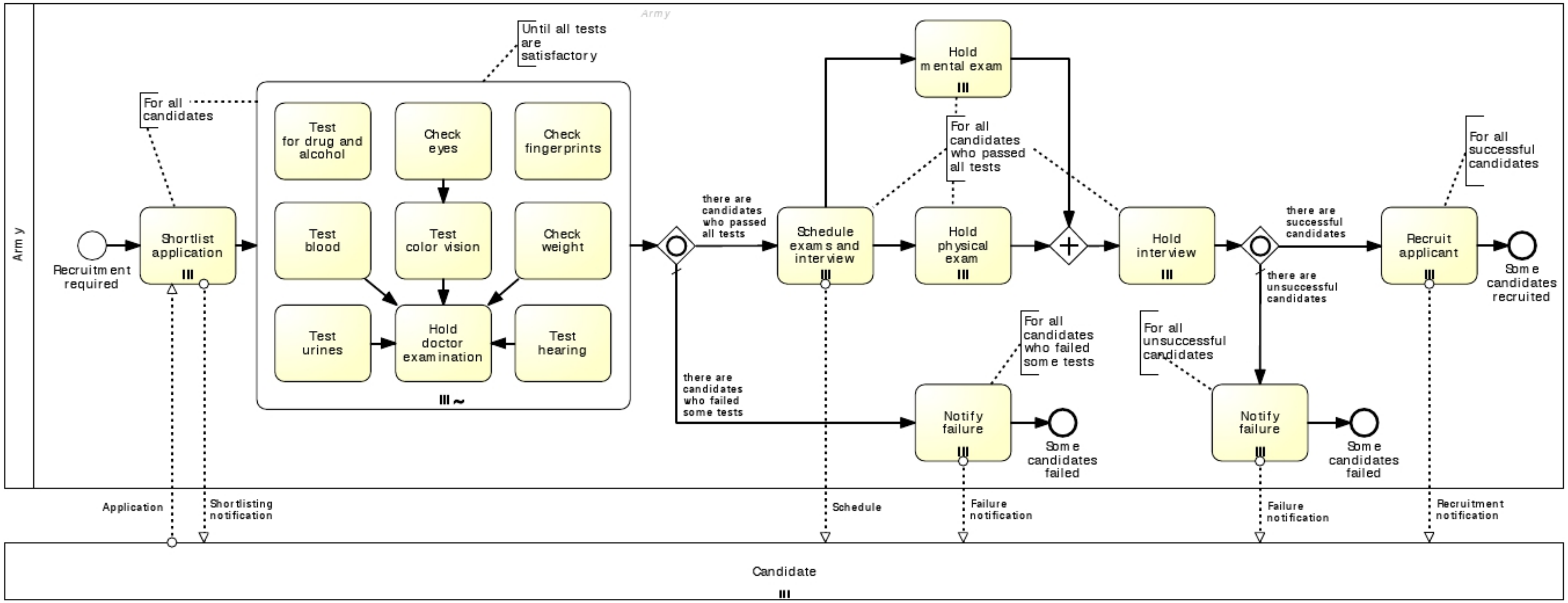
Using an ad-hoc sub-process to model uncontrolled repetition: example



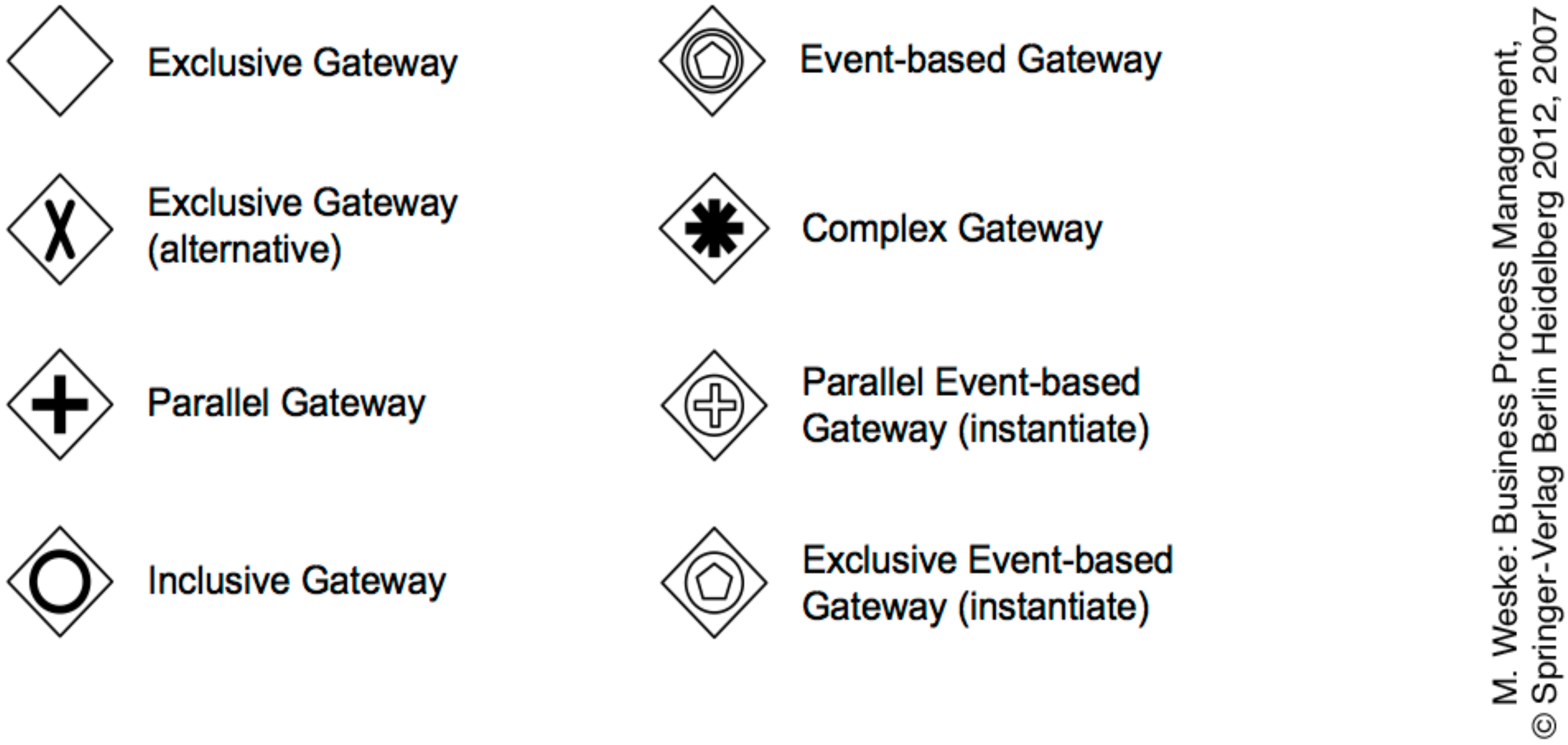
Exercise - Model the following process snippet

A typical army recruitment process starts by shortlisting all candidates' applications. Those shortlisted are then called to sit the following tests: drug and alcohol, eye, color vision, hearing, blood, urine, weight, fingerprinting and doctor examination. The color vision can only be done after the eye test, while the doctor examination can only be done after color vision, hearing, blood, urine and weight have been tested. Moreover, it may be required for some candidates to repeat some of these tests multiple times in order to get a correct assessment, e.g. the blood test may need to be repeated if the candidate has taken too much sugar in the previous 24 hours. The candidates that pass all tests are asked to sit a mental exam and a physical exam, followed by an interview. Only those that also pass these two exams and perform well in the interview can be recruited in the army.

Solution


Army
Army
Recruitment required
For all candidates
Shortlist application
Until all tests are satisfactory
Test for drug and alcohol
Check eyes
Check fingerprints
Test blood
Test color vision
Check weight
Test urines
Hold doctor examination
Test hearing
there are candidates who passed all tests
there are candidates who failed some tests
Schedule exams and interview
Hold mental exam
For all candidates who passed all tests
Hold physical exam
Hold interview
For all candidates who failed some tests
Notify failure
Some candidates failed
there are successful candidates
there are unsuccessful candidates
For all unsuccessful candidates
Notify failure
Some candidates failed
For all successful candidates
Recruit applicant
Some candidates recruited
Application
Shortlisting notification
Schedule
Failure notification
Failure notification
Recruitment notification
Candidate

Types of gateways



Exclusive Gateway

Exclusive Gateway (alternative)

Parallel Gateway

Inclusive Gateway

Event-based Gateway

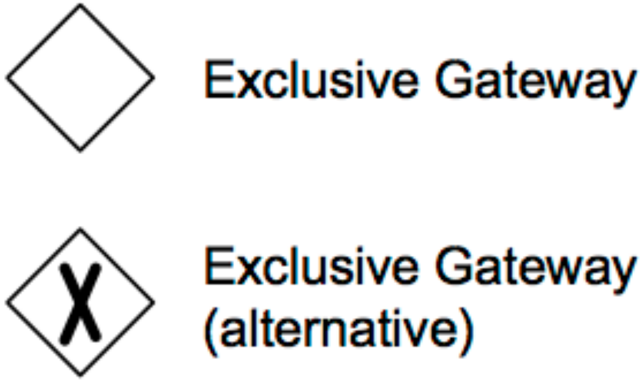
Complex Gateway

Parallel Event-based Gateway (instantiate)

Exclusive Event-based Gateway (instantiate)

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

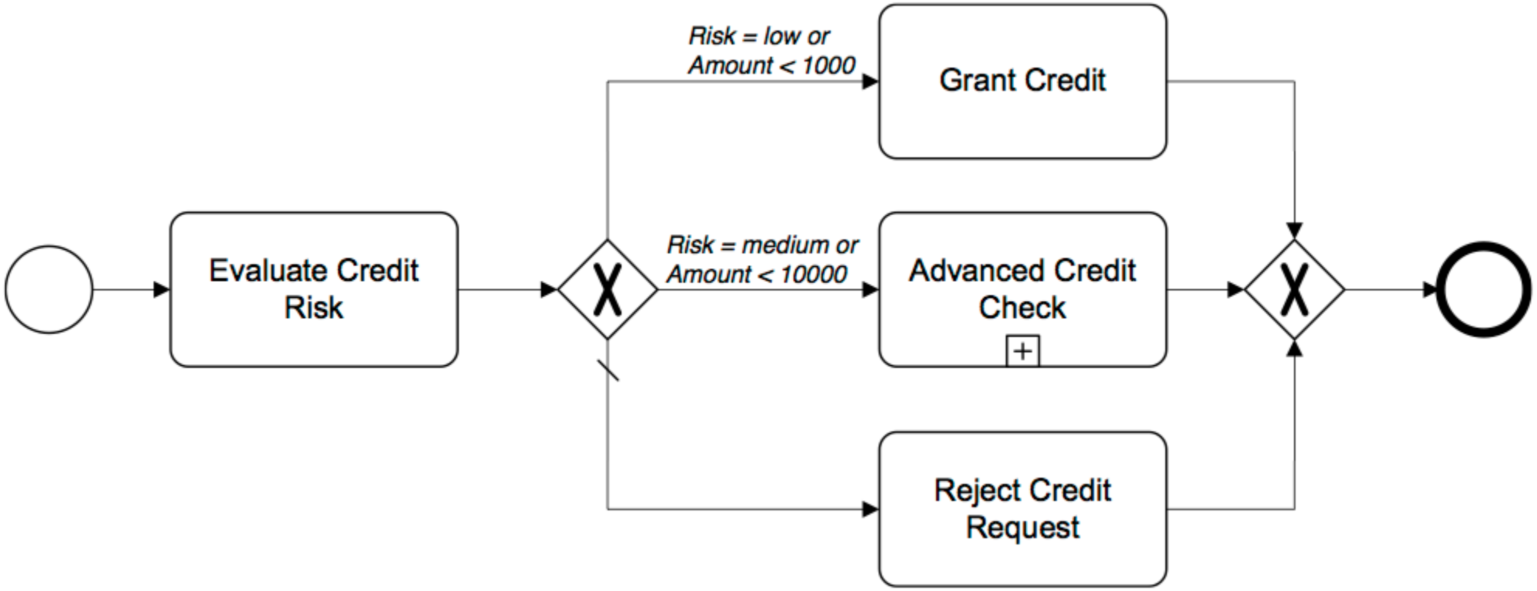
Exclusive Gateways



- Indicates **locations within a business process where the sequence flow can take two or more alternative paths**.
- Only **one of the paths** can be taken.
- Depicted by a **diamond shape** that **may contain** a marker that is shaped like an “X”.
- We use a **XOR-join to merge** two or more alternative branches that may have previously been **forked with a XOR-split**.

Exclusive gateway with default flow

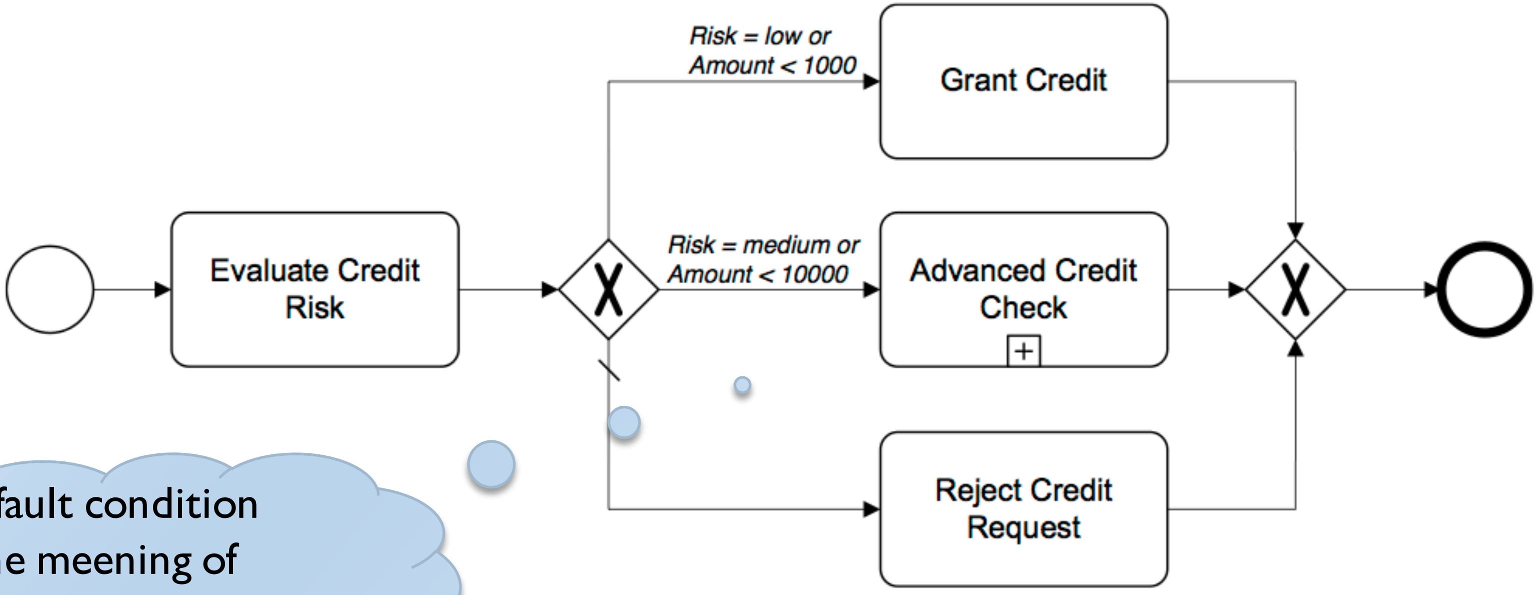
- Exclusive gateways are locations within a process where there are two or more alternative paths.
- The criteria for the decision, which the exclusive gateway represents, exist as conditions on each of the outgoing sequence flow.
- When a token arrives at an exclusive gateway, there is an immediate evaluation of the conditions that are on the gateway's outgoing sequence flow. One of those conditions must always evaluate to true.



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

Exclusive gateway with default flow

- One way for the modeler to ensure that the process does not get stuck at an exclusive gateway is to use a default condition for one of the outgoing sequence flow.
- The default condition can complement a set of standard conditions to provide an automatic escape mechanism in case all the standard conditions evaluate to false.
- The default is chosen if all the other sequence flow conditions turn out to be false.

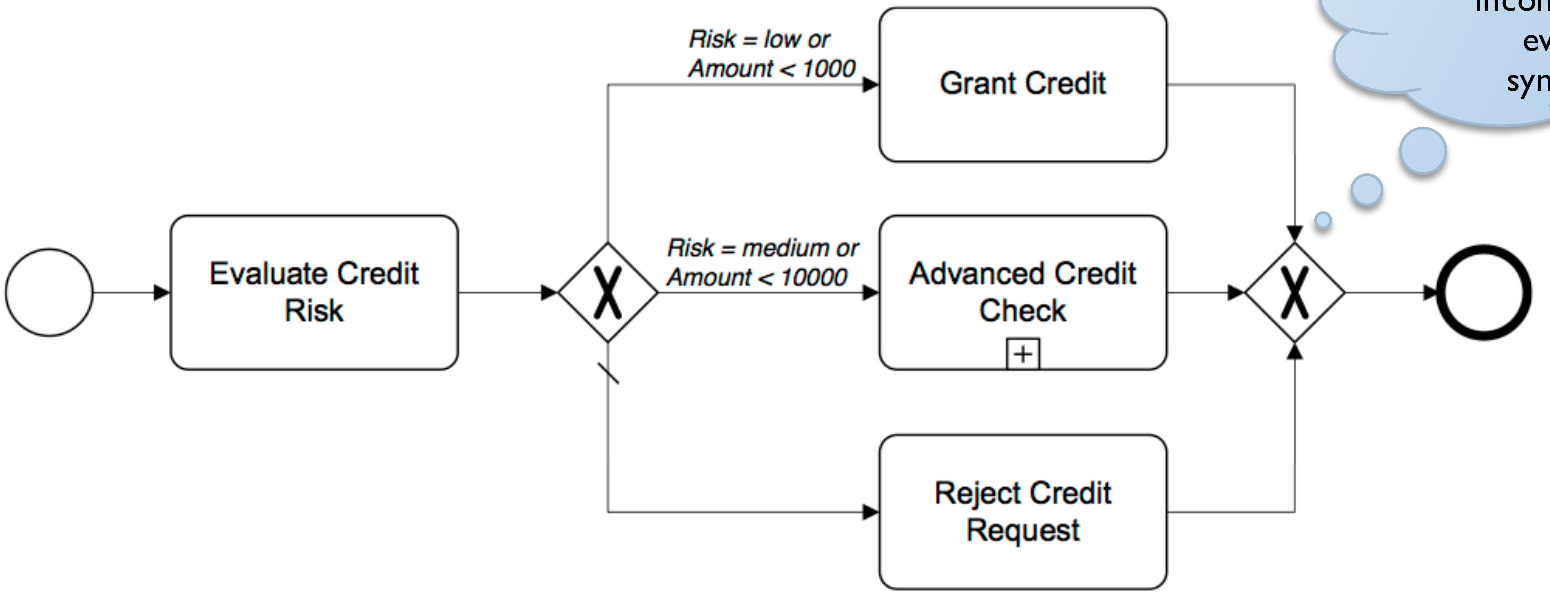


The default condition has the meening of “otherwise” and it can be unlabeled

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

Exclusive gateway with default flow

- When a token arrives at the exclusive gateway, there is no evaluation of conditions (on the incoming sequence flow), and immediately moves down the outgoing sequence flow.



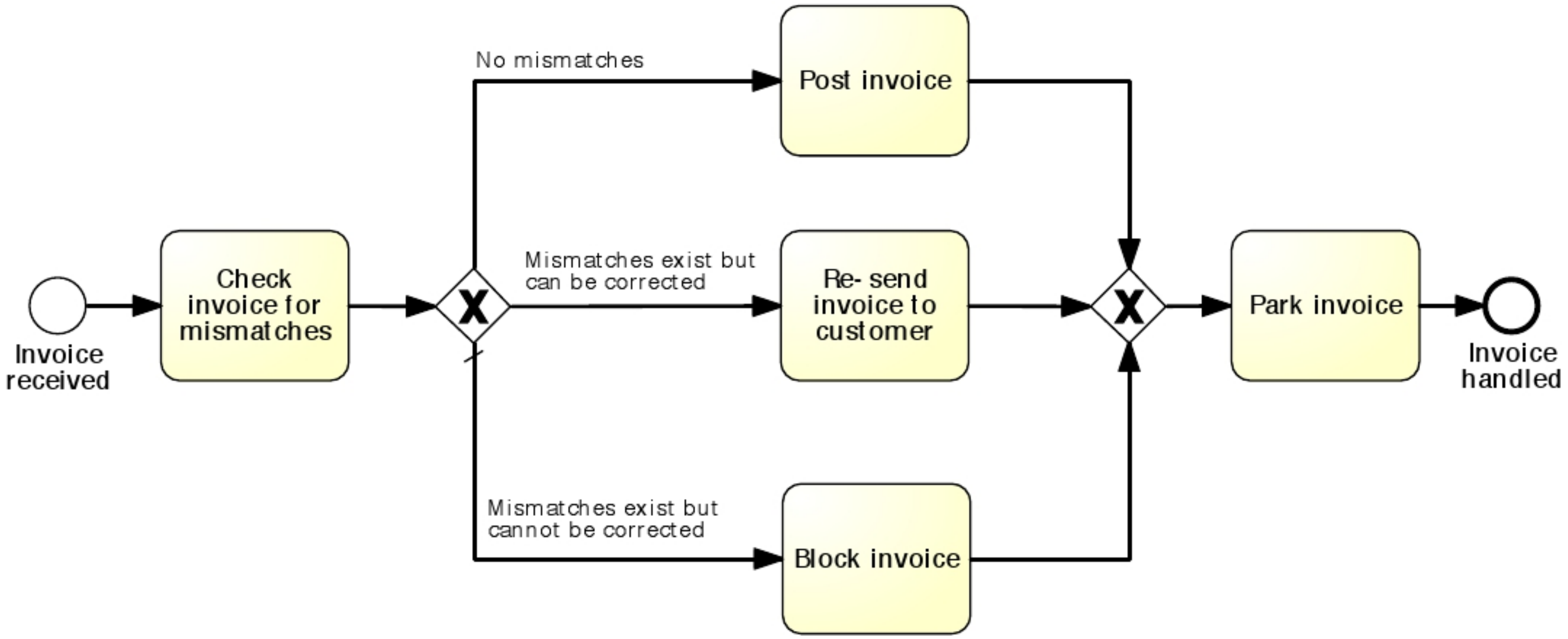
The token is passed-through. There is not delay incoming condition evalution or syncronization

M. Weske: Business Process Manag
© Springer-Verlag Berlin Heidelberg 20

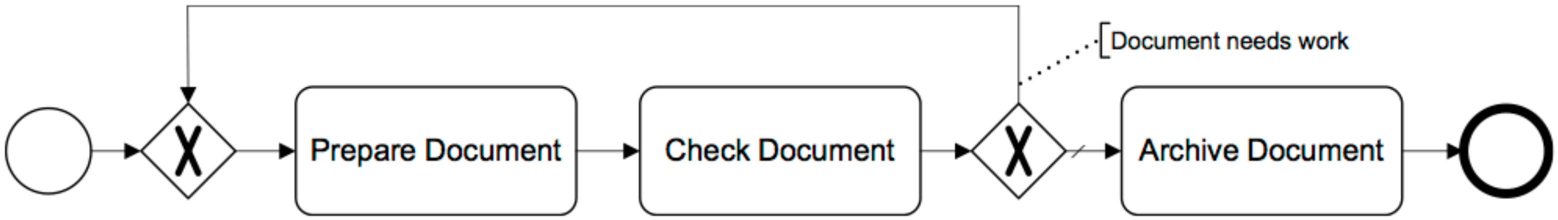
Exercise

- As soon as an invoice is received from a customer, it needs to be checked for mismatches.
- The check may result in either of these three options:
  i) there are no mismatches, in which case the invoice is posted;
  ii) there are mismatches but these can be corrected, in which case the invoice is re-sent to the customer; and
  iii) there are mismatches but these cannot be corrected, in which case the invoice is blocked.
- Once one of these three activities is performed the invoice is parked and the process completes.

