

Business Process Modeling

Process modeling, BPMN

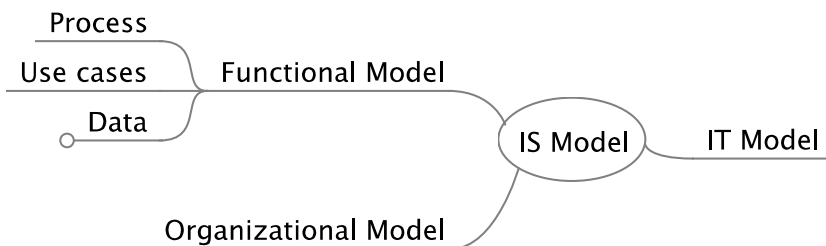


SoftEng
<http://softeng.polito.it>

© Maurizio Morisio, 2015



Functional model

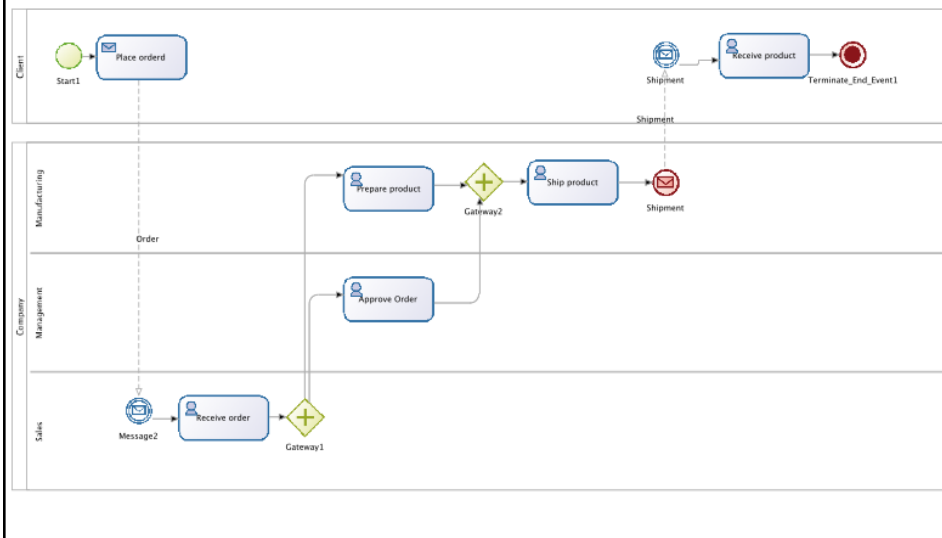


Functional model – submodels

- Process flow
 - ♦ Process modeling
 - UML Activity Diagrams
 - BPMN
- Information
 - ♦ Conceptual modeling
 - UML Class diagrams
 - (Entity–Relationships)
- Interaction
 - ♦ Interaction modeling
 - Use cases