Solution

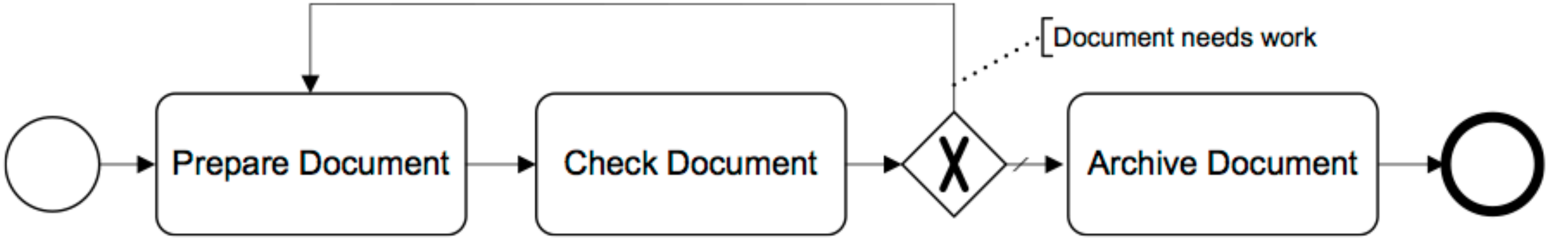

No mismatches
Post invoice
Invoice received
Check invoice for mismatches
Mismatches exist but can be corrected
Re- send invoice to customer
Park invoice
Invoice handled
Mismatches exist but cannot be corrected
Block invoice

Exclusive gateways realizing a loop



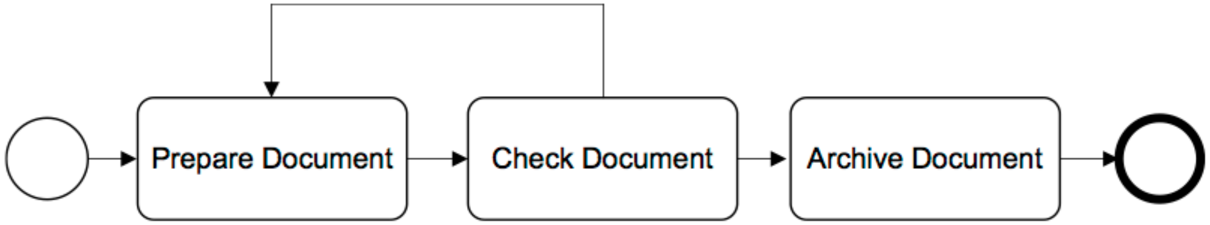
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

BP with uncontrolled flow



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

BP with split and join activities - livelock

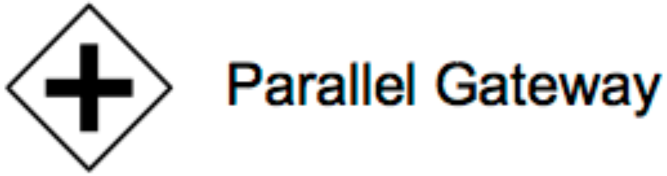


M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

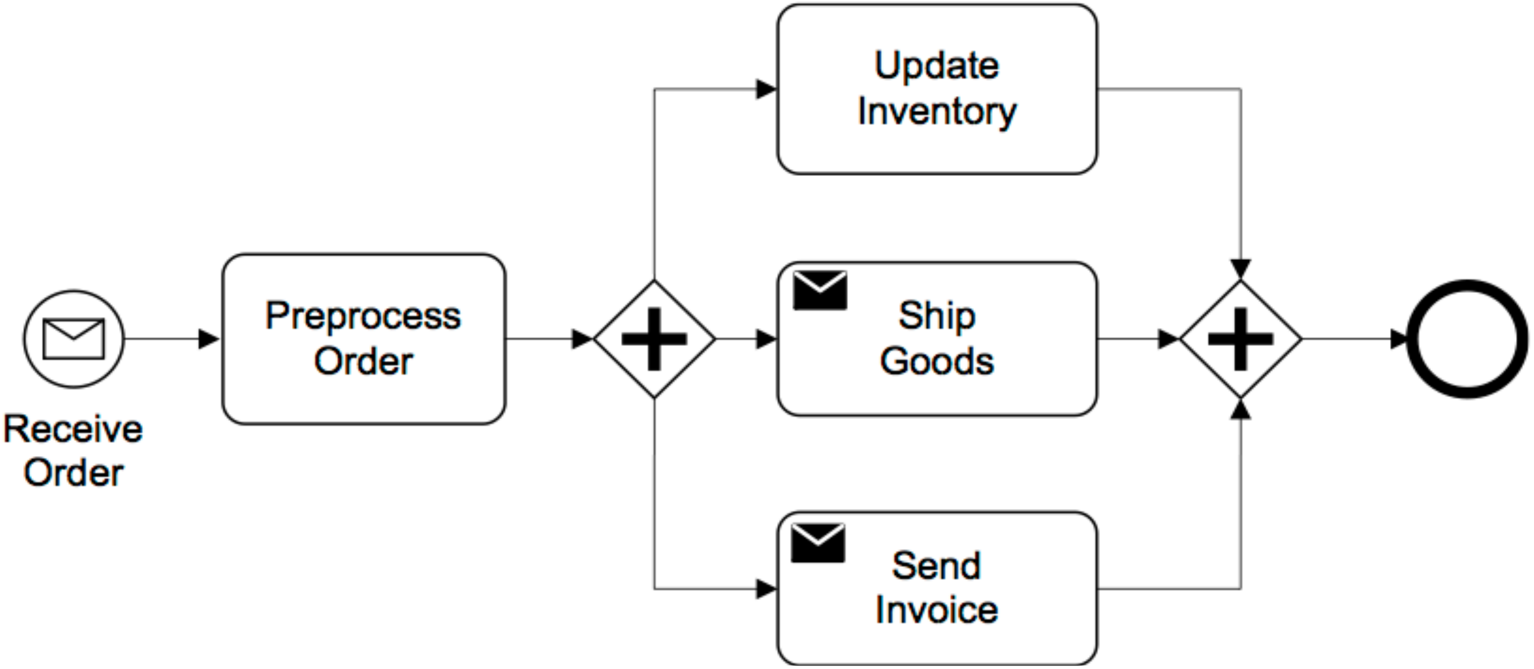
Parallel Gateways

- Provide a mechanism to synchronize parallel flows (AND-join) and to create parallel flows (AND-split), with activities that can be executed concurrently.

- Depicted by a diamond shape that must contain a marker that is shaped like a plus sign.



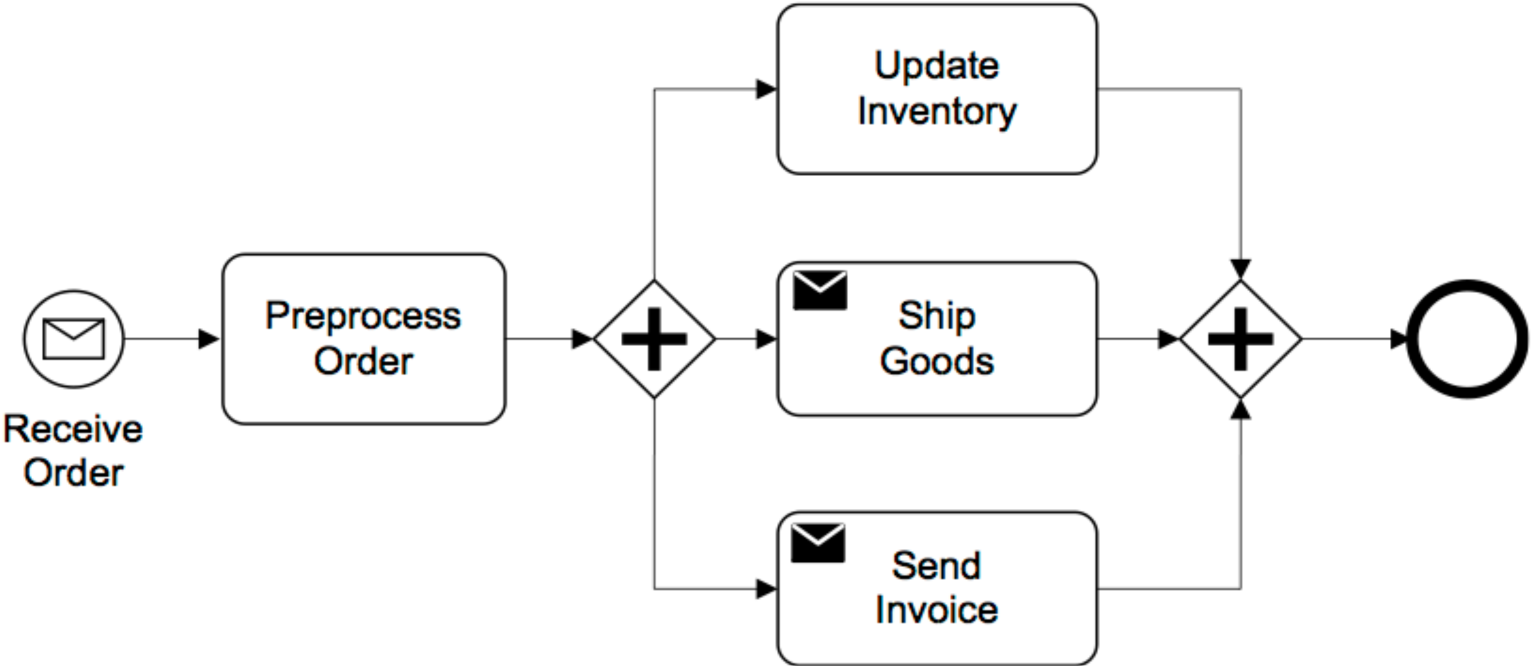
Parallel gateway (Splitting Behaviour)



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

- When a token arrives at a parallel gateway, there is no evaluation of any conditions on the outgoing sequence flow.
- The parallel gateway will create parallel paths.
- This means that the gateway will create a number of tokens that are equal to the number of outgoing sequence flow. One token moves down each of those outgoing sequence flow.

Parallel gateway (Merging Behaviour)

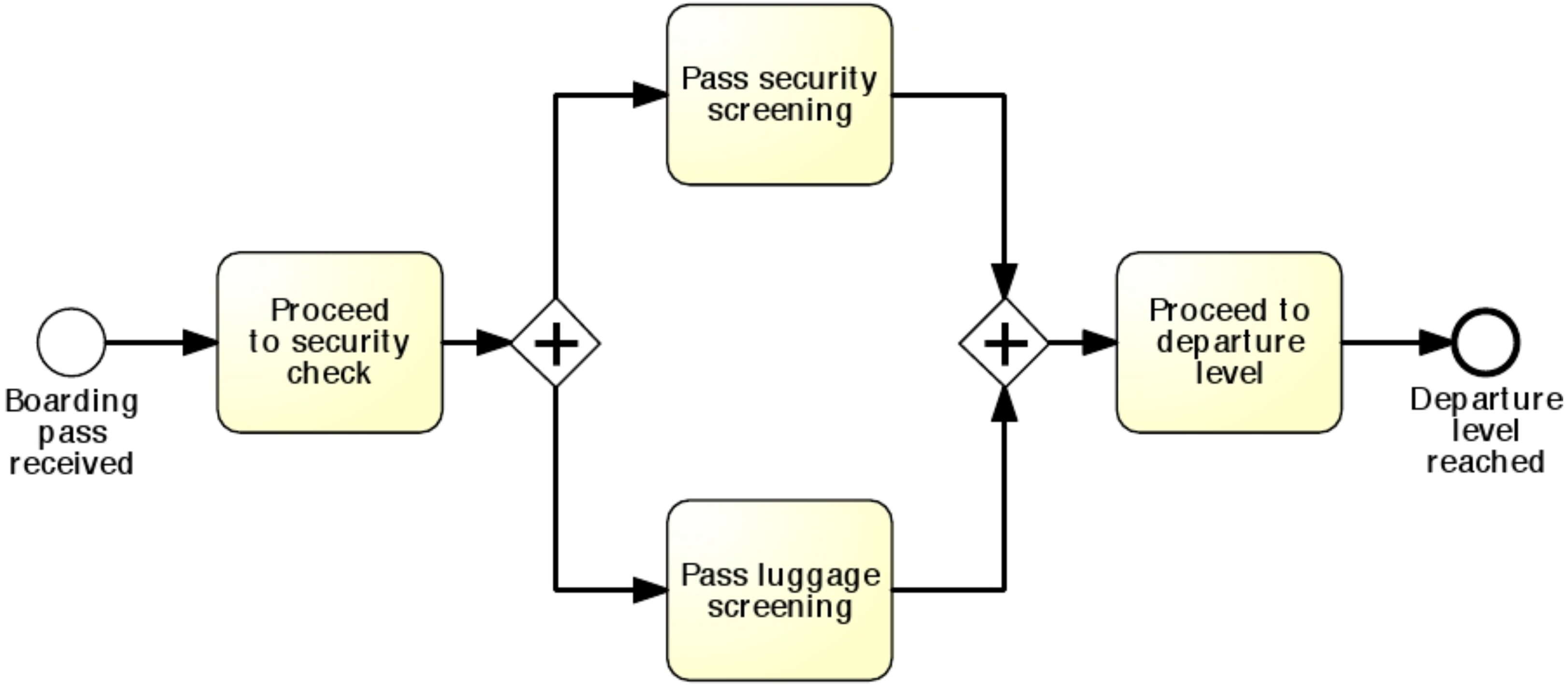


M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

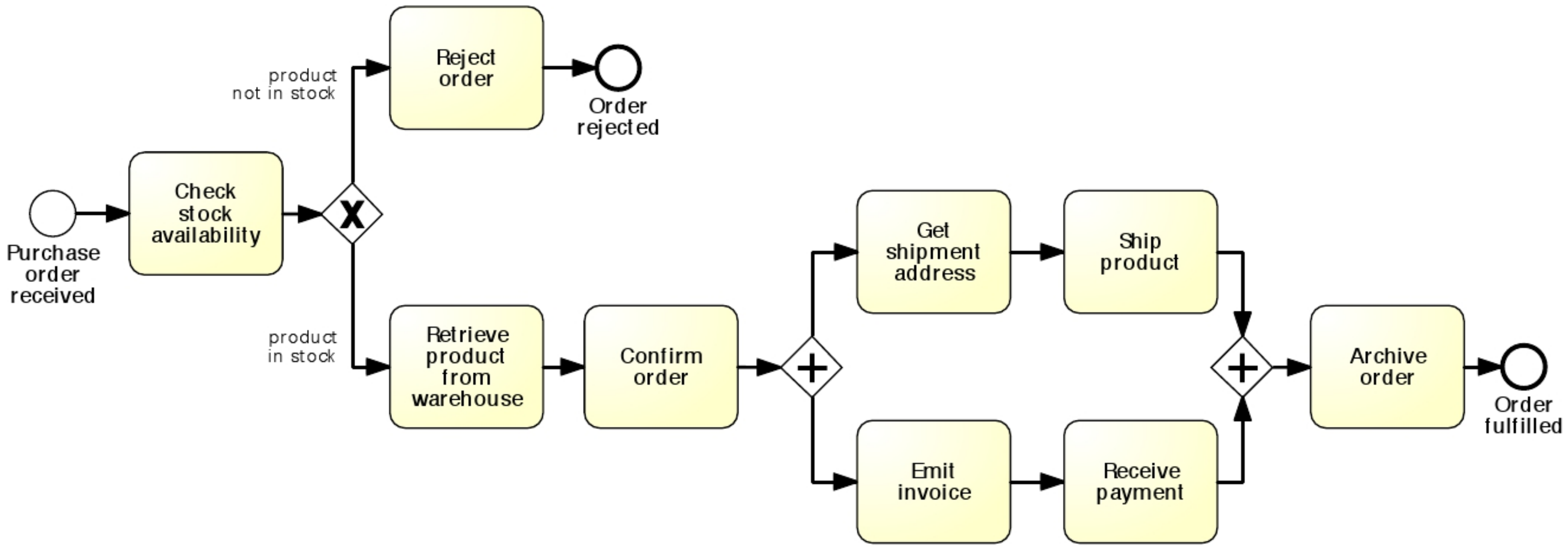
- To synchronize the flow, the parallel gateway will wait for a token to arrive from each incoming sequence flow.
- When the first token arrives, there is no evaluation of a condition for the incoming sequence flow, but the token is "held" at the gateway and does not continue.
- When all the tokens are arrived, then they are merged and one token moves down the outgoing sequence flow.

Exercise

- Once the boarding pass has been received, passengers proceed to the security check. Here they need to pass the personal security screening and the luggage screening. Afterwards, they can proceed to the departure level.

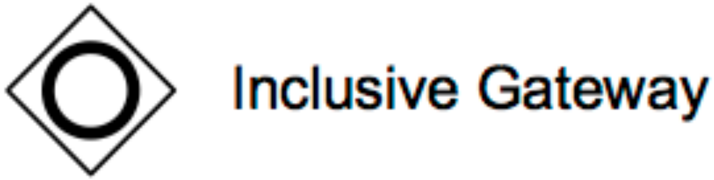


Gateways Combination

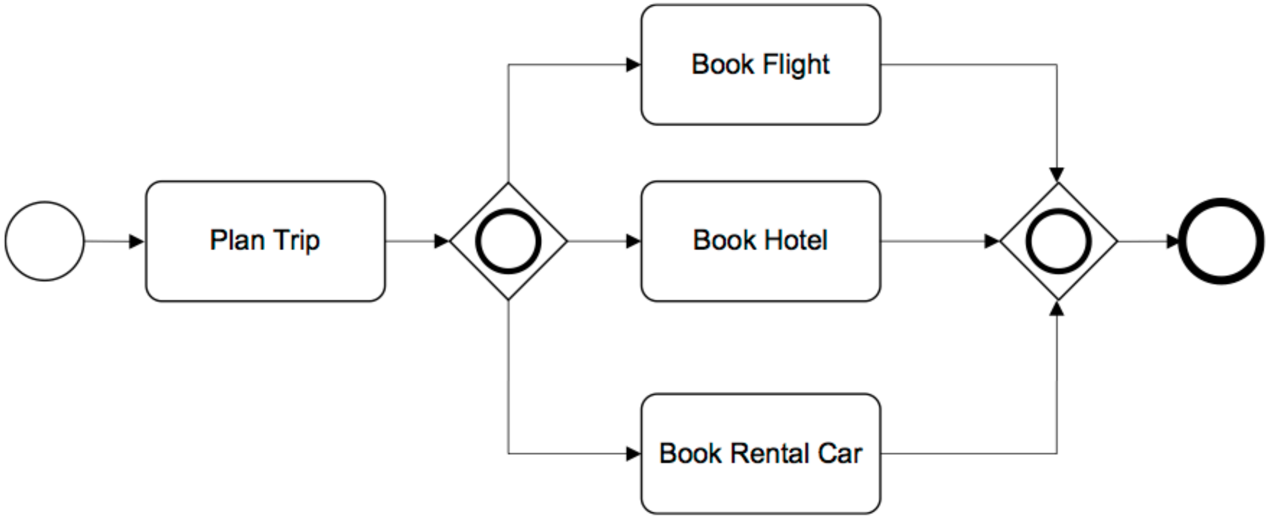


Inclusive Gateways

- Sometimes we may need to take one or more branches after a decision activity.
- To model situations where a decision may lead to one or more options being taken at the same time, we need to use an inclusive (OR) split gateway.
- An OR-split is similar to the XOR-split, but the conditions on its outgoing branches do not need to be mutually exclusive, i.e. more than one of them can be true at the same time.
- When we encounter an OR-split, we thus take one or more branches depending on which conditions are true.



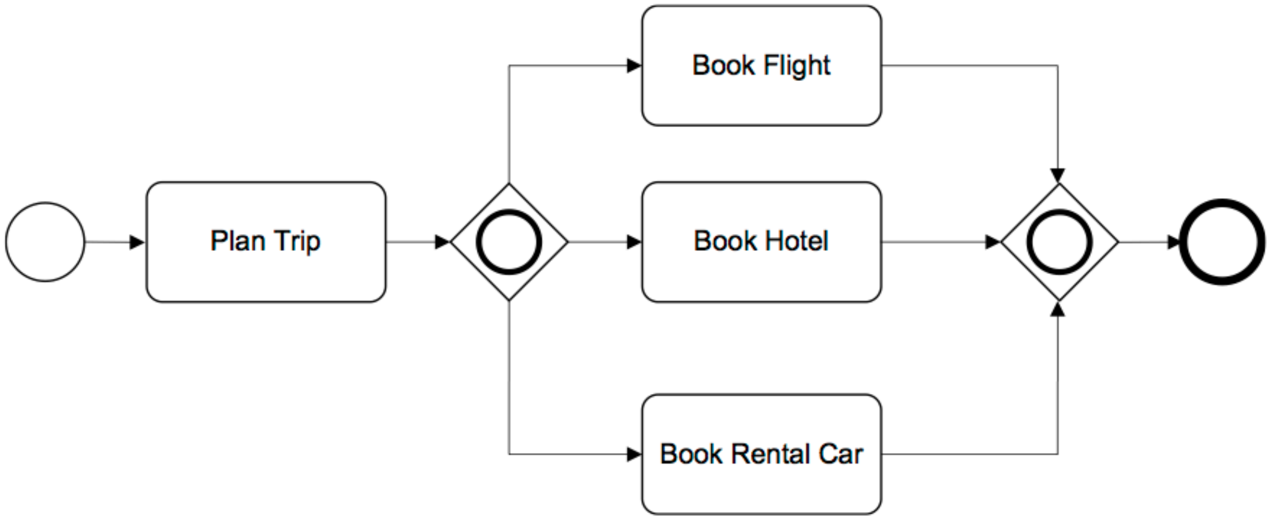
Inclusive or gateway (Splitting Behaviour)



Weske: Business Process Management, ringer-Verlag Berlin Heidelberg 2012, 2007

- Inclusive gateways support decisions where more than one outcome is possible at the decision point.
- Inclusive gateway with multiple outgoing sequence flows creates one or more paths based on the conditions on those sequence flow.
- In terms of token semantics, this means that the OR-split takes the input token and generates a number of tokens equivalent to the number of output conditions that are true.
- Every condition that evaluates to true will result in a token moving down that sequence flow.
- At least one of those conditions must evaluate to true.

Inclusive or gateway (Merging Behaviour)



Weske: Business Process Management,
ringer-Verlag Berlin Heidelberg 2012, 2007

- When the first token arrives at the gateway, the gateway will "look" upstream for each of the other incoming sequence flow to see if there is a token that might arrive at a later time.

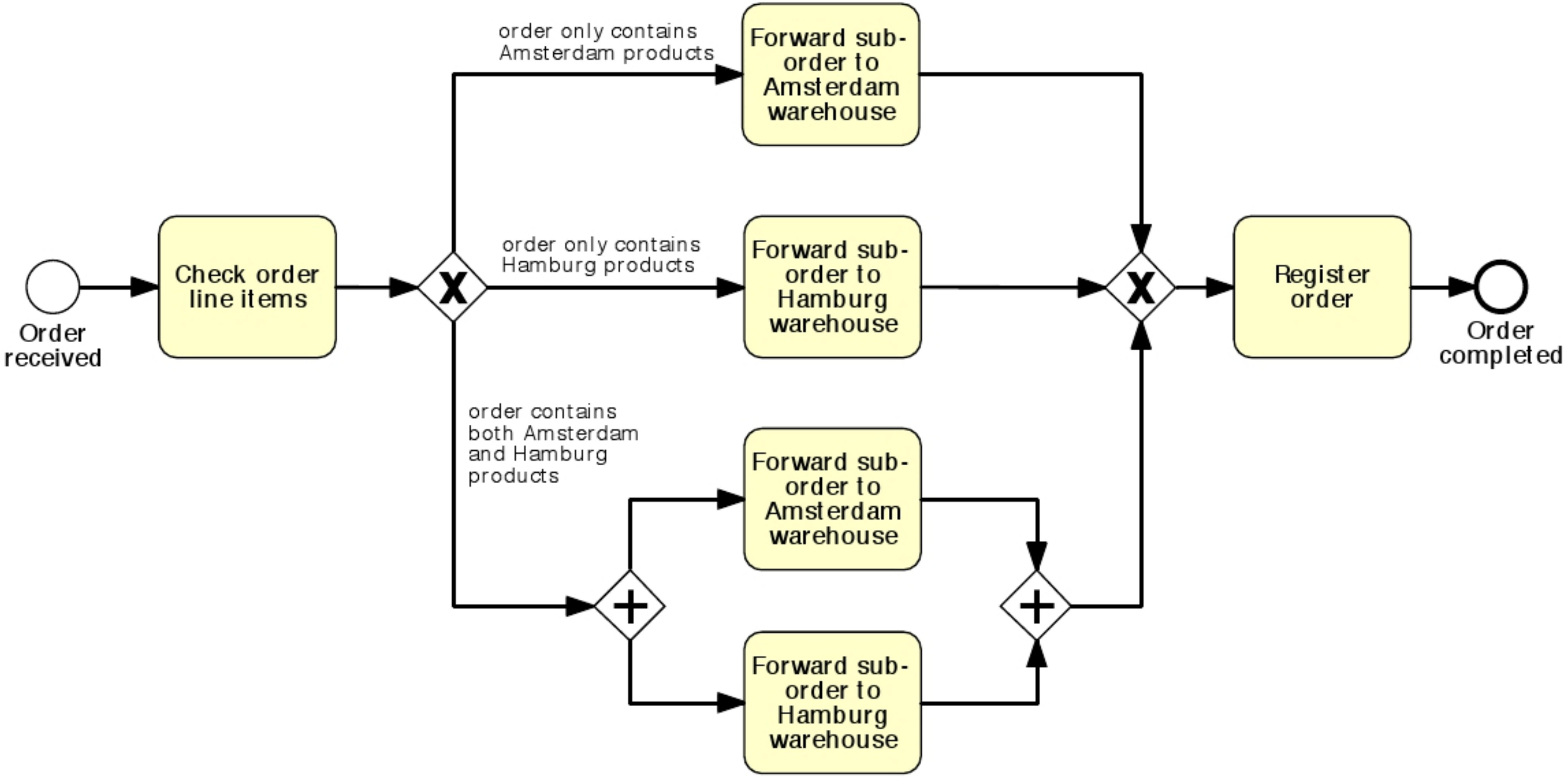
- Thus, the gateway will hold the first token that arrived in the upper path until the other token from the lower path arrives.

- When all the expected tokens have arrived at the gateway, the process flow is synchronized (the incoming tokens are merged) and then a token moves down the gateway's outgoing sequence flow.

Exercise

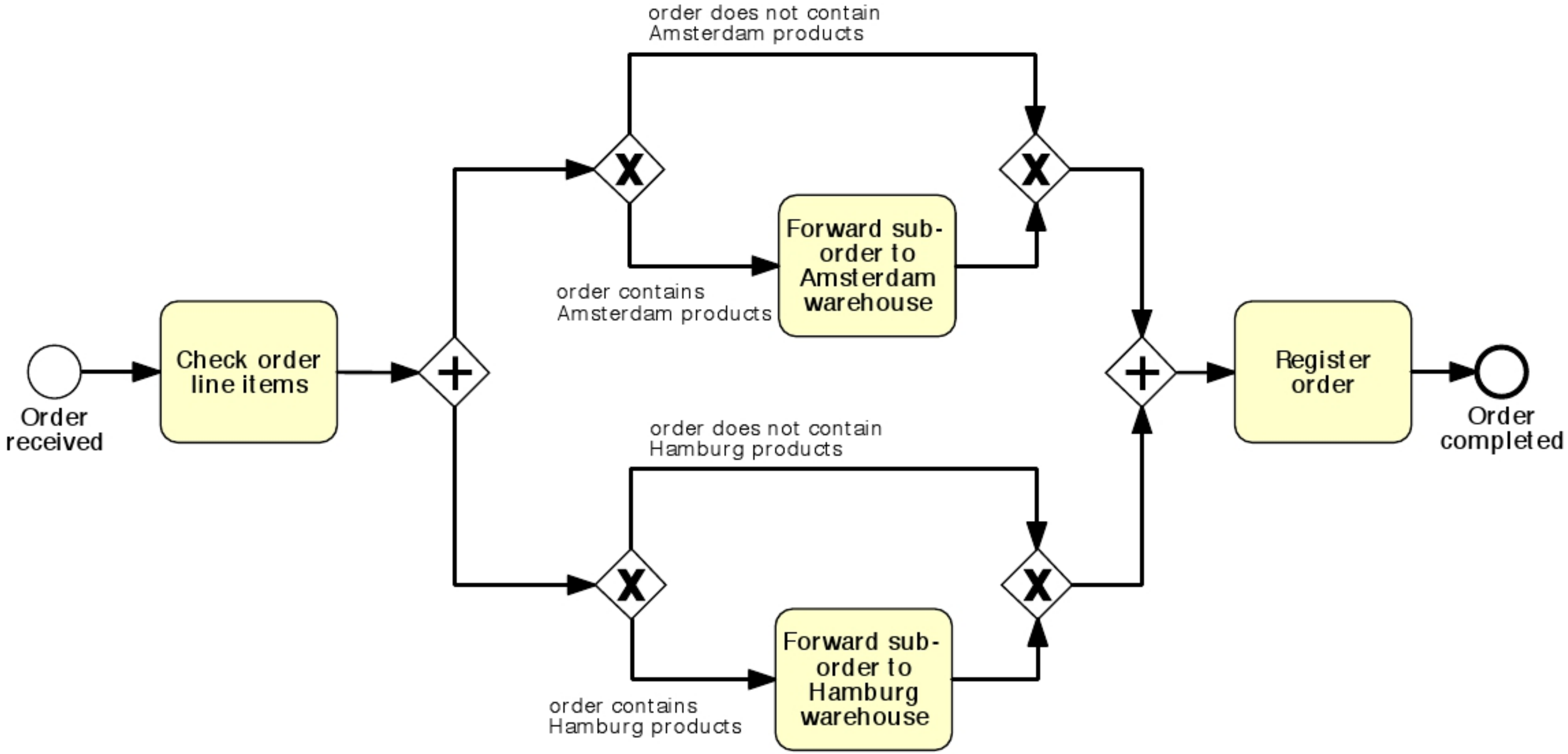
- A company has two warehouses that store different products: Amsterdam and Hamburg.
- When an order is received, it is distributed across these warehouses: if some of the relevant products are maintained in Amsterdam, a sub-order is sent there; likewise, if some relevant products are maintained in Hamburg, a sub-order is sent there.
- Afterwards, the order is registered and the process completes.

A first solution

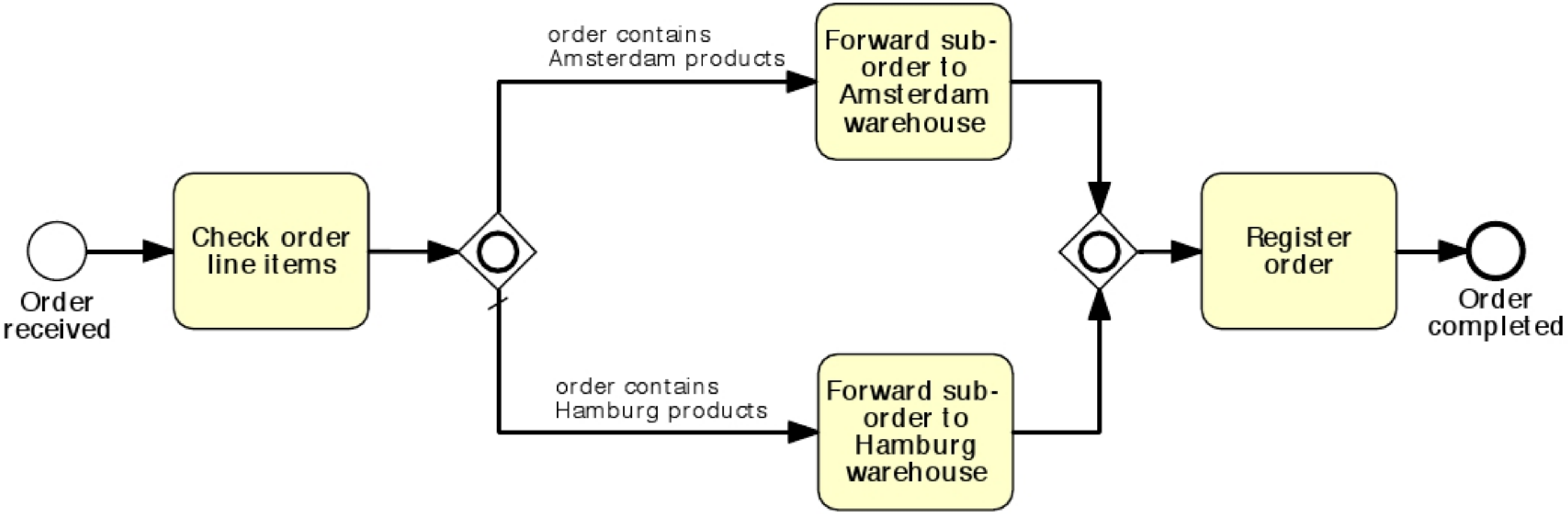


Order received
Check order line items
order only contains Amsterdam products
Forward sub-order to Amsterdam warehouse
order only contains Hamburg products
Forward sub-order to Hamburg warehouse
order contains both Amsterdam and Hamburg products
Forward sub-order to Amsterdam warehouse
Forward sub-order to Hamburg warehouse
Register order
Order completed

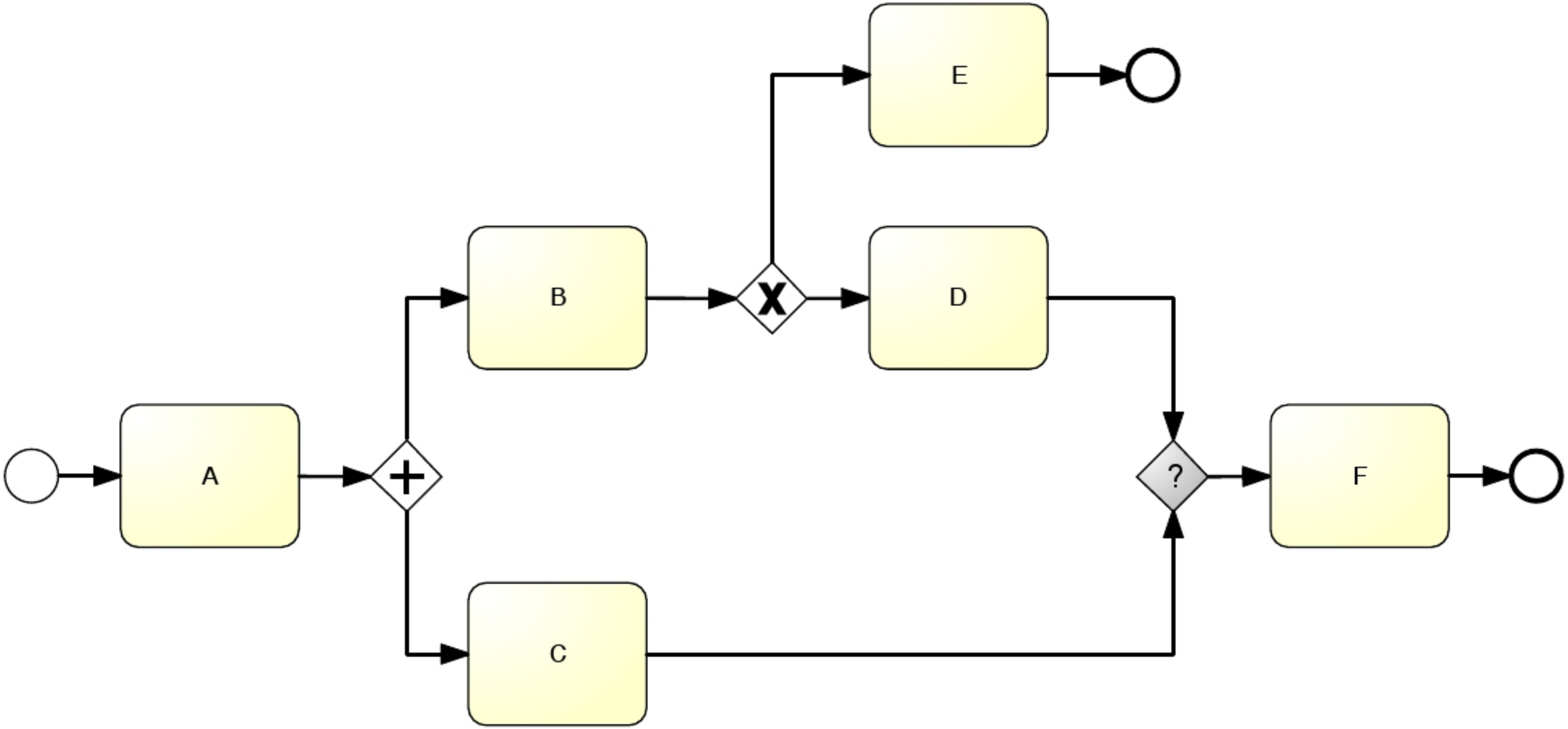
A second solution


order does not contain Amsterdam products
order contains Amsterdam products
Forward sub-order to Amsterdam warehouse
Order received
Check order line items
Register order
Order completed
order does not contain Hamburg products
order contains Hamburg products
Forward sub-order to Hamburg warehouse

A third solution with OR gateways


order contains Amsterdam products
Forward sub-order to Amsterdam warehouse
Order received
Check order line items
Register order
Order completed
order contains Hamburg products
Forward sub-order to Hamburg warehouse

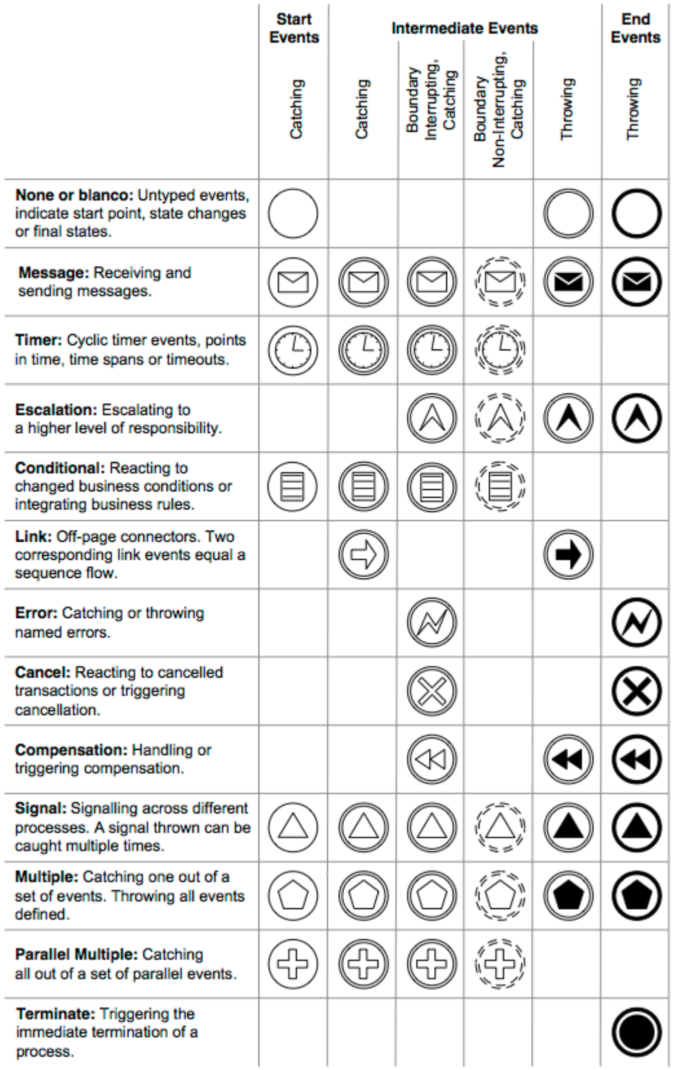
What type should we assign to this join?



Solution

- Since the OR-join semantics is not simple, the presence of this element in a model may confuse the reader.
- Thus, we suggest to use it only when it is strictly required.
- Clearly, it is easy to see that an OR-join must be used whenever we need to synchronize control from a preceding OR-split. Similarly, we should use an AND-join to synchronize control from a preceding AND-split and an XOR-join to merge a set of branches that are mutually exclusive.

Handling BPMN Events



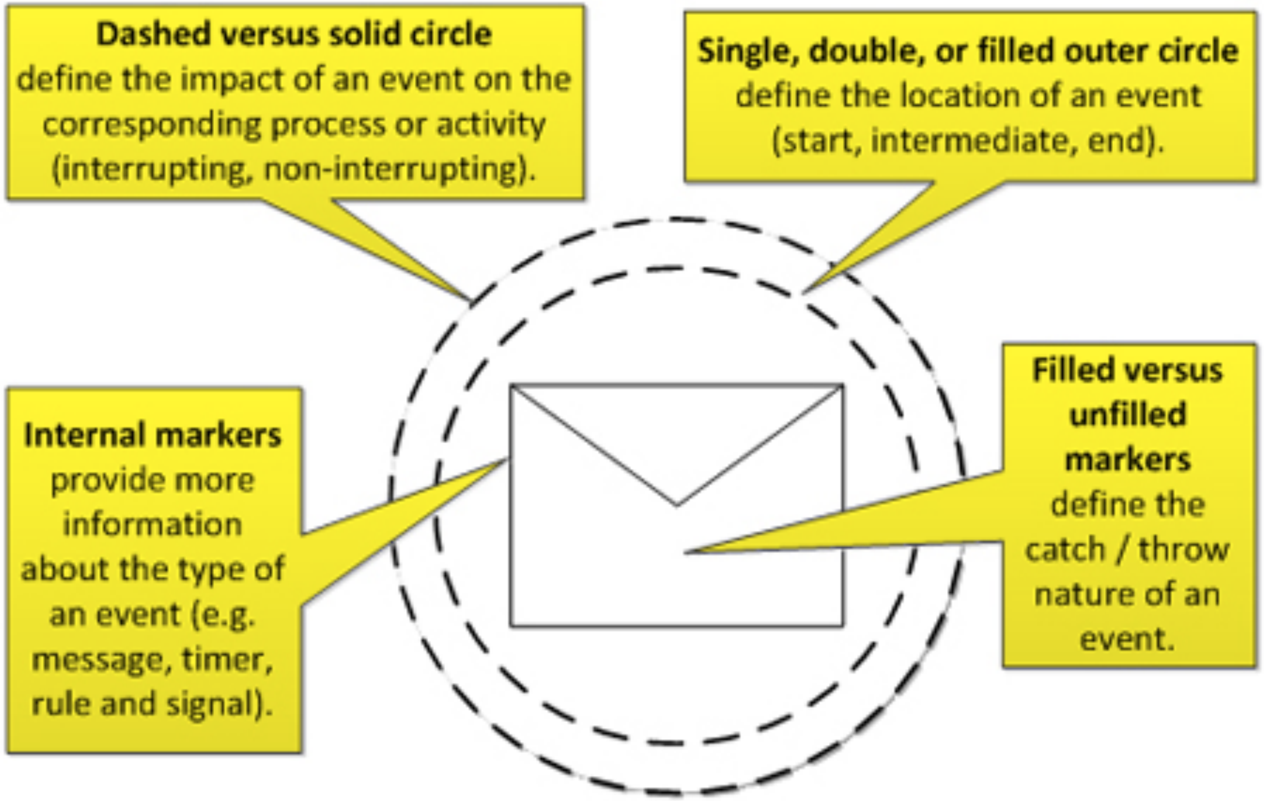
	Start Events	Intermediate Events				End Events
	Catching	Catching	Boundary Interrupting, Catching	Boundary Non-Interrupting, Catching	Throwing	Throwing
None or blanco: Untyped events, indicate start point, state changes or final states.	○				○	○
Message: Receiving and sending messages.	✉	✉	✉	✉	✉	✉
Timer: Cyclic timer events, points in time, time spans or timeouts.	🕒	🕒	🕒	🕒		
Escalation: Escalating to a 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.			ϟ			ϟ
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.						●

- While it comes natural to give a name (also called *label*) to each activity, we should not forget to give labels to events as well.

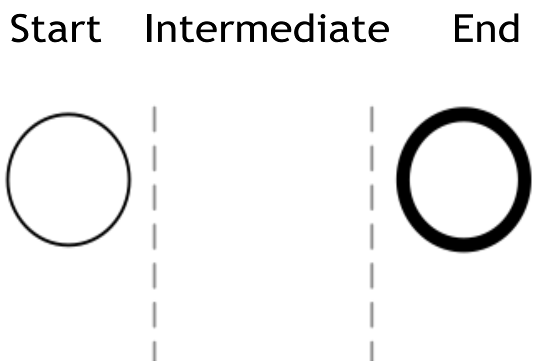
  - For example, giving a name to each start event allows us to communicate what triggers an instance of the process, meaning, when should a new instance of the process be started.

  - Similarly, giving a label to each end event allows us to communicate what conditions hold when an instance of the process completes, i.e. what the outcome of the process is.

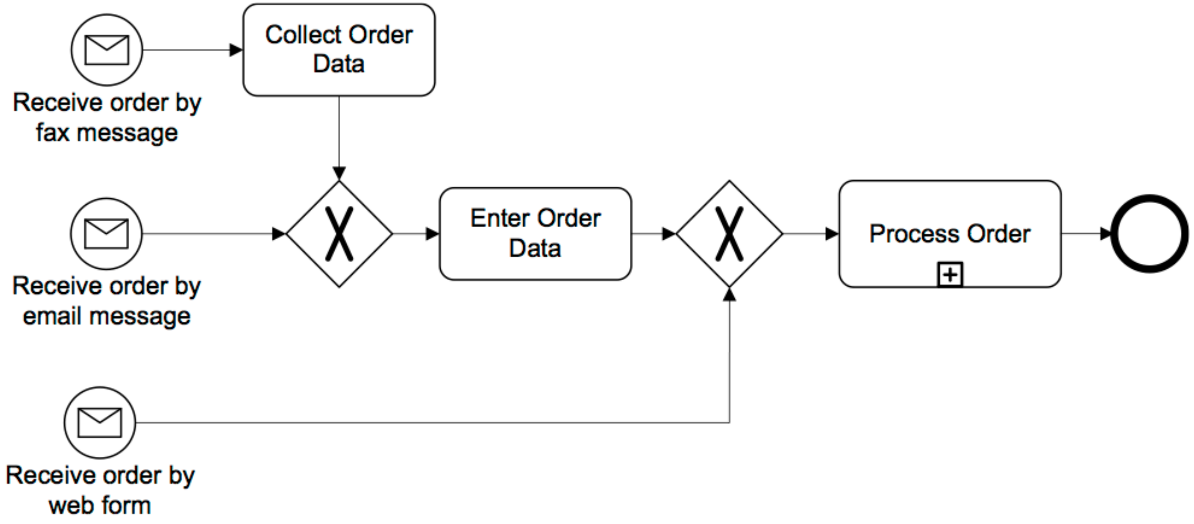
More on Events


Dashed versus solid circle define the impact of an event on the corresponding process or activity (interrupting, non-interrupting).
Single, double, or filled outer circle define the location of an event (start, intermediate, end).
Internal markers provide more information about the type of an event (e.g. message, timer, rule and signal).
Filled versus unfilled markers define the catch / throw nature of an event.

BPMN Events

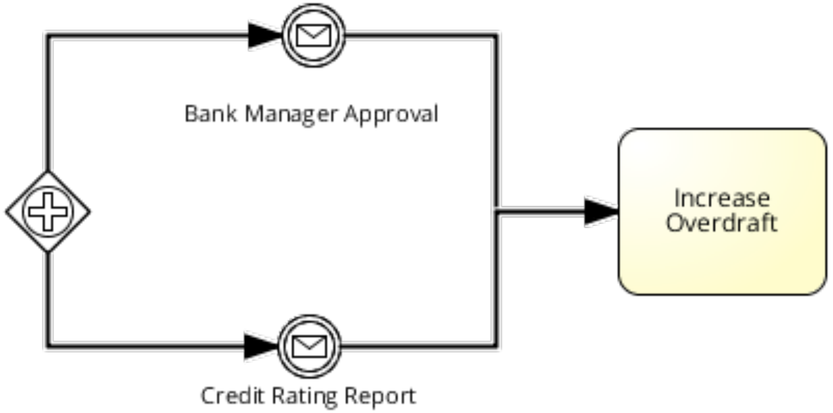


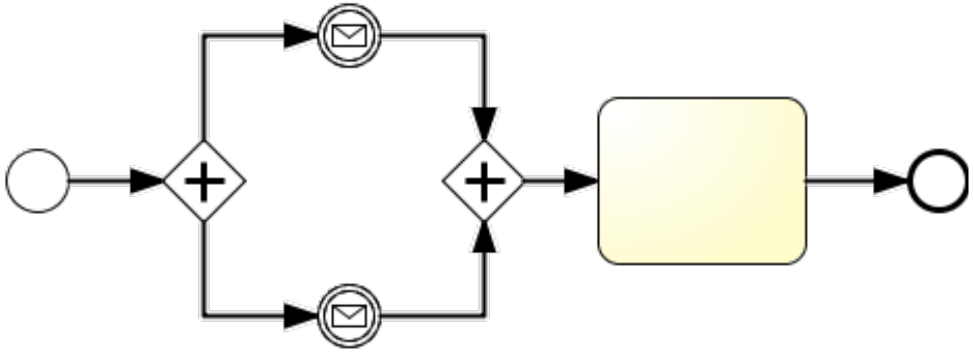
Untyped Event – Indicates that an instance of the process is created (start) or completed (end), without specifying the cause for creation/completion

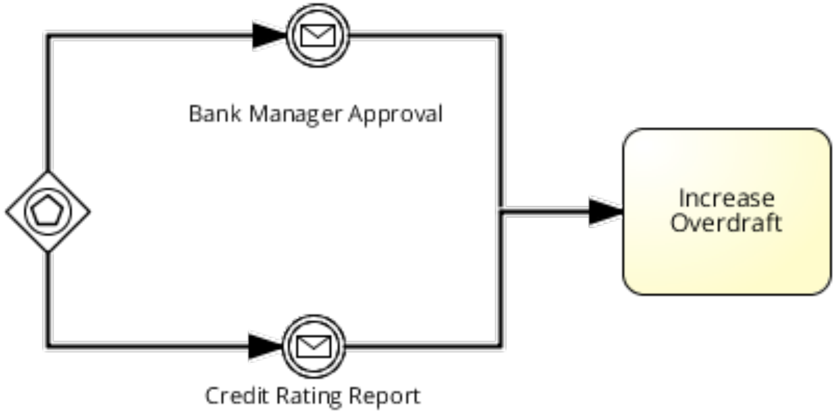

Collect Order Data
Receive order by fax message
Receive order by email message
Receive order by web form
Enter Order Data
Process Order

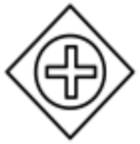
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

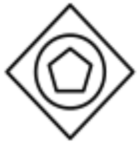
Start Event Gateways



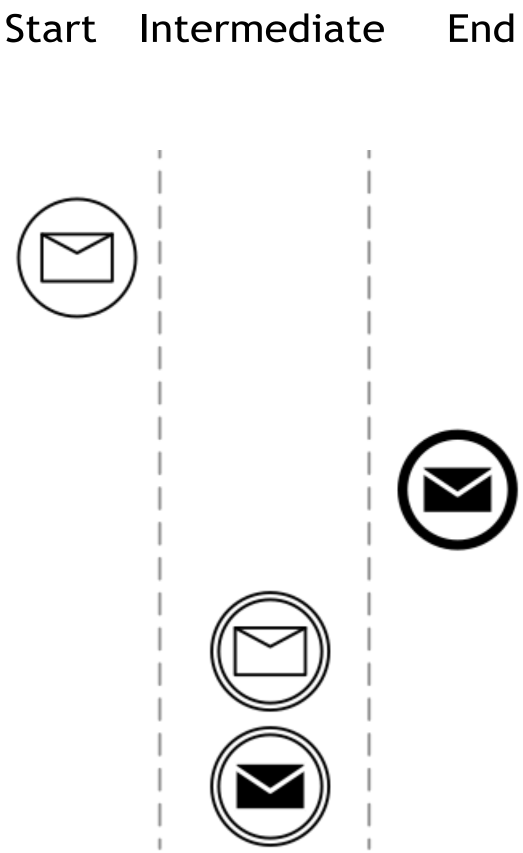





Parallel Event-based Gateway (instantiate)


Exclusive Event-based Gateway (instantiate)

Message Events

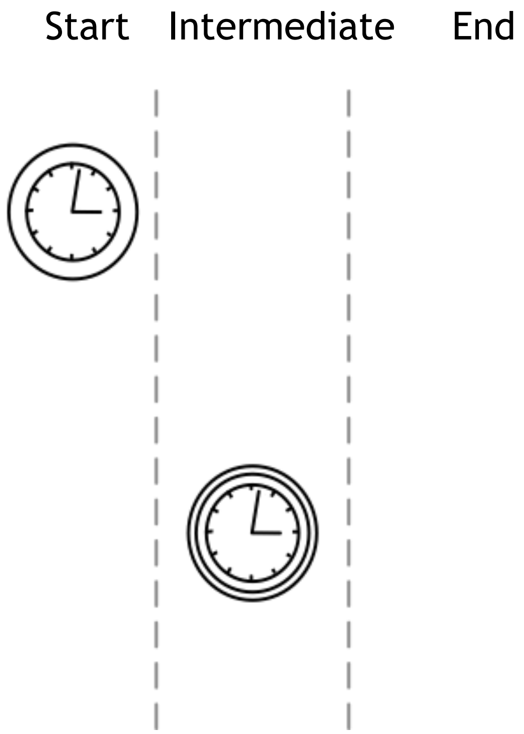


Start Message Event – Indicates that an instance of the process is created when a message is **received**

End Message Event – Indicates that an instance of the process is completed when a message is **sent**

Intermediate Message Event – Indicates that an event is expected to occur during the process. The event is triggered when a message is **received** or **sent**

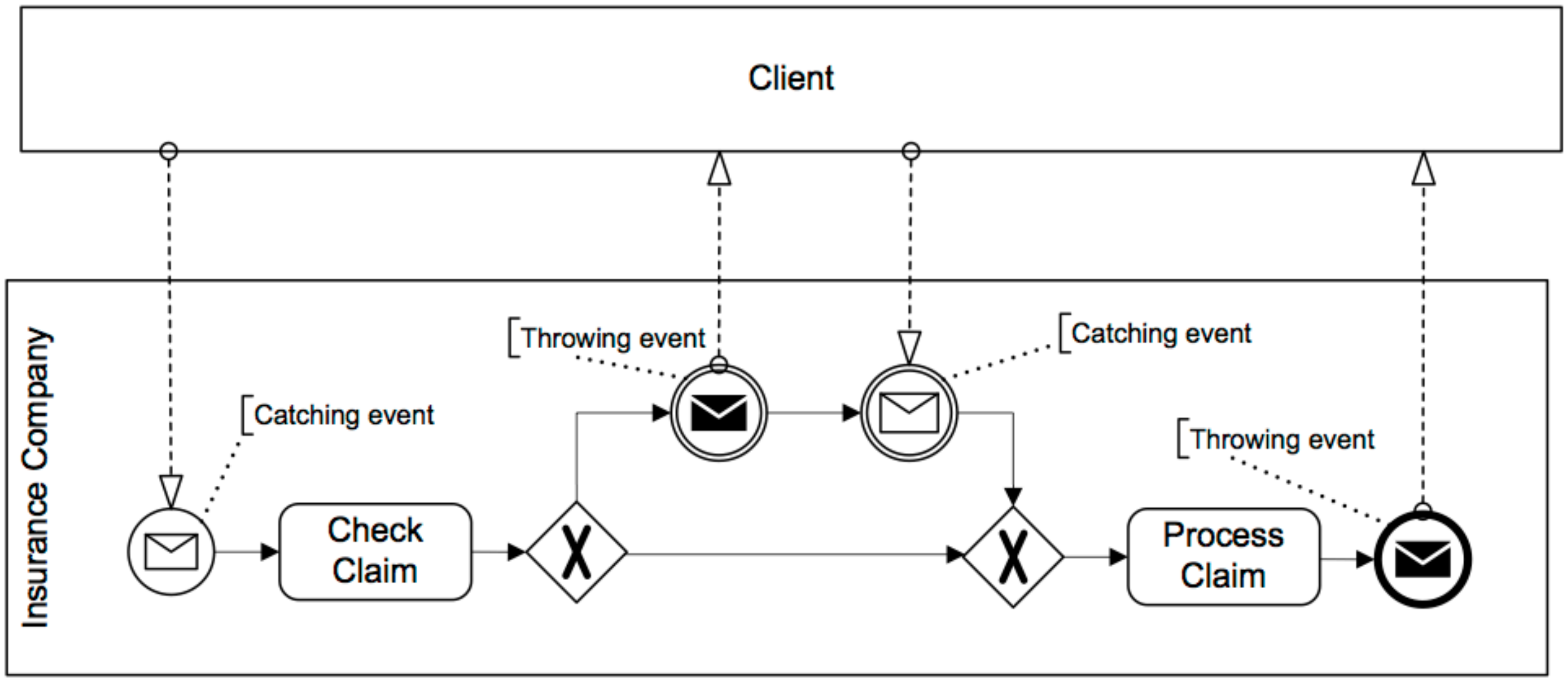
BPMN Events



Start Timer Event – Indicates that an instance of the process is created at certain date(s)/time(s), e.g. start process at 6pm every Friday

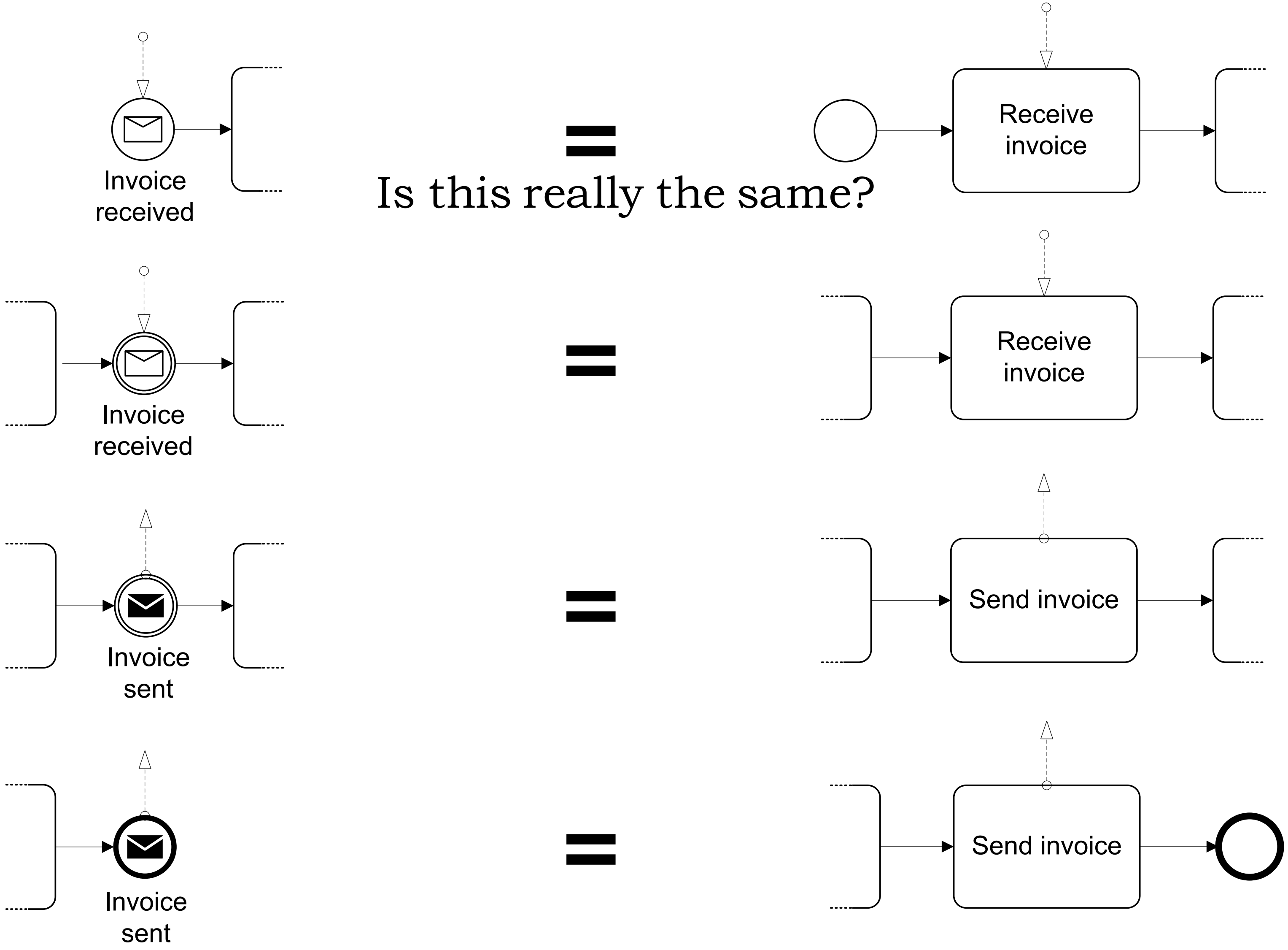
Intermediate Timer Event – Triggered at certain date(s)/time(s), or after a time interval has elapsed since the moment the event is “enabled” (delay)

Message Events: Throwing and Catching Events



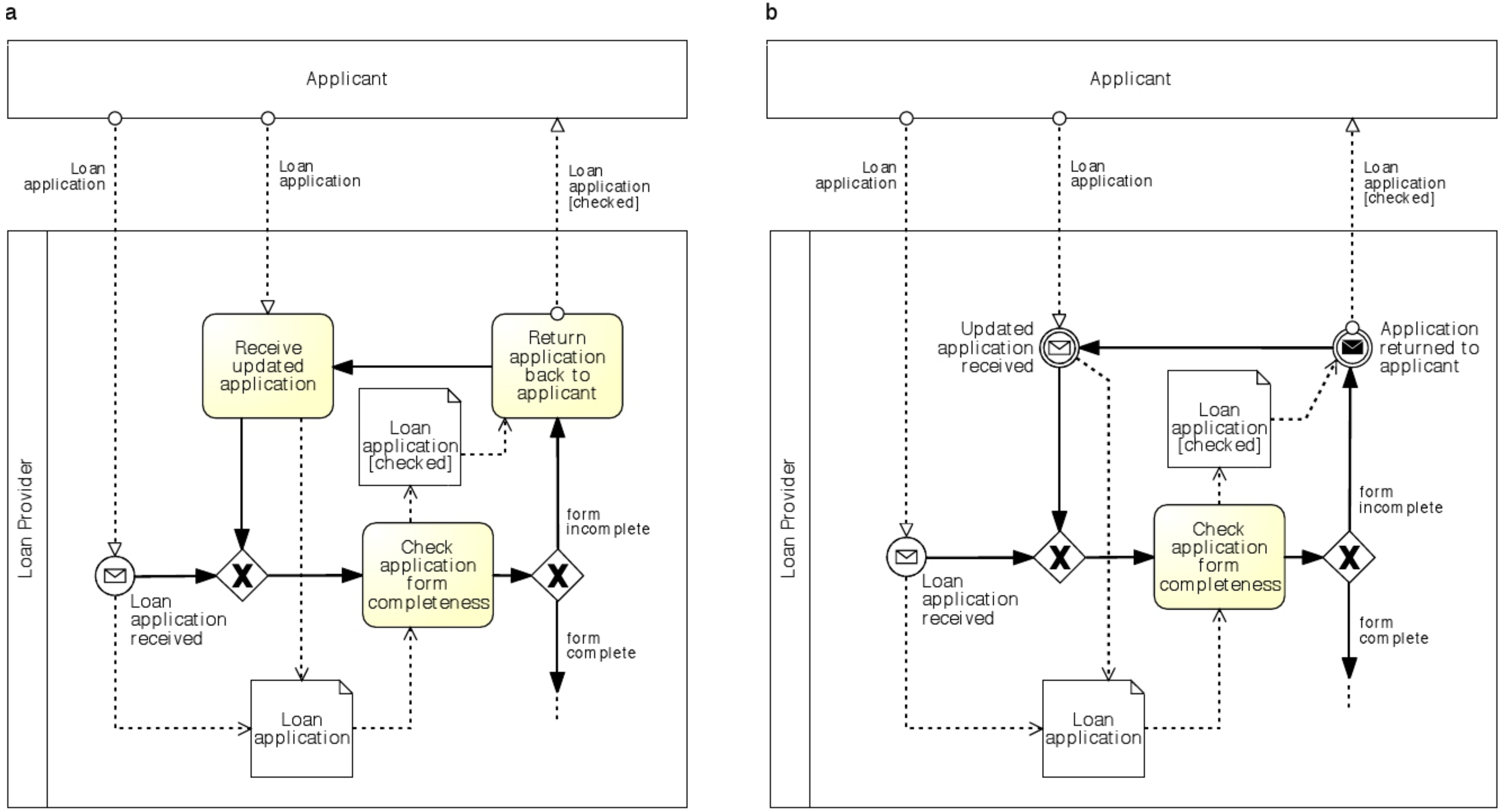
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Comparison with sending/receiving tasks



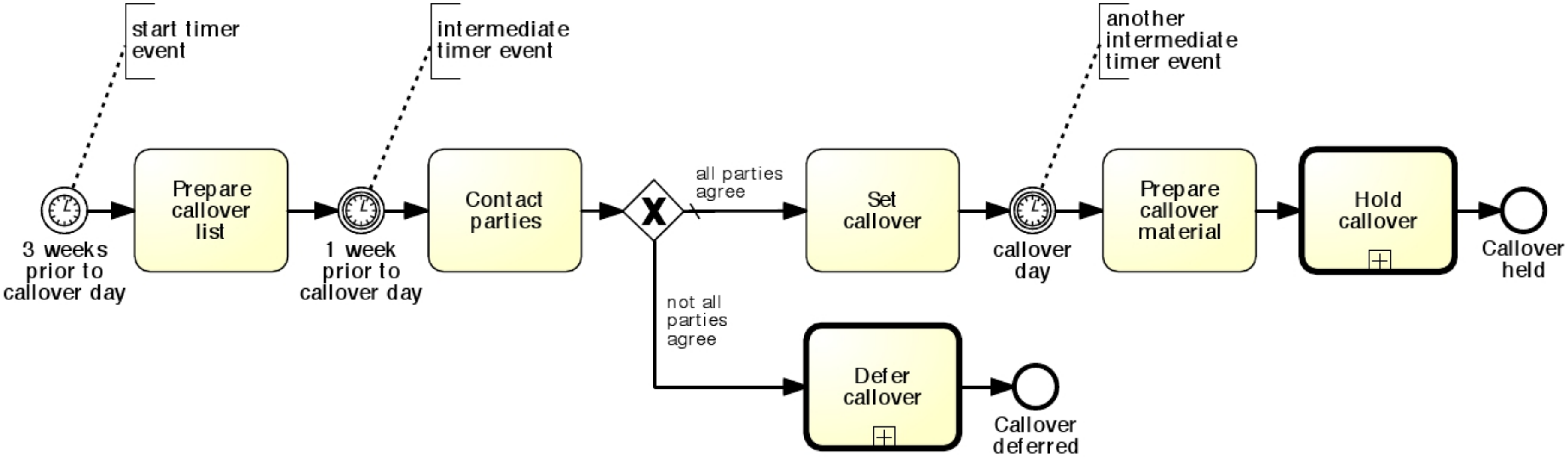
Is this really the same?

When to use what?


a
Applicant
Loan application
Loan application
Loan application [checked]
Loan Provider
Receive updated application
Return application back to applicant
Loan application [checked]
Check application form completeness
form incomplete
form complete
Loan application received
Loan application
b
Applicant
Loan application
Loan application
Loan application [checked]
Loan Provider
Updated application received
Application returned to applicant
Loan application [checked]
Check application form completeness
form incomplete
form complete
Loan application received
Loan application

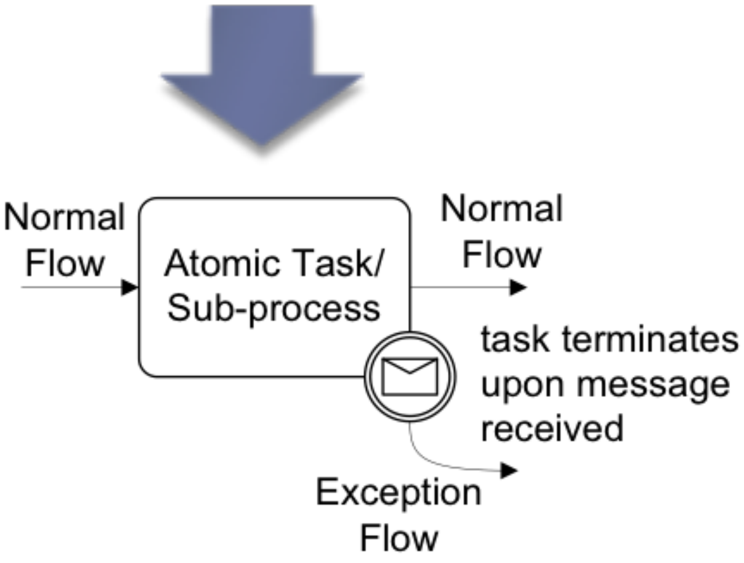
Use message events only when the corresponding activity would simply send or receive a message and do nothing else

Temporal Event and Example

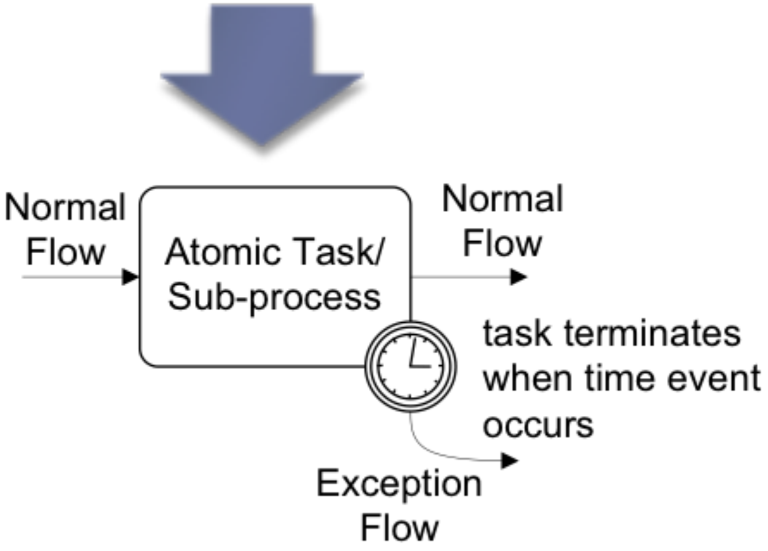


Modelling Exceptional Behaviours

(esternal)



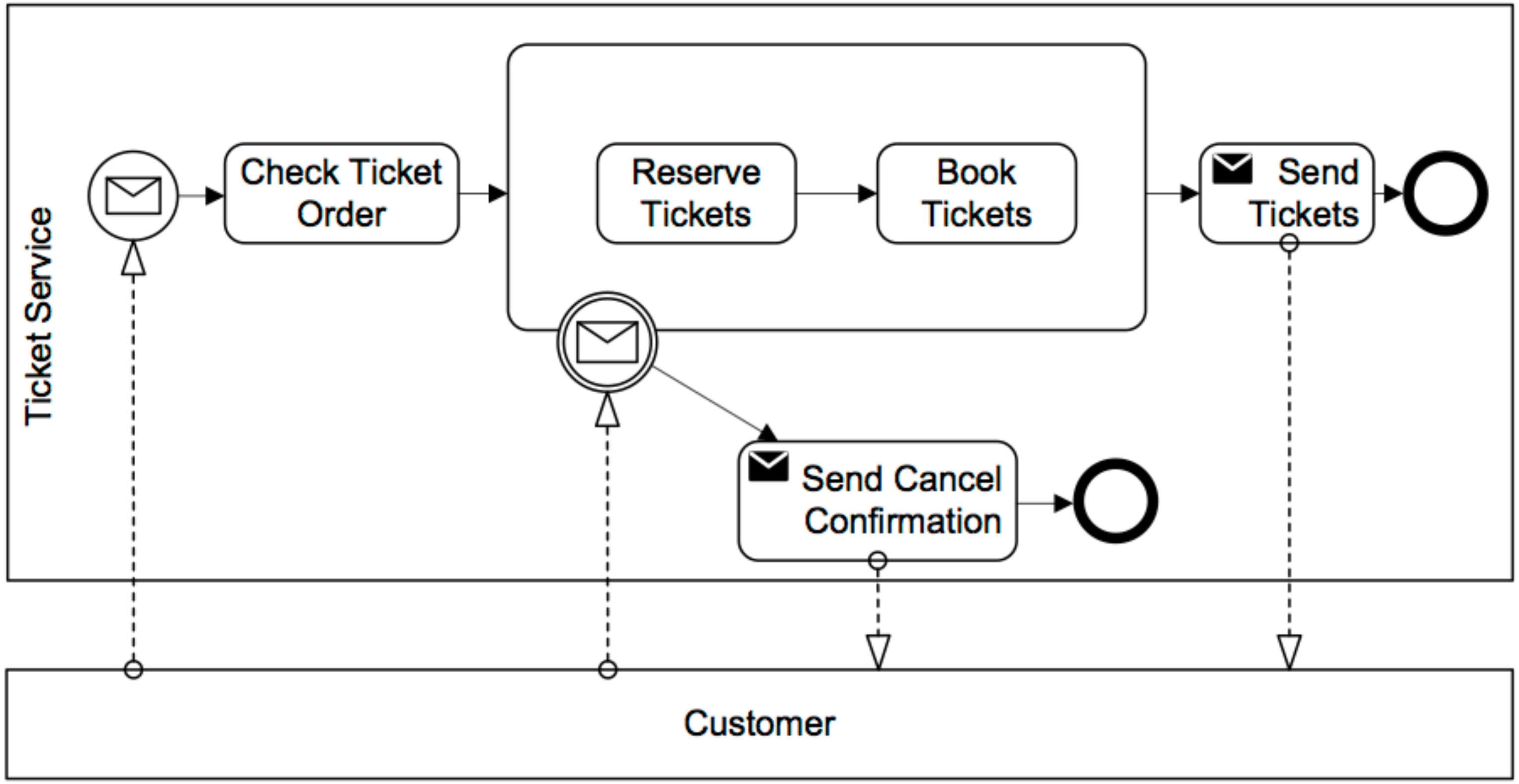
(time dependent)



Both of them can be interrupting or not interrupting

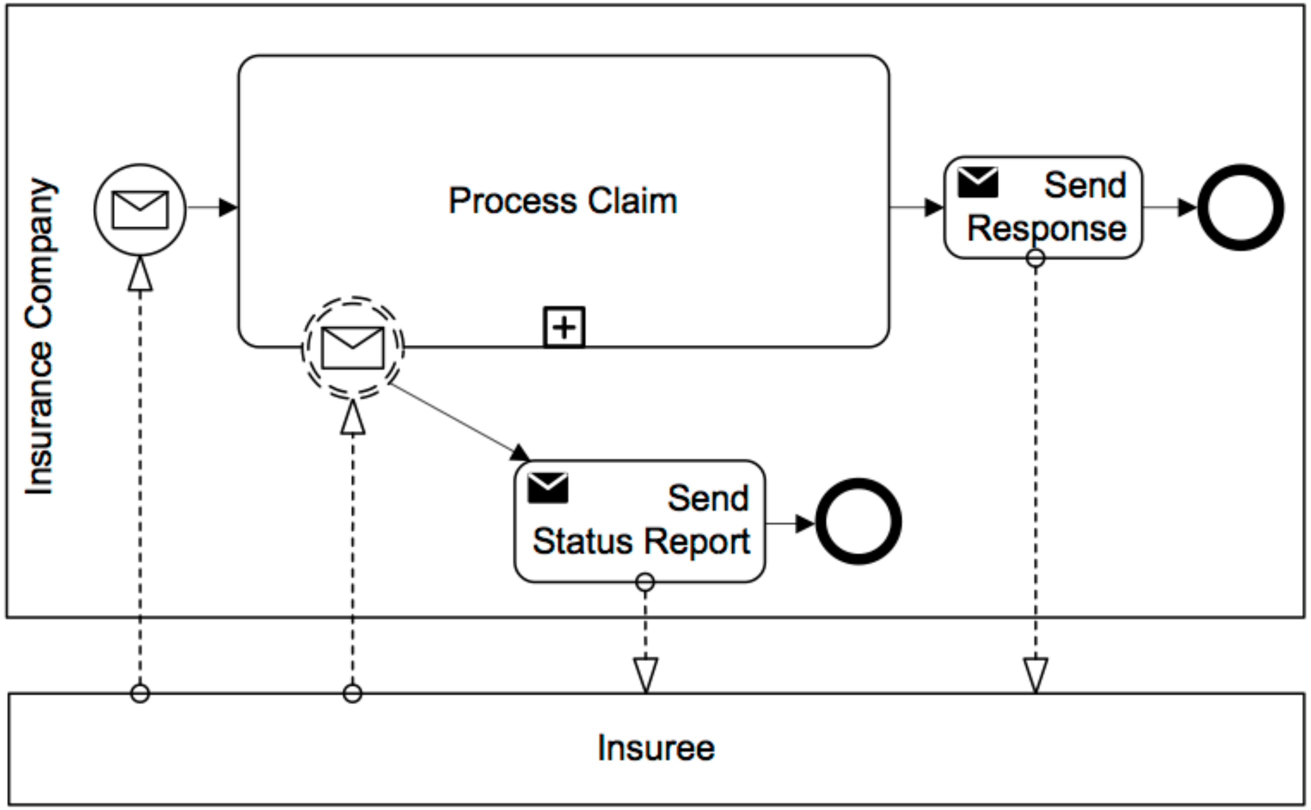
??? (internal) ???

BP with interrupting boundary event



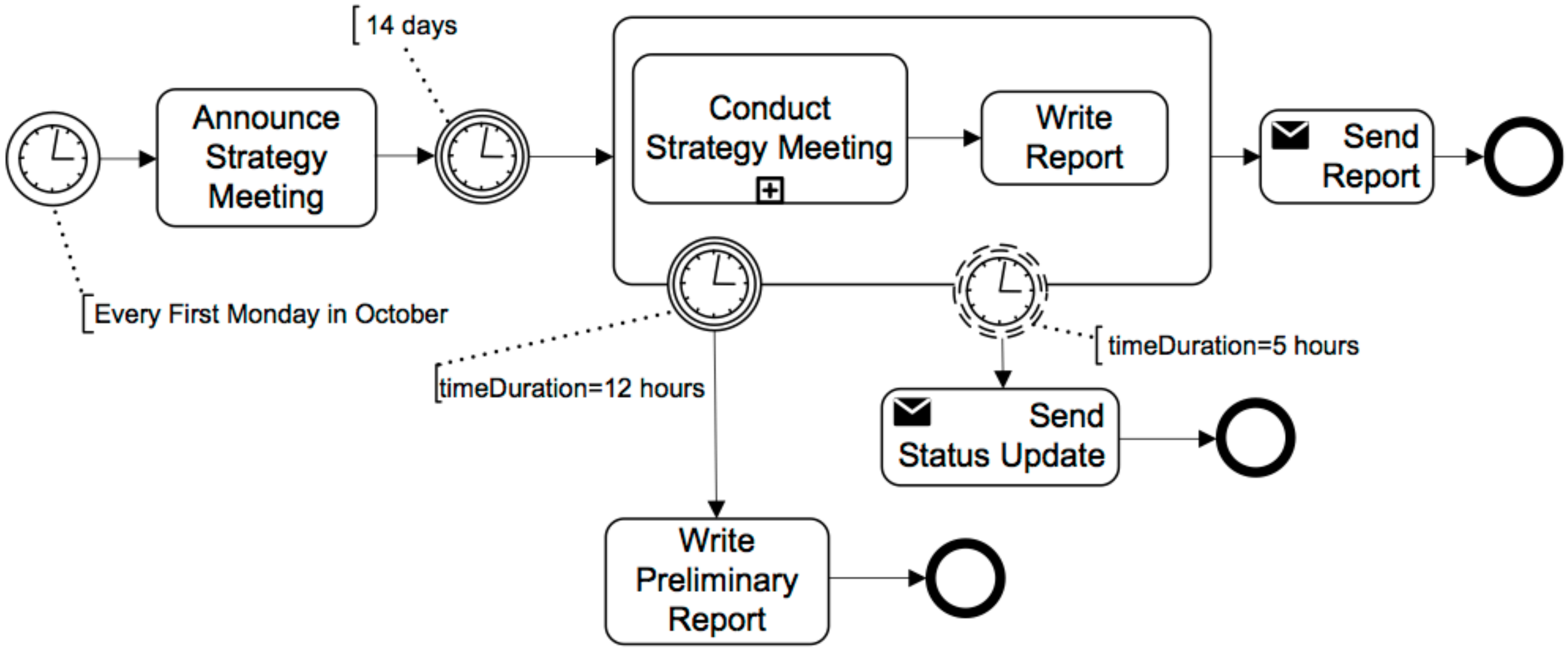
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

BP with non-interrupting boundary event



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

Interrupting and non interrupting boundary time

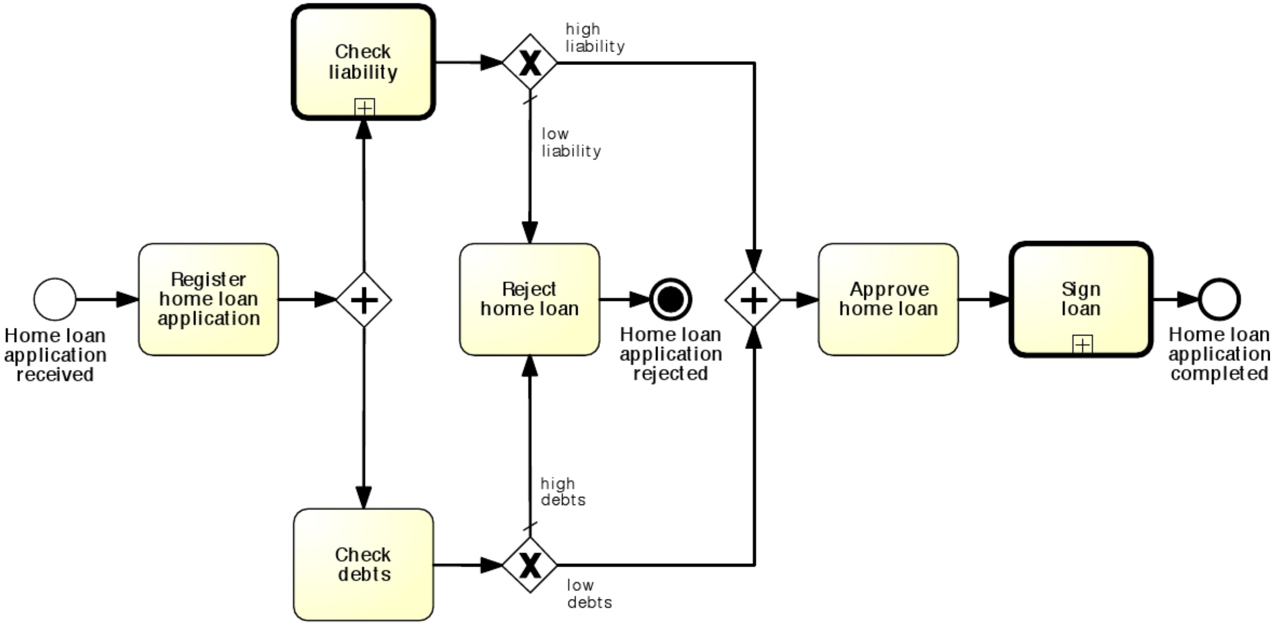


M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Process Abortion

- The simplest way of handling an exception is to abort the running process and signal an improper process termination.

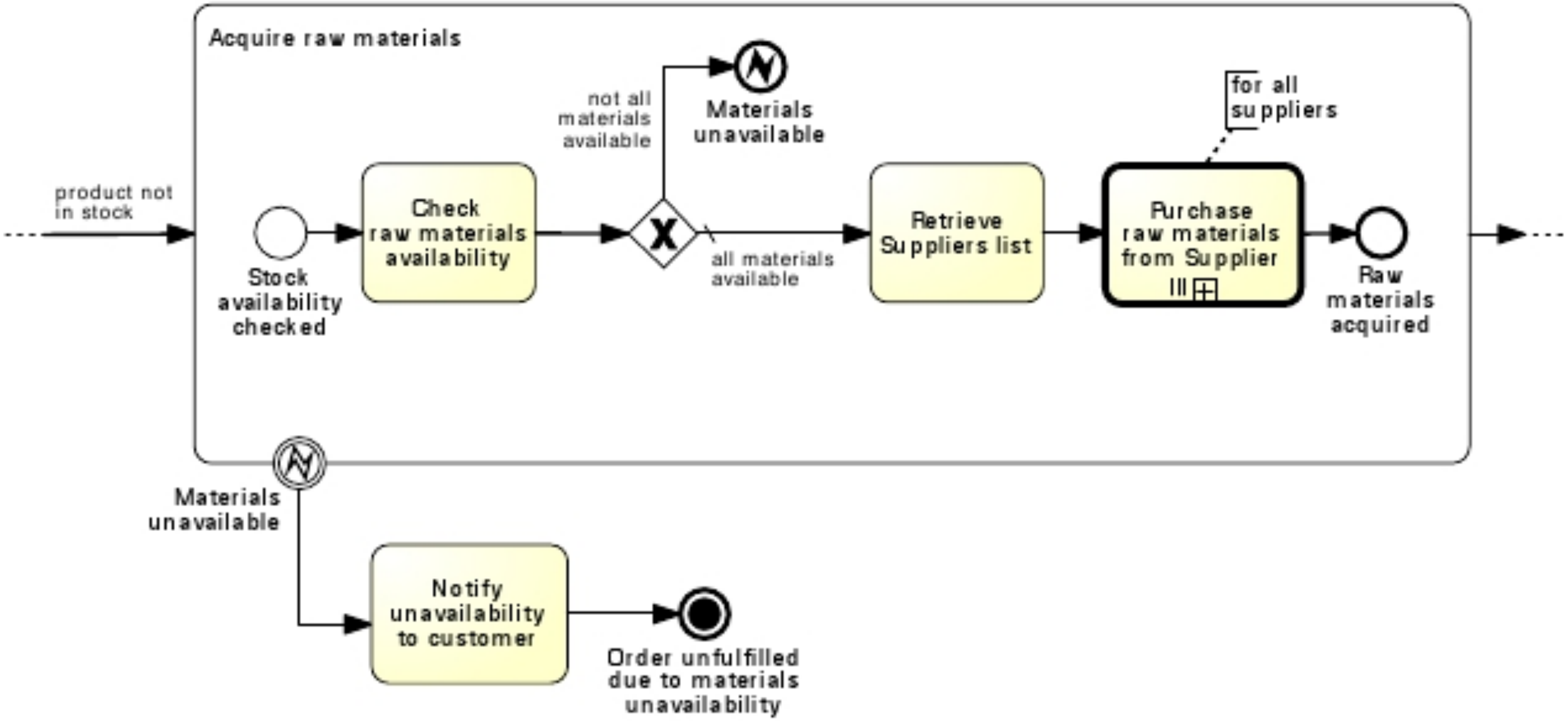
- Observe that if a terminate event is triggered from within a sub-process, it will not cause the abortion of the parent process but only that of the sub-process, i.e. the terminate event is only propagated downwards in a process hierarchy.


Check liability
high liability
low liability
Home loan application received
Register home loan application
Reject home loan
Home loan application rejected
Approve home loan
Sign loan
Home loan application completed
high debts
Check debts
low debts

Exception (internal)

Instead of aborting the whole process, we can handle an exception by interrupting the specific activity that has caused the exception

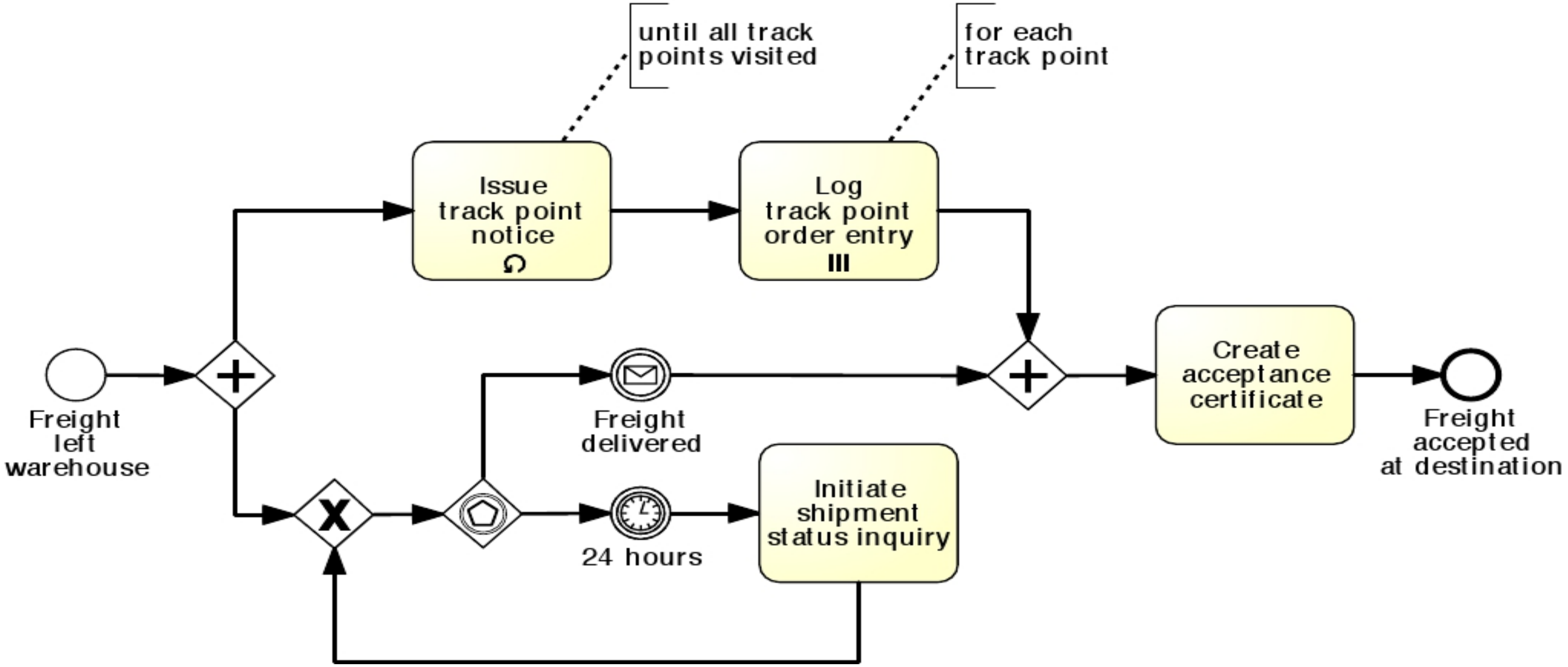
- An end error event is used to interrupt the enclosing sub-process and throw an exception
- This exception is then caught by an intermediate catching error event which is attached to the boundary of the same sub-process
- In turn, this *boundary event* triggers the recovery procedure through an outgoing branch which is called *exception flow*


Acquire raw materials
product not in stock
Stock availability checked
Check raw materials availability
not all materials available
Materials unavailable
all materials available
Retrieve Suppliers list
for all suppliers
Purchase raw materials from Supplier
Raw materials acquired
Materials unavailable
Notify unavailability to customer
Order unfulfilled due to materials unavailability

Racing Events

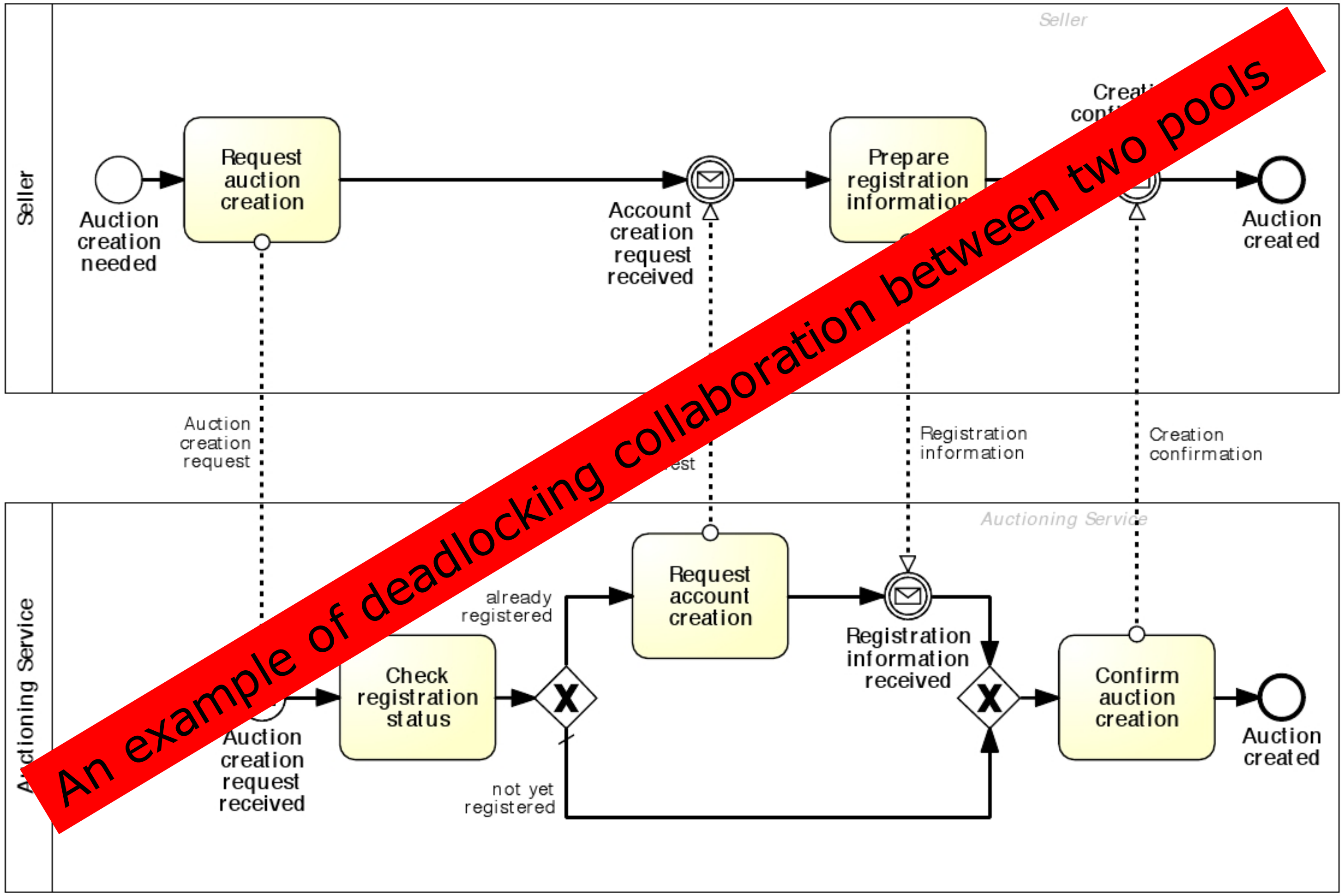
- A typical scenario encountered when modeling processes with events is the one where two external events *race* against one another

- The first of the two events that occurs determines the continuation of the process.

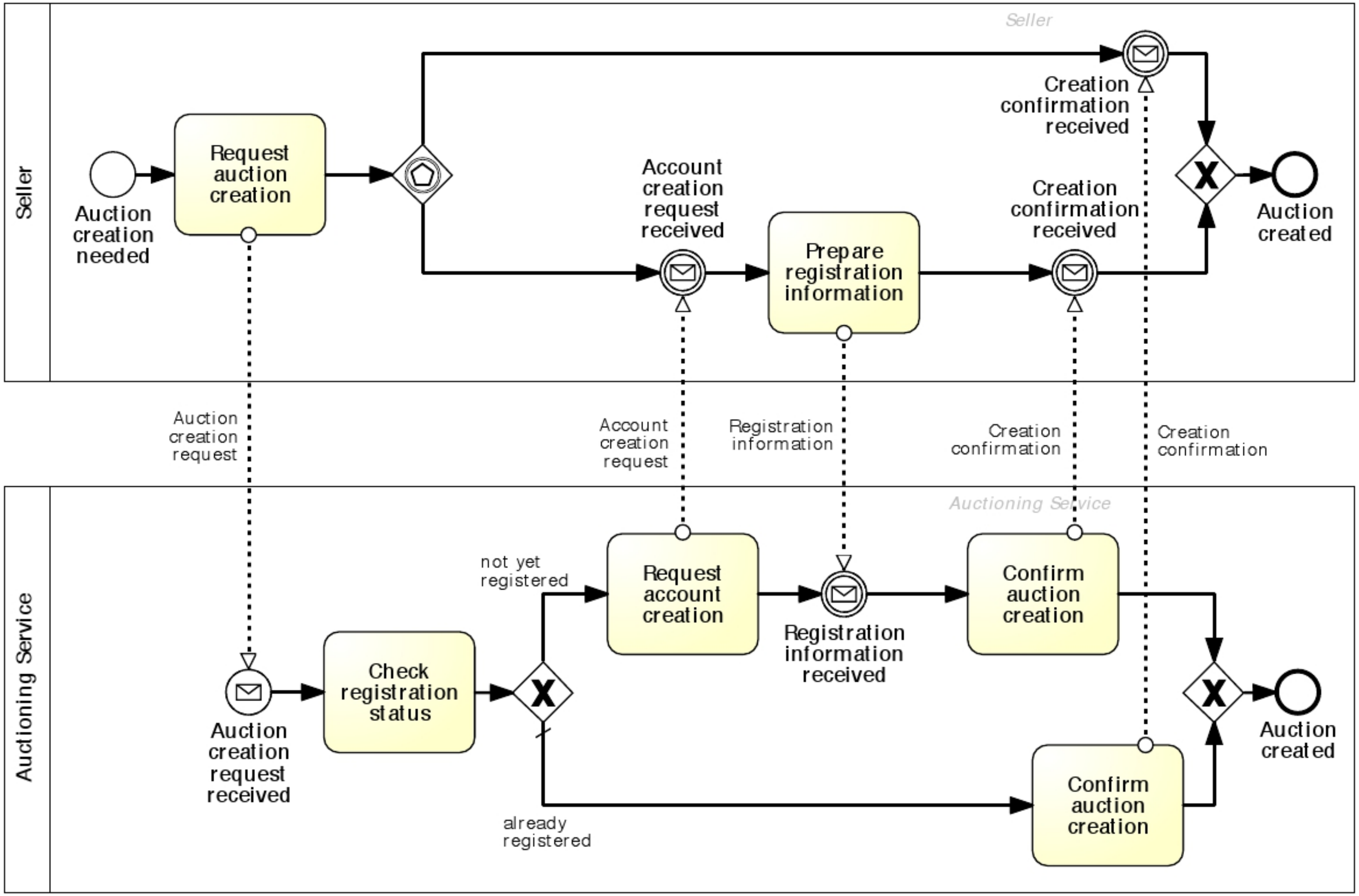

until all track points visited
for each track point
Issue track point notice
Log track point order entry
Freight left warehouse
Freight delivered
24 hours
Initiate shipment status inquiry
Create acceptance certificate
Freight accepted at destination

Racing Event: Event Based Gateway

- Event-based gateways can be used to avoid behavioral anomalies in the communication between pools.

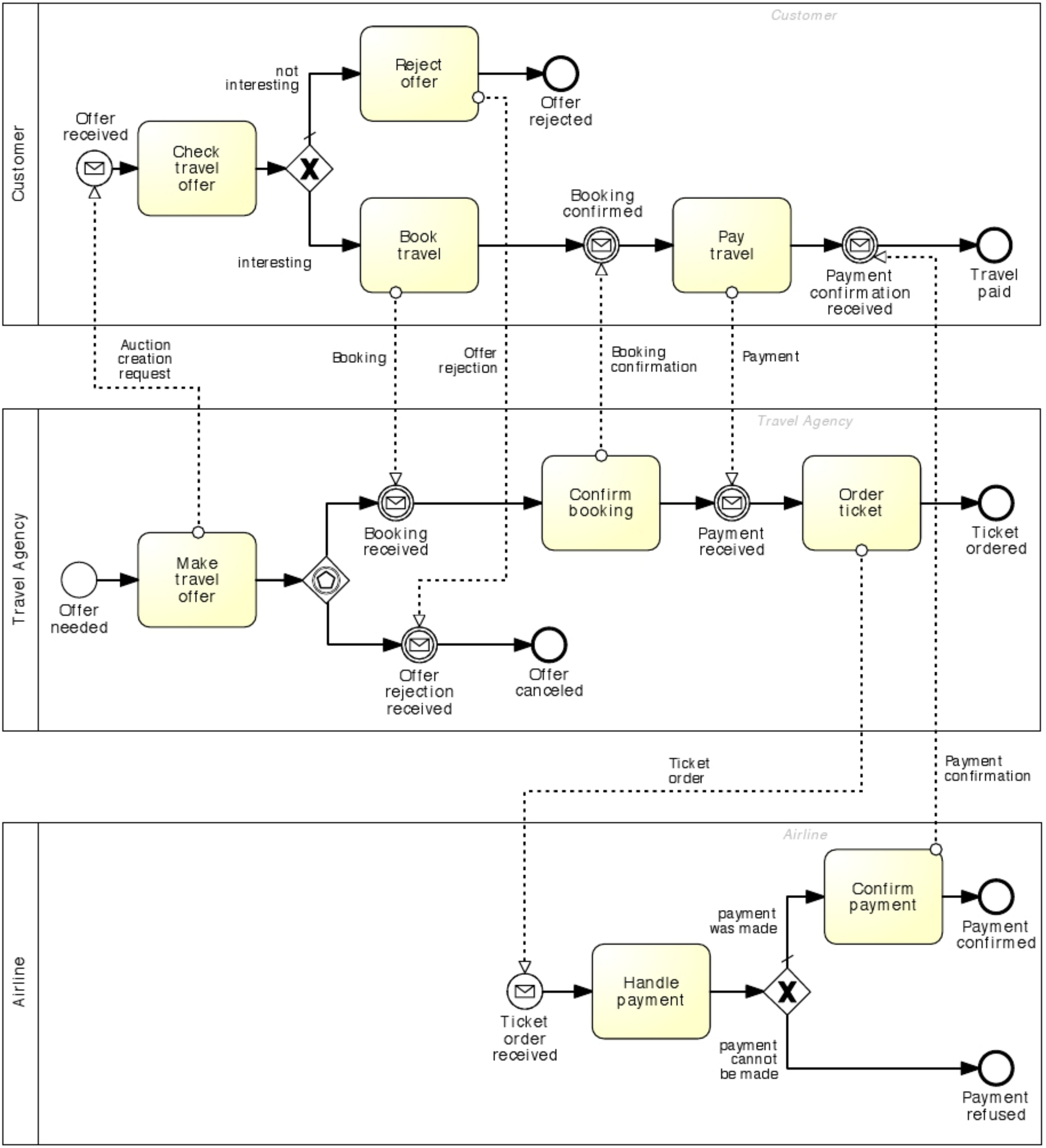


Solution

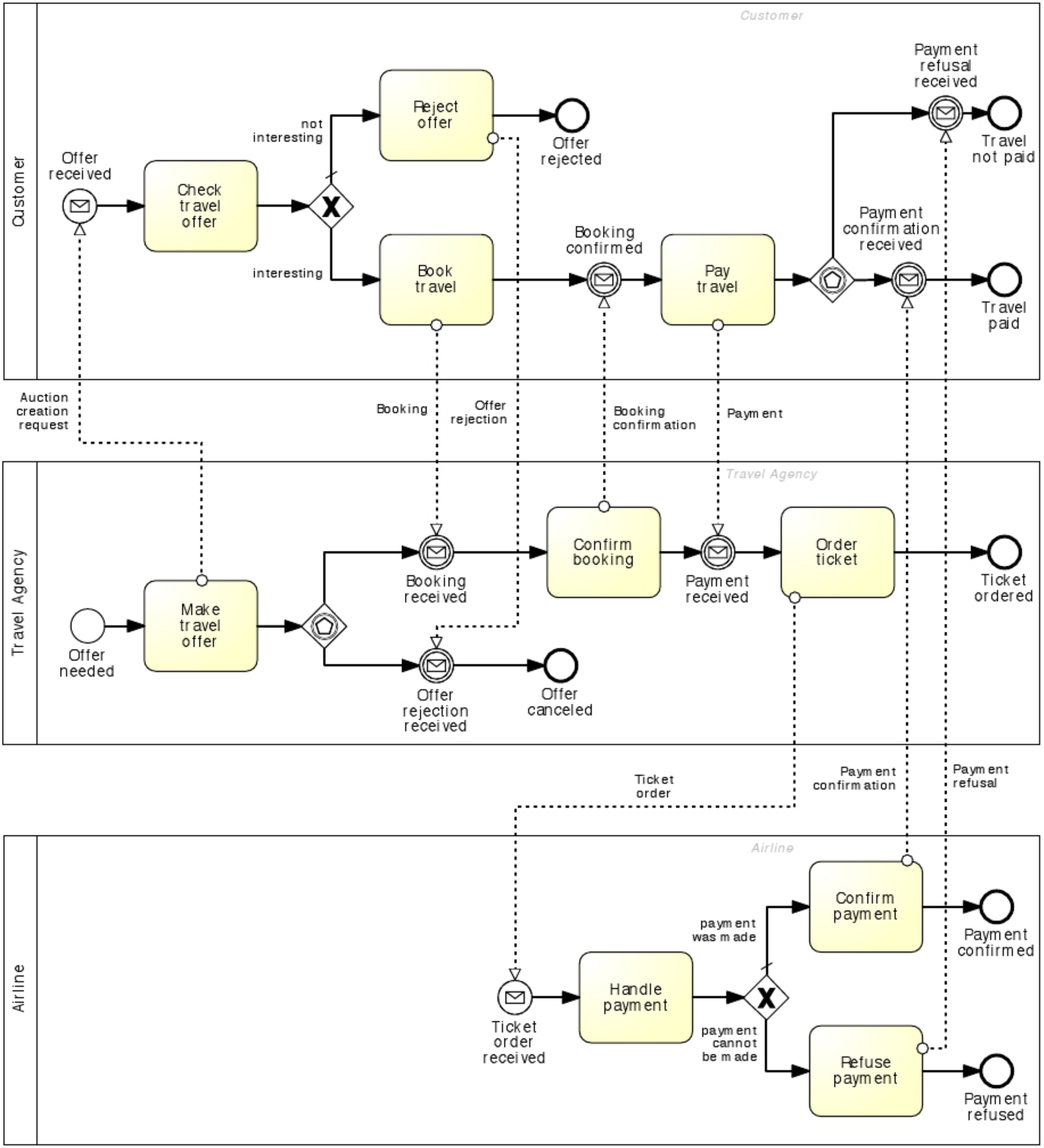

Seller
Seller
Auction creation needed
Request auction creation
Account creation request received
Prepare registration information
Creation confirmation received
Creation confirmation received
Auction created
Auction creation request
Account creation request
Registration information
Creation confirmation
Creation confirmation
Auctioning Service
Auctioning Service
Auction creation request received
Check registration status
not yet registered
Request account creation
Registration information received
Confirm auction creation
already registered
Confirm auction creation
Auction created

Exercise

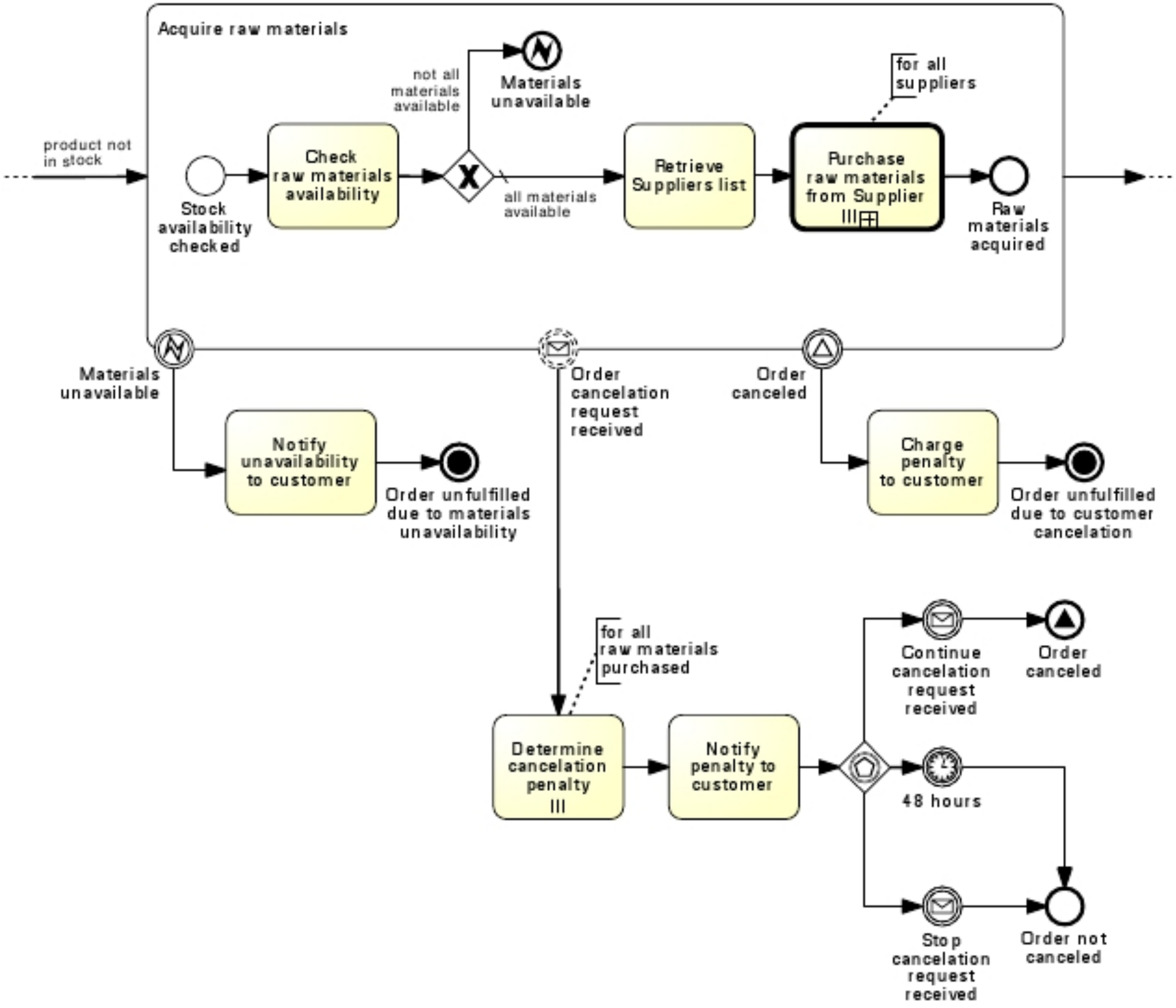
Fix the collaboration diagram



Solution

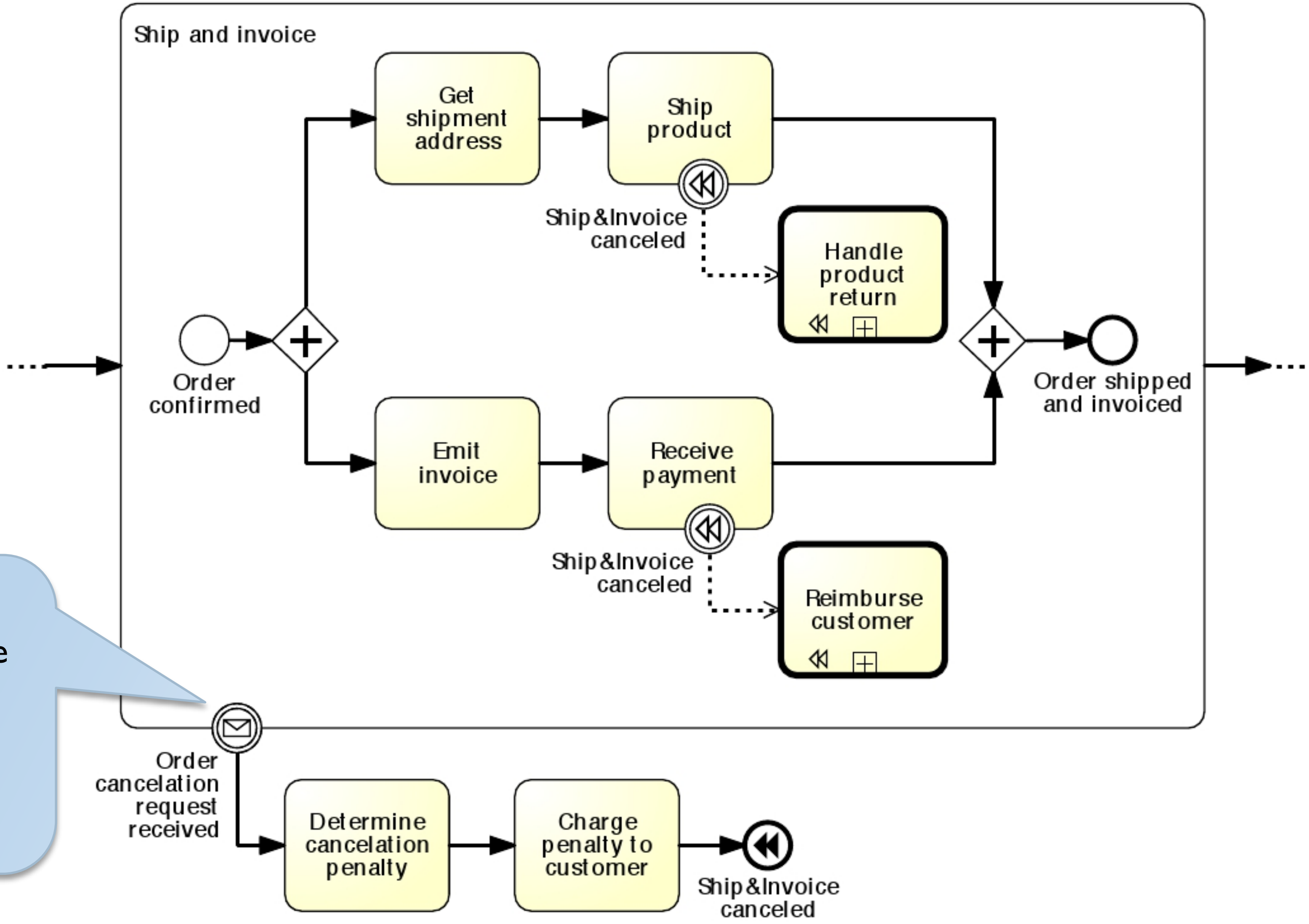

Customer
Customer
Offer received
Check travel offer
not interesting
interesting
Reject offer
Offer rejected
Book travel
Booking confirmed
Pay travel
Payment refusal received
Travel not paid
Payment confirmation received
Travel paid
Auction creation request
Booking
Offer rejection
Booking confirmation
Payment
Travel Agency
Travel Agency
Offer needed
Make travel offer
Booking received
Offer rejection received
Offer canceled
Confirm booking
Payment received
Order ticket
Ticket ordered
Ticket order
Payment confirmation
Payment refusal
Airline
Airline
Ticket order received
Handle payment
payment was made
payment cannot be made
Confirm payment
Payment confirmed
Refuse payment
Payment refused

Events Combination



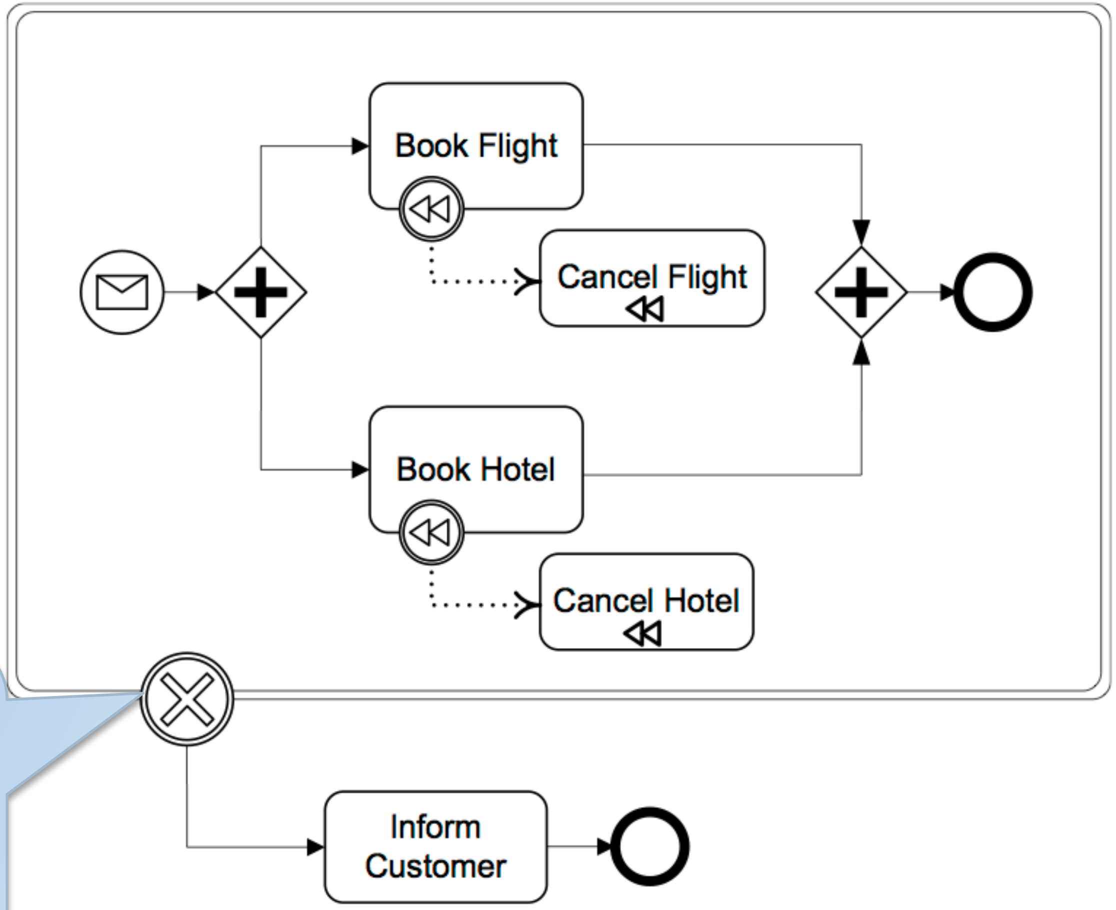
Acquire raw materials
product not in stock
Stock availability checked
Check raw materials availability
not all materials available
Materials unavailable
all materials available
Retrieve Suppliers list
for all suppliers
Purchase raw materials from Supplier
Raw materials acquired
Materials unavailable
Notify unavailability to customer
Order unfulfilled due to materials unavailability
Order cancelation request received
Order canceled
Charge penalty to customer
Order unfulfilled due to customer cancelation
for all raw materials purchased
Determine cancelation penalty
Notify penalty to customer
Continue cancelation request received
Order canceled
48 hours
Stop cancelation request received
Order not canceled

Activity Compensation



As soon as the message is received the compensation is activated.

Transaction and compensation



MORE IN GENERAL

(An event trigger the compensation)

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

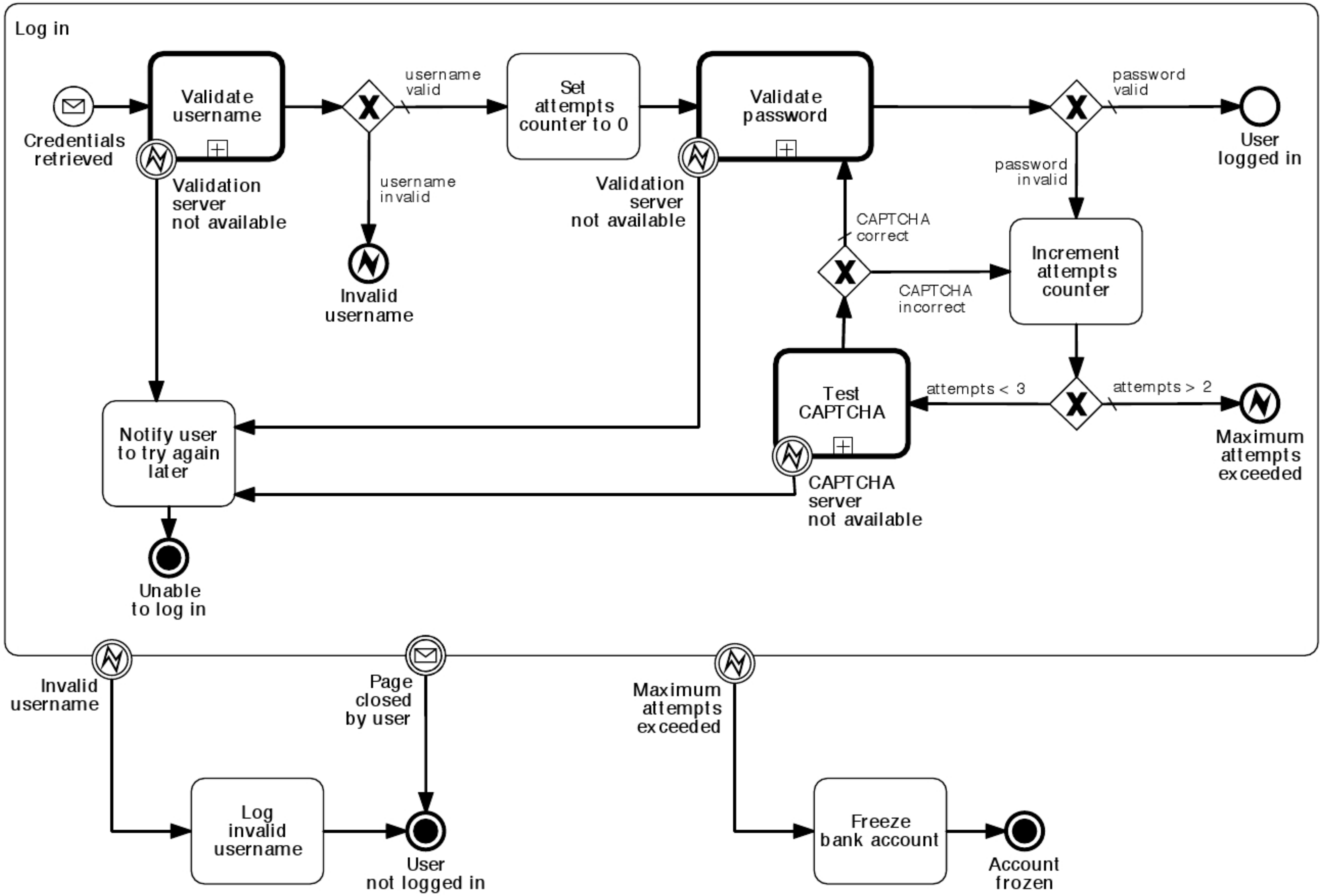
Reccomandations

- Before adding exceptions it is important to understand the sunny-day scenario well
- Start by modeling the sunny day scenario.
- Then think of all possible situations that can go wrong.
  - For each of these exceptions, identify what type of exception handling mechanism needs to be used
    - First, determine the cause of the exception: internal or external
    - Next, decide if aborting the process is enough, or if a recovery procedure needs to be triggered
    - Finally, evaluate whether the interrupted activity needs to be compensated as part of the recovery procedure.

Exercise - Model the following routine for logging into an Internet bank account

The routine for logging into an Internet bank account starts once the credentials entered from the user have been retrieved. First, the username is validated. If the username is not valid, the routine is interrupted and the invalid username is logged. If the username is valid, the number of password trials is set to zero. Then the password is validated. If this is not valid, the counter for the number of trials is incremented and if lower than three, the user is asked to enter the password again, this time together with a CAPTCHA test to increase the security level. If the number of failed attempts reaches three times, the routine is in- terrupted and the account is frozen. Moreover, the username and password validation may be interrupted should the validation server not be available. Similarly, the server to test the CAPTCHA may not be available at the time of log in. In these cases, the procedure is inter- rupted after notifying the user to try again later. At any time during the log in routine, the customer may close the web-page, resulting in the interruption of the routine.

Solution

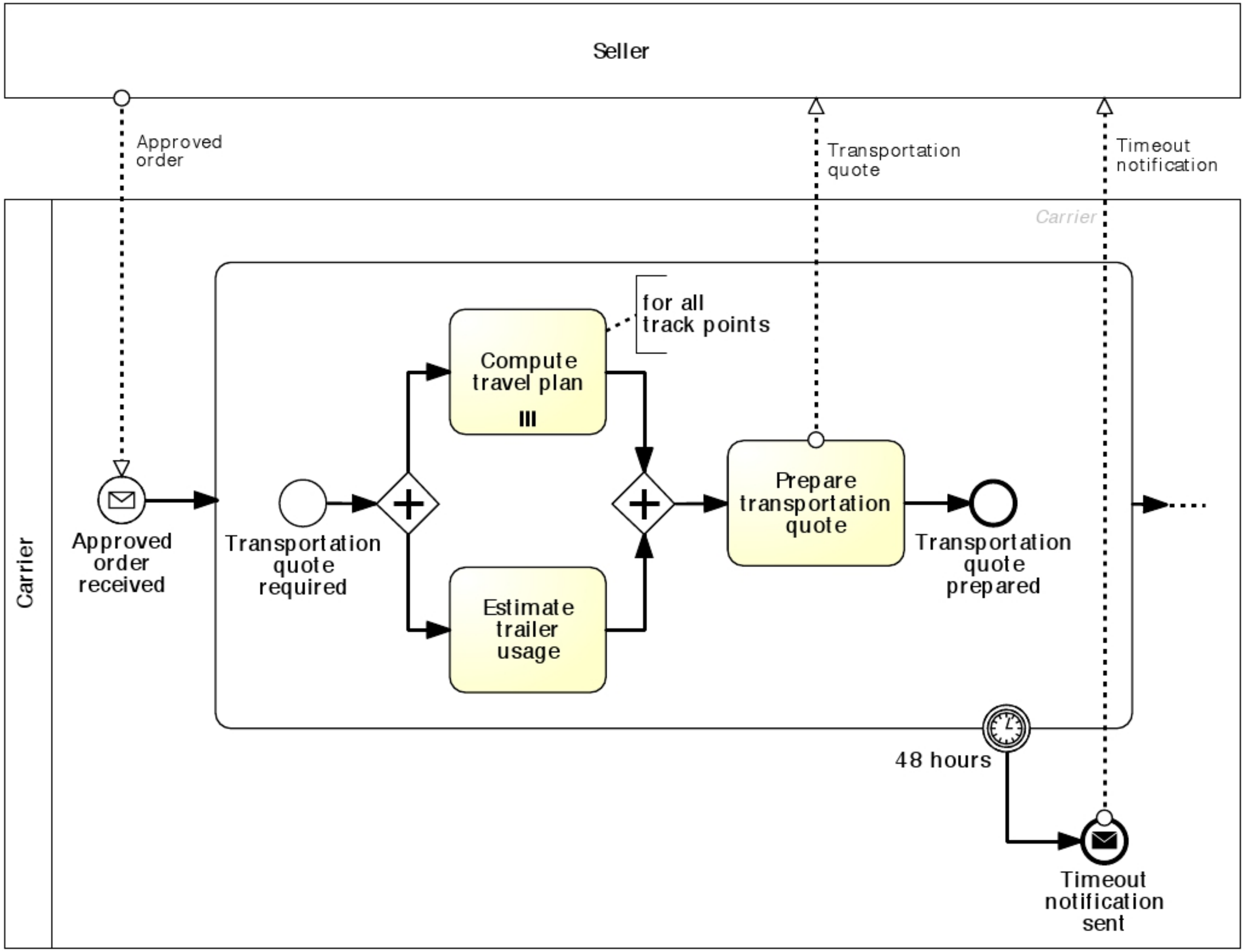


Log in
Credentials retrieved
Validate username
Validation server not available
username valid
username invalid
Invalid username
Set attempts counter to 0
Validate password
Validation server not available
password valid
password invalid
User logged in
Increment attempts counter
CAPTCHA correct
CAPTCHA incorrect
Test CAPTCHA
attempts < 3
attempts > 2
Maximum attempts exceeded
CAPTCHA server not available
Notify user to try again later
Unable to log in
Invalid username
Log invalid username
Page closed by user
User not logged in
Maximum attempts exceeded
Freeze bank account
Account frozen

Exercise - Model the following process fragment

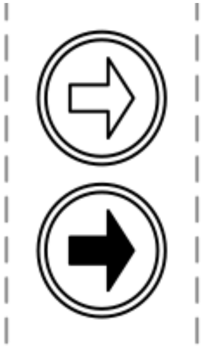
Once a wholesale order has been confirmed, the supplier transmits this order to the car- rier for the preparation of the transportation quote. In order to prepare the quote, the carrier needs to compute the route plan (including all track points that need to be traversed during the travel) and estimate the trailer usage (e.g. whether it is a full track-load, half track-load or a single package). By contract, wholesale orders have to be dispatched within four days from the receipt of the order. This implies that transportation quotes have to be prepared within 48 hours from the receipt of the order to remain within the terms of the contract.

Solution

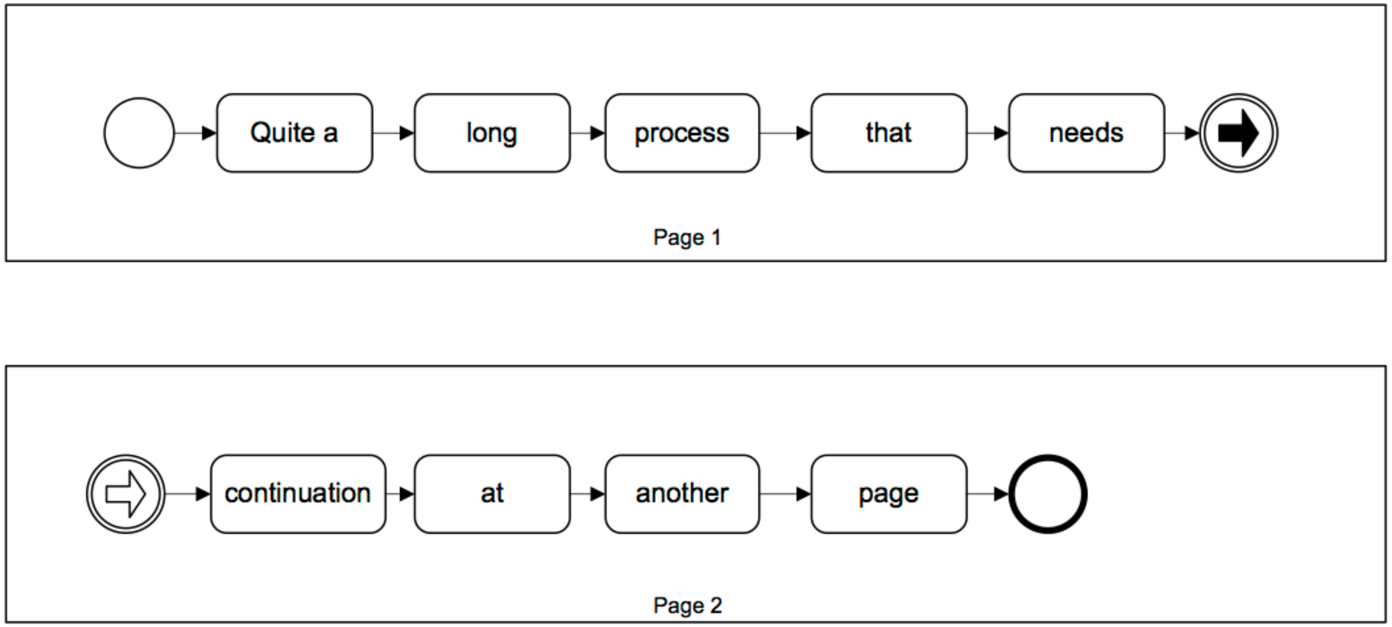

Seller
Approved order
Transportation quote
Timeout notification
Carrier
Carrier
for all track points
Compute travel plan
Approved order received
Transportation quote required
Prepare transportation quote
Transportation quote prepared
Estimate trailer usage
48 hours
Timeout notification sent

Link Events

Start Intermediate End

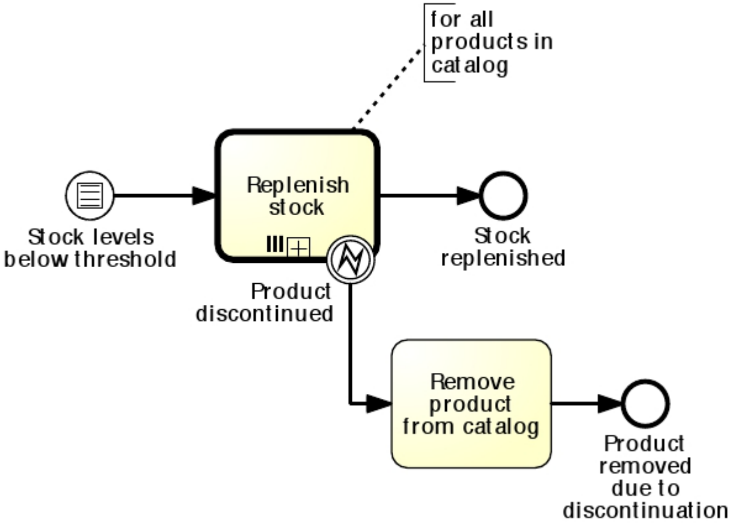


Intermediate Link Event – Indicates that the process flow is resumed from a previous diagram (represented elsewhere), or that it continues in a subsequent diagram (represented elsewhere). Often used as an off-page connector.



Processes and Business Rules

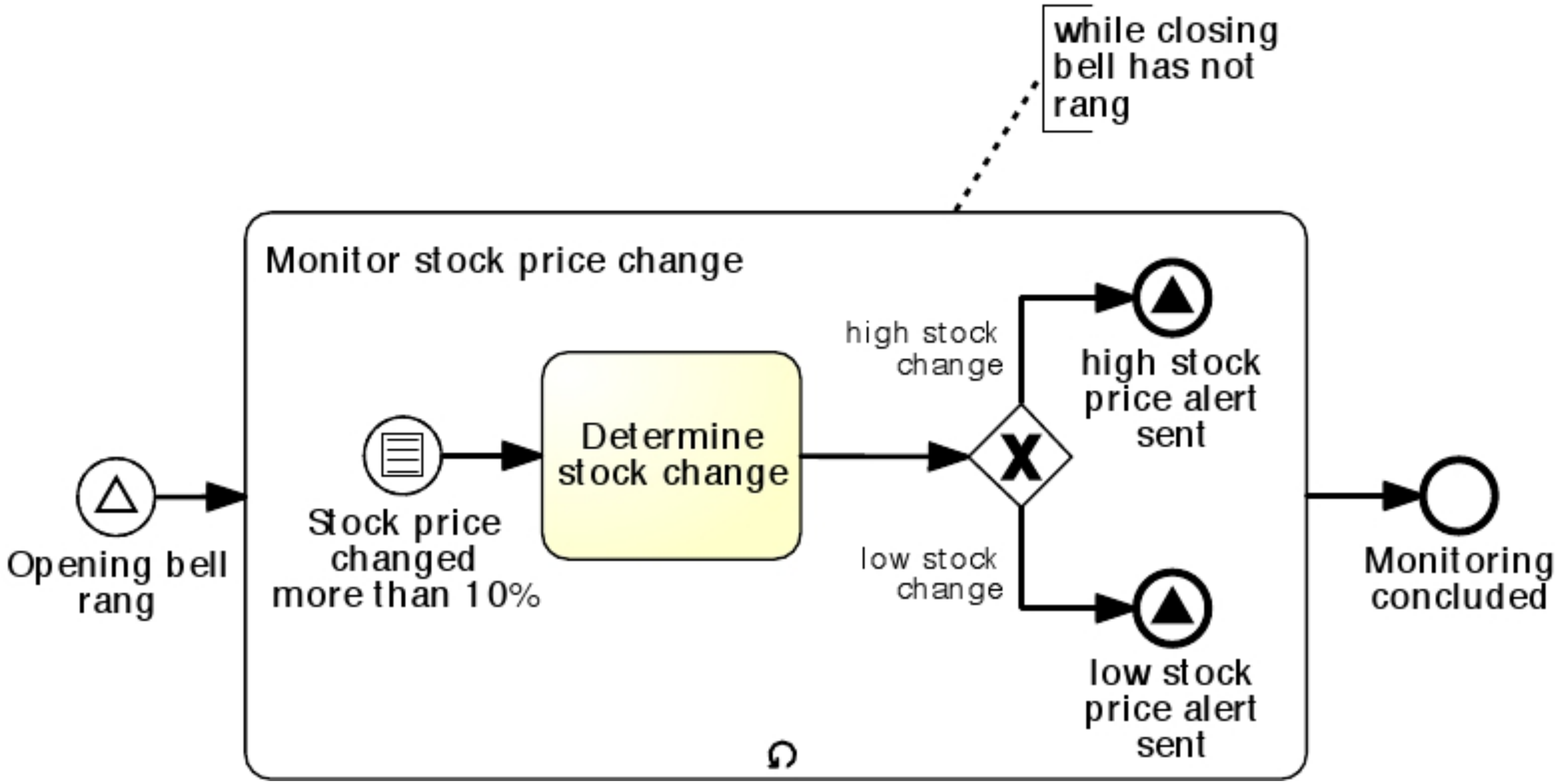
- A business rule implements an organizational policy or practice.
  - For example, in an online shop, platinum customers have a 20 % discount for each purchase above €250.
- Business rules can appear in different forms in a process model.
  - They can be modeled
    - in a decision activity
    - in the condition of a flow coming out of an (X)OR-split
    - **A third option is to use a dedicated BPMN event called *conditional event***


for all products in catalog
Stock levels below threshold
Replenish stock
Stock replenished
Product discontinued
Remove product from catalog
Product removed due to discontinuation

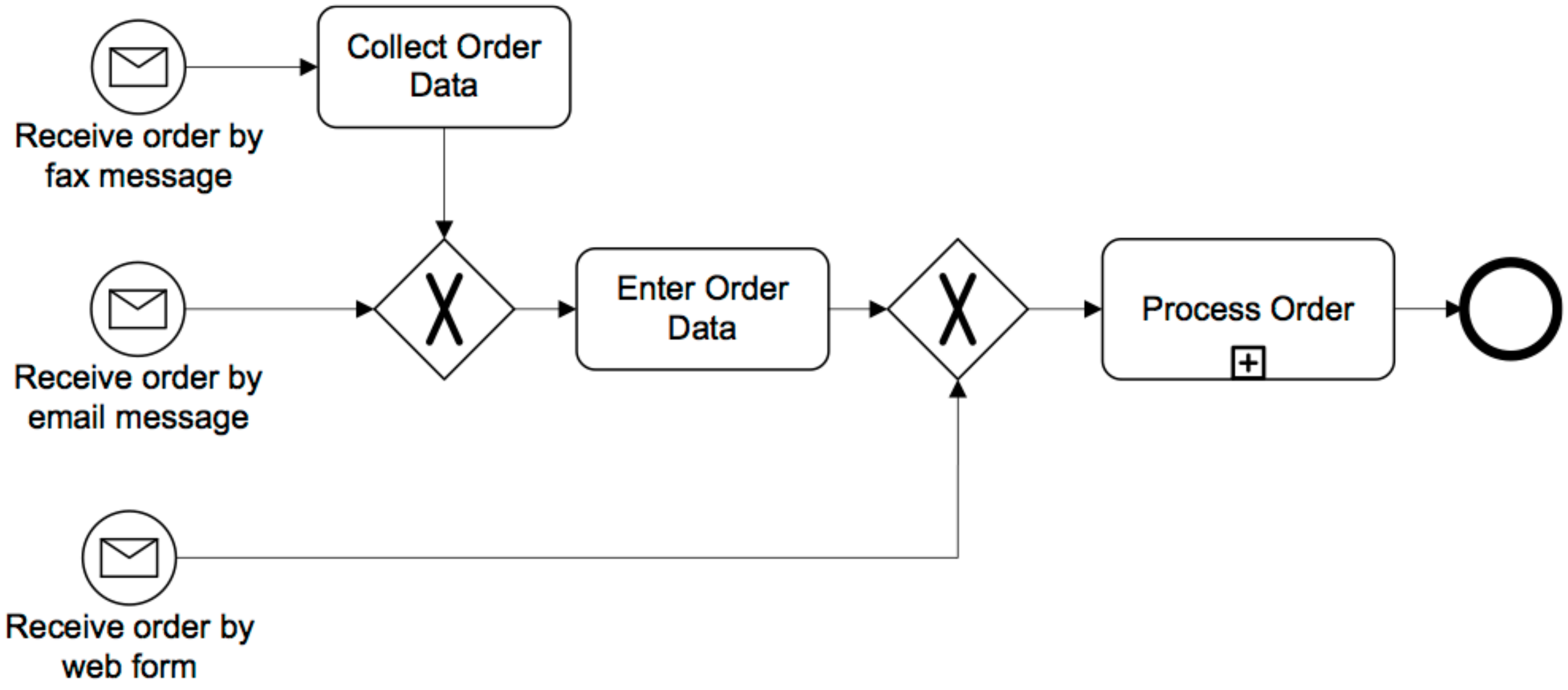
Exercise - Model the following business process snippet

- In a stock exchange, stock price variations are continuously monitored during the day. A day starts when the opening bell rings and concludes when the closing bell rings. Between the two bells, every time the stock price changes by more than 10 %, the entity of the change is first determined. Next, if the change is high, a "high stock price" alert is sent, otherwise a "low stock price" alert is sent.

Solution

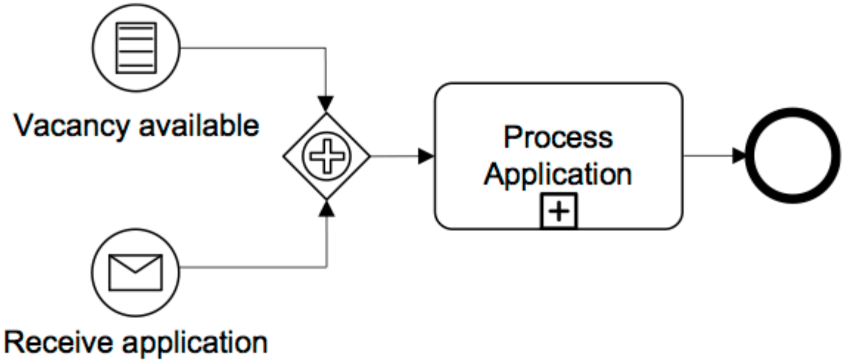

while closing bell has not rang
Monitor stock price change
Opening bell rang
Stock price changed more than 10%
Determine stock change
high stock change
low stock change
high stock price alert sent
low stock price alert sent
Monitoring concluded

Multiple alternative start events


Collect Order Data
Receive order by fax message
Enter Order Data
Process Order
Receive order by email message
Receive order by web form

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

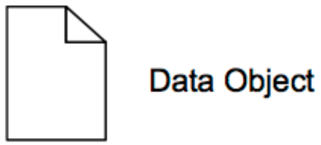
Multiple Start Events



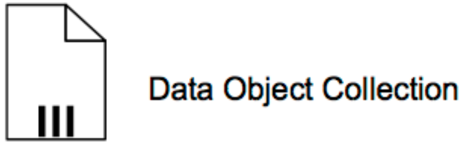
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Both need to occur to instantiate the process

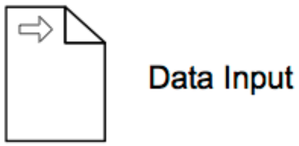
Information Artifacts



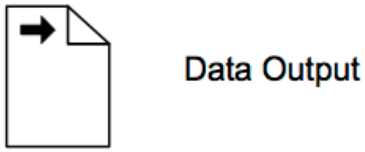
- **Data Objects** are a mechanism to show how data is required or produced by activities. Represent input and output of a process activity.



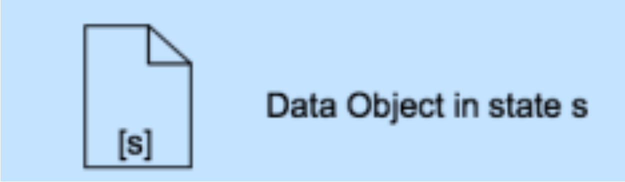
- **A Collection Data Object** represents a collection of information, e.g., a list of order items.



- **A Data Input** is an external input for the entire process. A kind of input parameter.



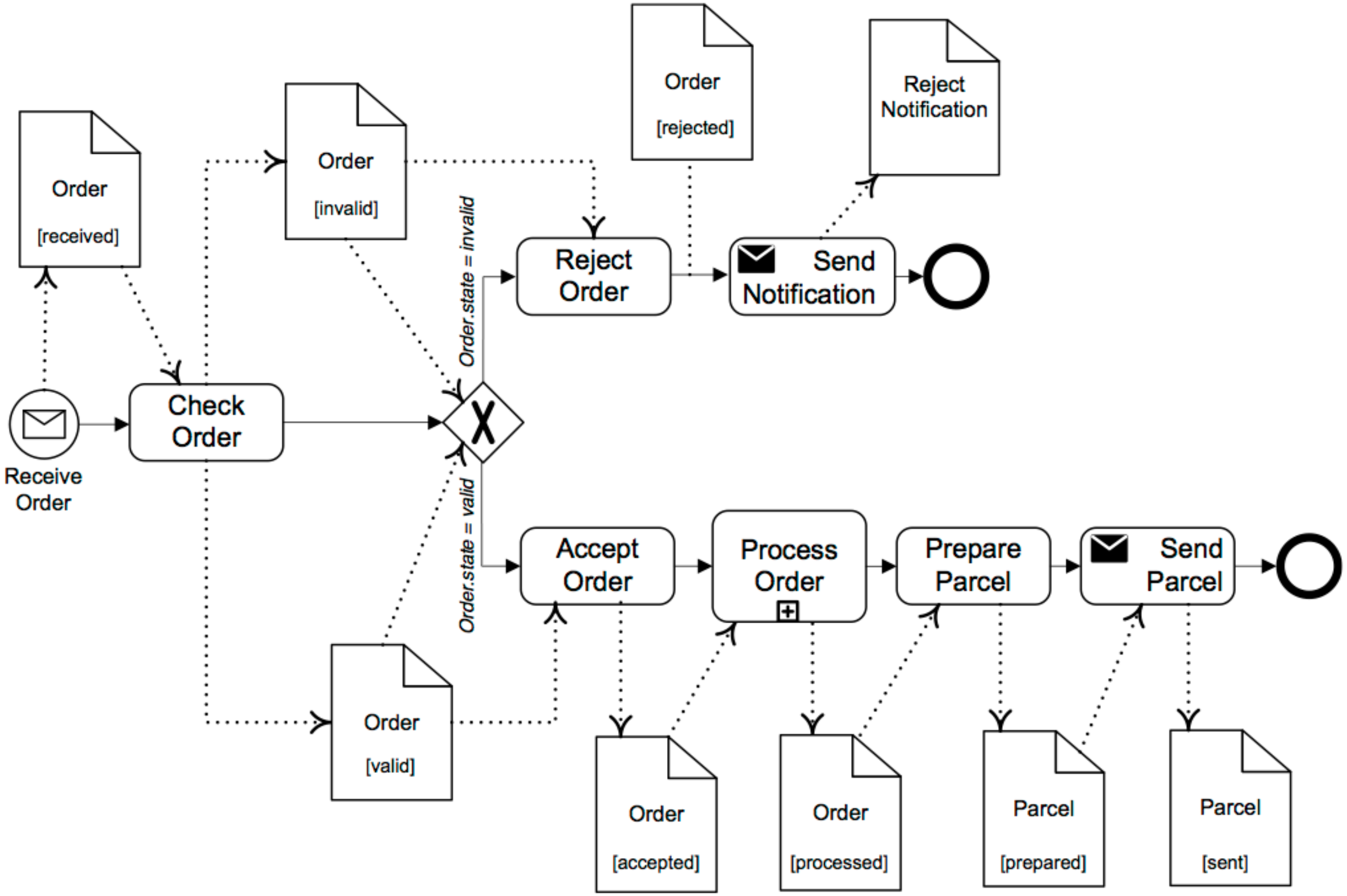
- **A Data Output** is data result of the entire process. A kind of output parameter.



Do data objects affect the token flow?

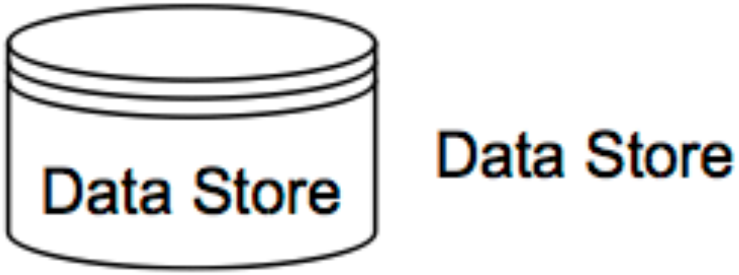
- Input data objects are required for an activity to be executed.
- Even if a token is available on the incoming arc of that activity, the latter cannot be executed until all input data objects are also available.
- A data object is available if it has been created as a result of completing a preceding activity (whose output was the data object itself), or because it is an input to the whole process (like Purchase order).
- Output data objects only affect the token flow indirectly, i.e. when they are used by subsequent activities.

Process diagram involving data-object



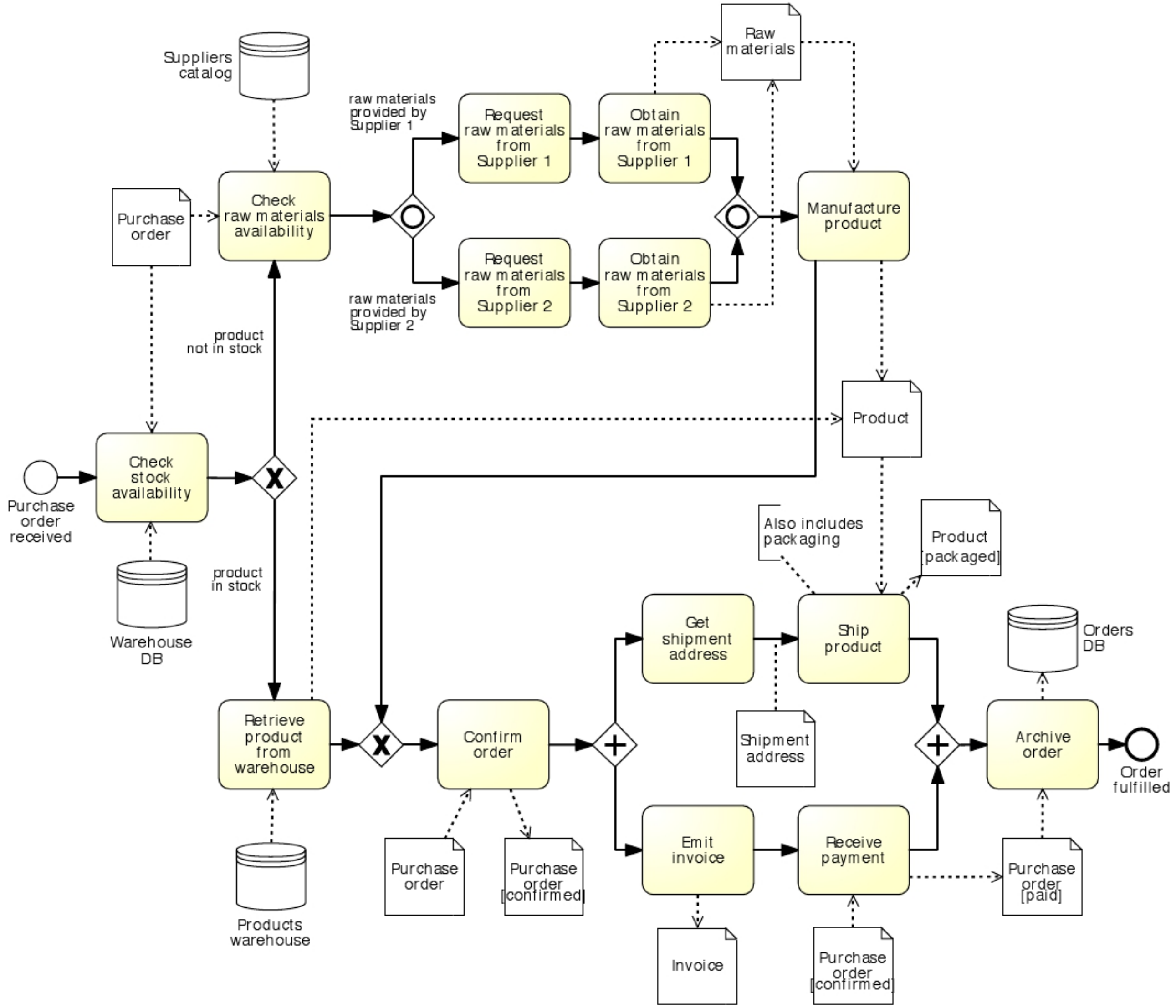
M. Weske: Business Process Management,
© Springer-Verlag Berlin Heidelberg 2012, 2007

Data Store



- **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

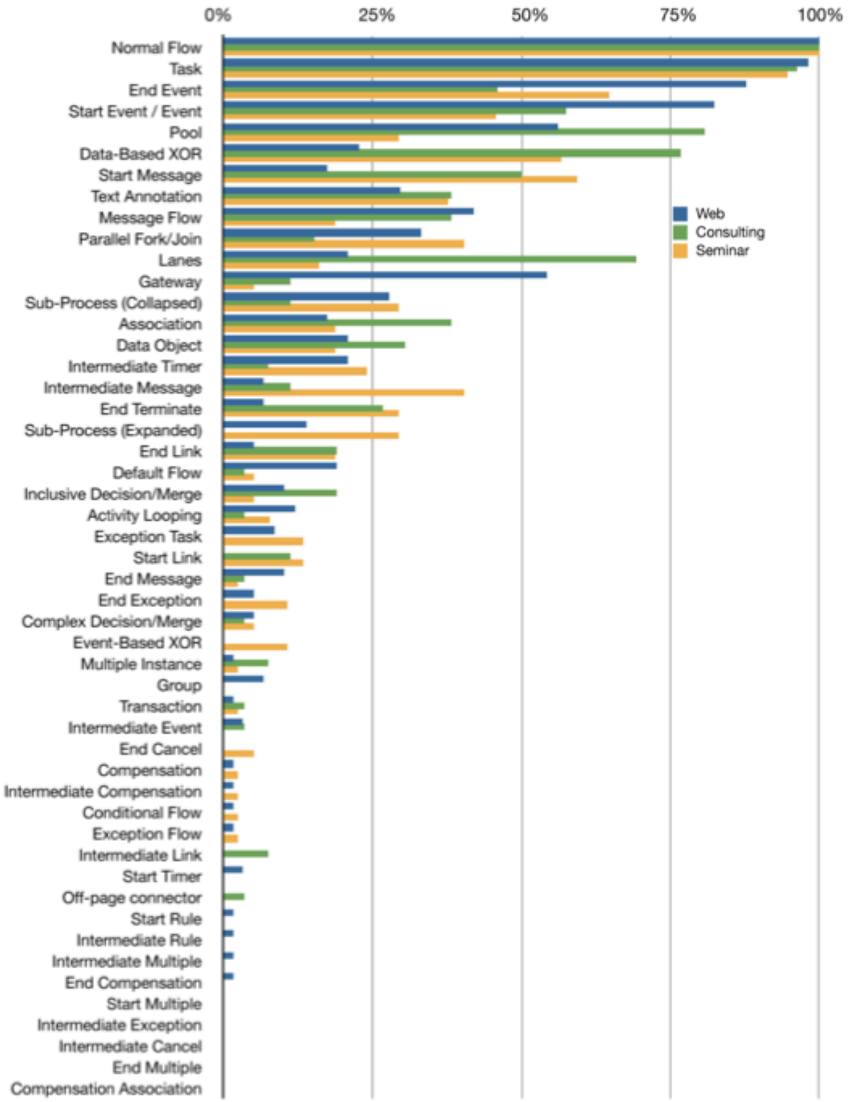
Process Diagram Involving Information Artifact



Do we always need to model data objects?

- Data objects help the reader understand the flow of business data from one activity to the other
- **However**, the price to pay is an increased complexity of the diagram
- **Use them only when they are needed for a specific purpose**, e.g. to highlight potential issues in the process under analysis or for automation

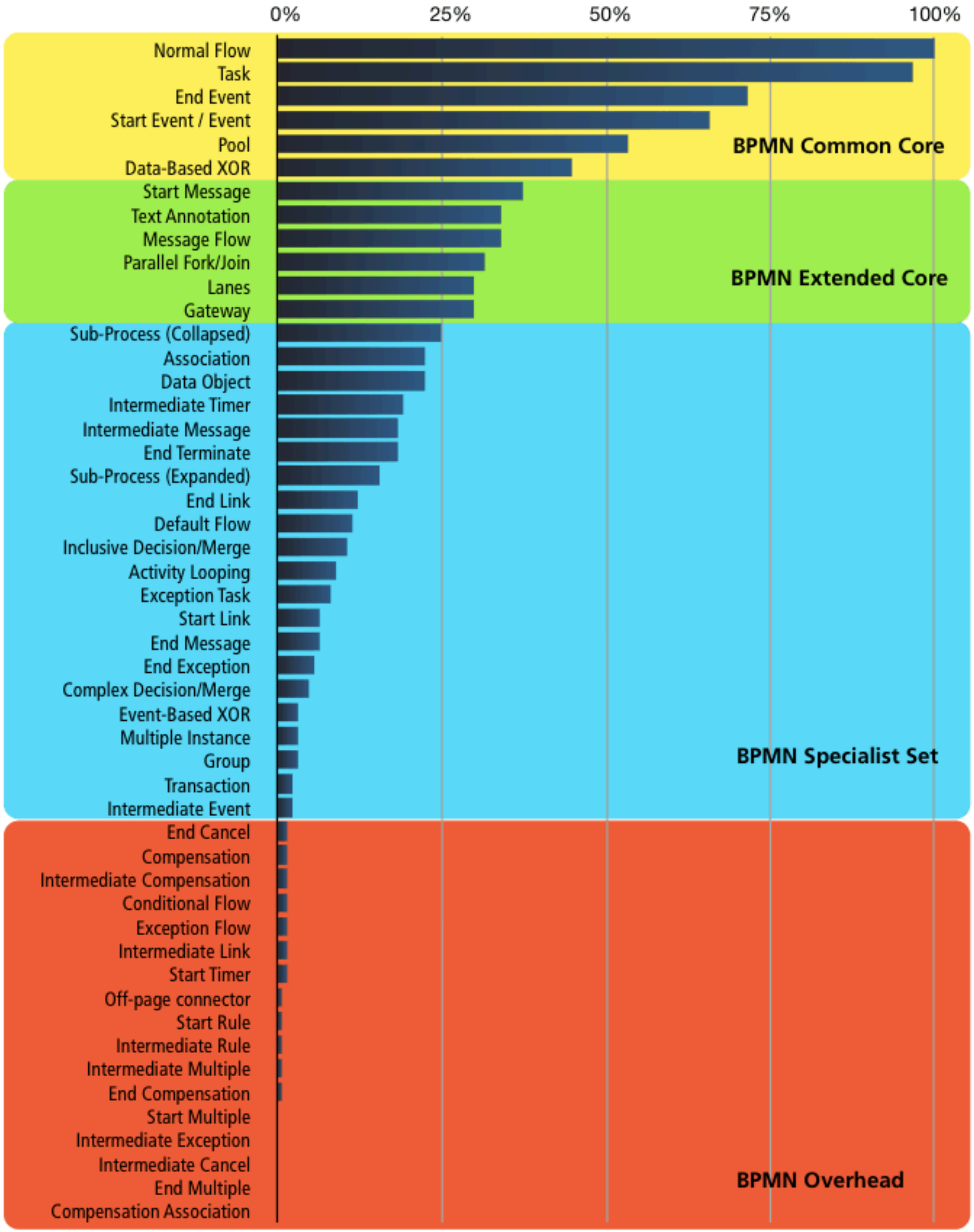
Use of BPMN elements into practice



The BPMN 2.0 [15] defines 50 constructs and their attributes. However, less than 20% of its vocabulary is used regularly in designing BP models [14].

zur Muehlen, Michael and Recker, Jan C. (2008) How Much Language is Enough? Theoretical and Practical Use of the Business Process Modeling Notation . In Proceedings 20th International Conference on Advanced Information Systems Engineering, Montpellier, France.

How much language do you need?

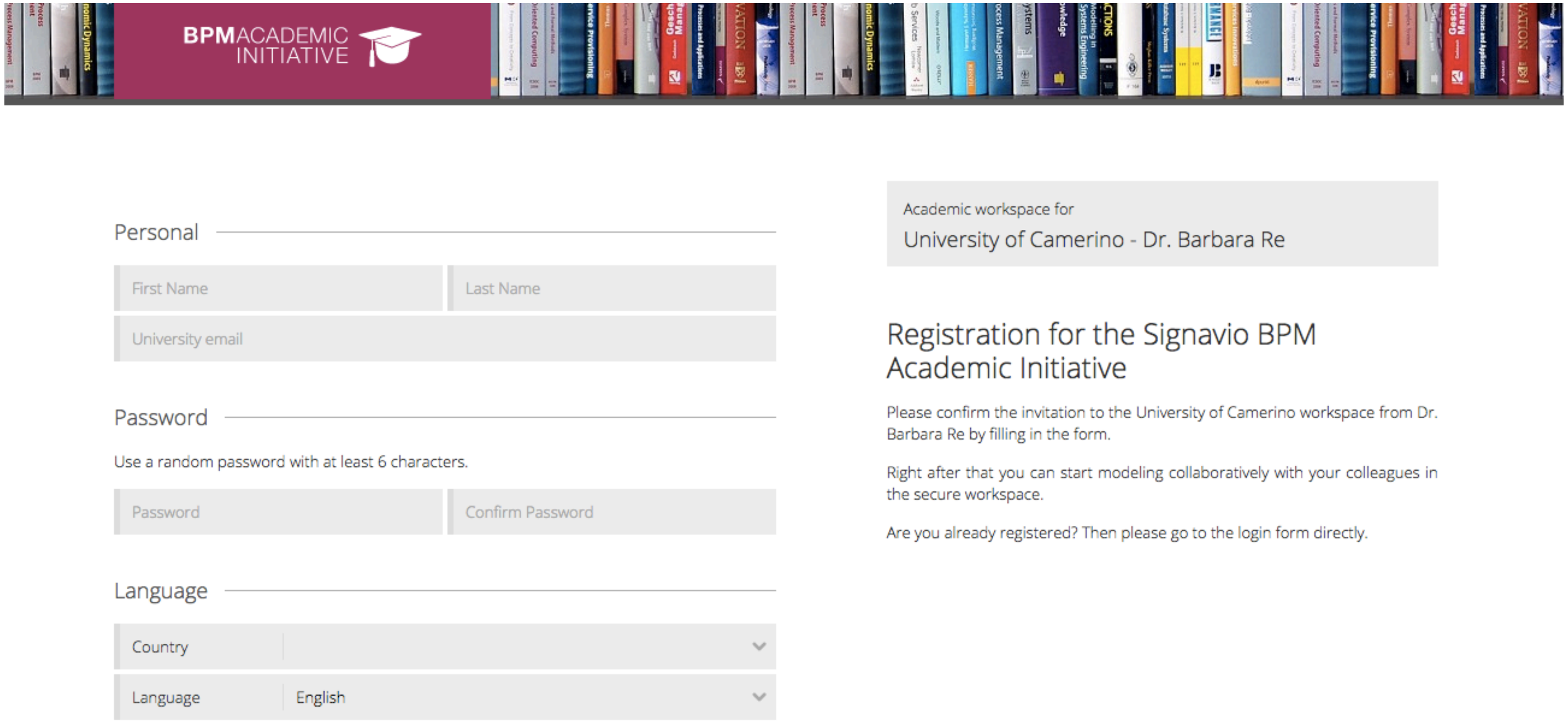


BPMN 2.0 free editors

- ADOxx - http://www.adoxx.org/live/home
- Oryx online editor: http://bpt.hpi.uni-potsdam.de/Oryx
- BizAgi editor (Windows): http://www.bizagi.com/
- Sketchpad Java editor (still in development): http://sourceforge.net/projects/sketchpadbpmn/develop
- Intalio|BPM community edition: http://www.intalio.com/products/bpm/community-edition/
- Eclipse BPMN modeler: http://www.eclipse.org/bpmn/
- Visio stencils: http://bpt.hpi.uni-potsdam.de/Public/BPMNCorner#Tooling
- Other tools & info: http://bpmn.org/BPMN_Supporters.htm

Modeling Environment

- http://academic.signavio.com/p/register?link=49b6ca940da04308ac5f133740119d41



Personal

First Name | Last Name

University email

Password

Use a random password with at least 6 characters.

Password | Confirm Password

Language

Country

Language: English

Academic workspace for
University of Camerino - Dr. Barbara Re

Registration for the Signavio BPM Academic Initiative

Please confirm the invitation to the University of Camerino workspace from Dr. Barbara Re by filling in the form.

Right after that you can start modeling collaboratively with your colleagues in the secure workspace.

Are you already registered? Then please go to the login form directly.

Business Process (Modelling Activities)

Execises

Using a BPMN 2.0 notation model the following process

- Cooking's Apple Pie
- Enrollment at university
- Buy a fly ticket
- Buy a book on-line
- Moving from one city to another

Let's Modelling: Insurance Claim

1. recording the receipt of the claim
2. establishing the type of the claim
3. checking covering of client's policy
4. checking the premium (payments up to date?)
5. rejection, if 3 or 4 has negative result
6. producing a rejection letter
7. roughly estimate the amount to be paid, if 3 & 4 have positive results 8. appointment of an assessor, if needed
9. revision of the amount offered to the client
10. recording client's reaction
11. assessment of objection: decision to revise 9 or take legal action 12 12. legal proceedings
13. payment of claim
14. filing and closure of claim

Modelling Travel agency

Travel agency: define a series of task for booking a flight, a hotel and optionally a car, with the possibility to change dates, to cancel the booking, to confirm the booking. Then, draw a process diagram relating the tasks.

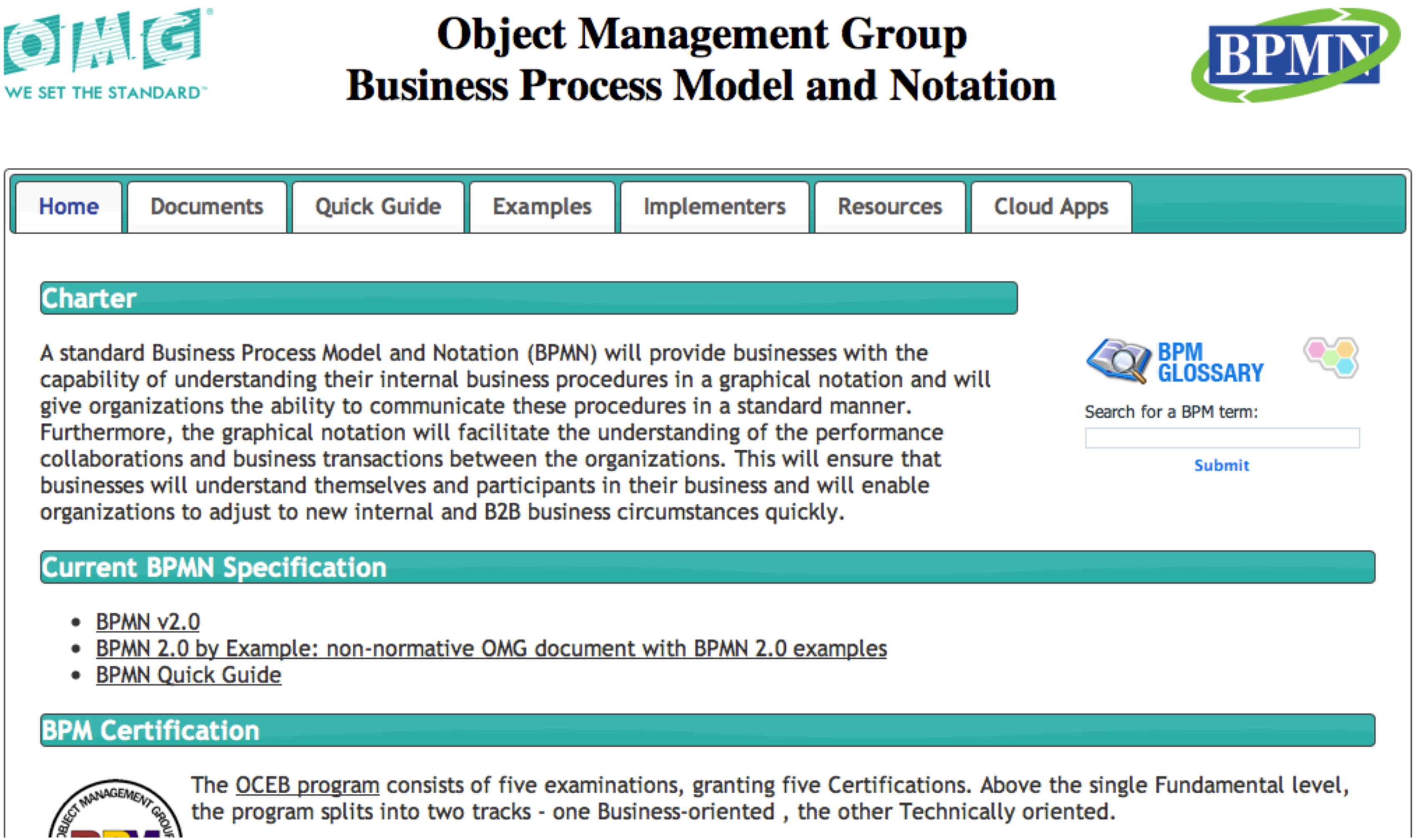
Modelling Coffee break

Coffee break: draw the process diagram for a vending machine that accepts a coin, then gives the possibility (1) to get a coffee or (2) to insert another coin and get either a cappuccino or a tea. Draw the process diagrams for a compatible and a "problematic" butler robot.

Modelling Bike Sharing and Bike Travel

Bike Sharing and Bike Travel: define a series of task for bike traveling, with the possibility to register, to track, and to maintain the bike. Then, draw a process diagram relating the tasks.

BPMN by example



Object Management Group
Business Process Model and Notation

Charter

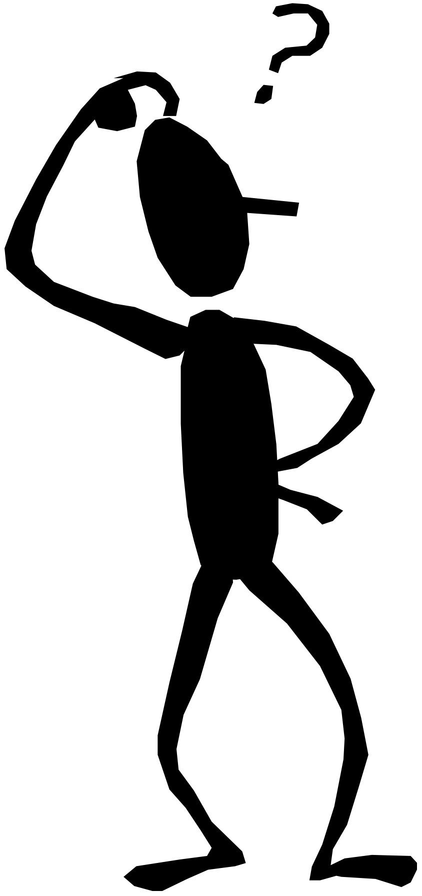
A standard Business Process Model and Notation (BPMN) will provide businesses with the capability of understanding their internal business procedures in a graphical notation and will give organizations the ability to communicate these procedures in a standard manner. Furthermore, the graphical notation will facilitate the understanding of the performance collaborations and business transactions between the organizations. This will ensure that businesses will understand themselves and participants in their business and will enable organizations to adjust to new internal and B2B business circumstances quickly.

Current BPMN Specification

- BPMN v2.0
- BPMN 2.0 by Example: non-normative OMG document with BPMN 2.0 examples
- BPMN Quick Guide

BPM Certification

The OCEB program consists of five examinations, granting five Certifications. Above the single Fundamental level, the program splits into two tracks - one Business-oriented , the other Technically oriented.



Questions?