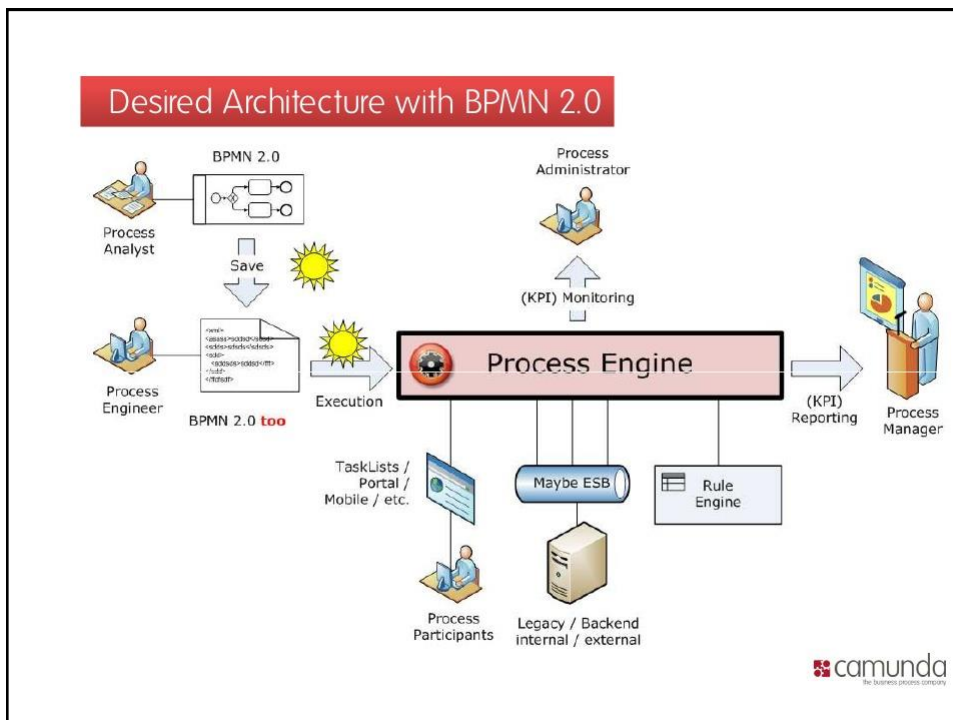
3

Example

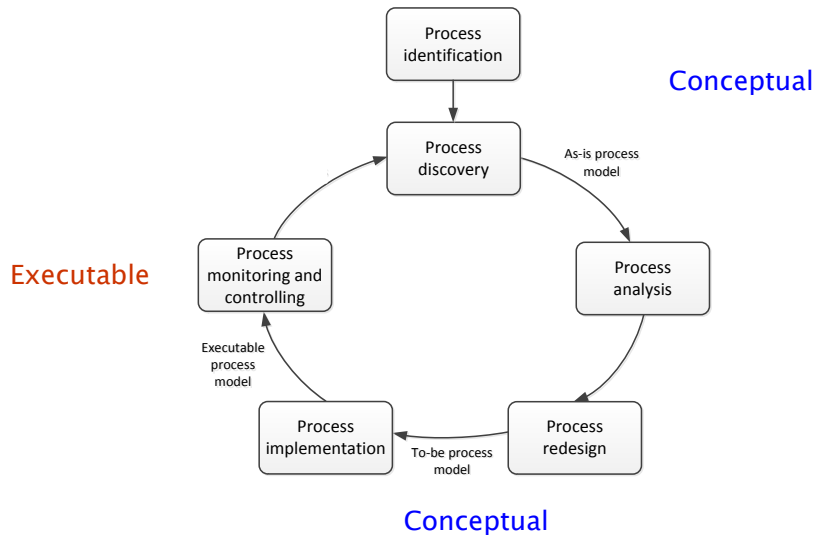


BPMN, usage

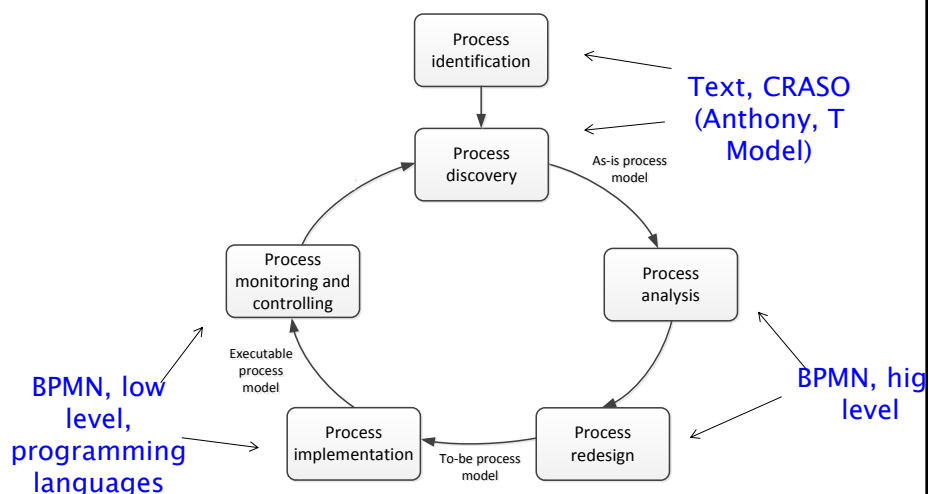
- Business Process Modelling Notation
- OMG – Object Management Group
- Model of existing processes
- Design of new processes
- Execution on process engine



Business process lifecycle



Levels of process description



Processes, conceptual level

- Free text
 - ♦ Receive order
 - ♦ Manage production of order
 - ♦ Manage warehouse
 - ♦ Record order payment and issue receipt
 - ♦ Hire new employee
 - ♦ ..

Processes, conceptual level

- T model
 - ♦ Support
 - Hire new employee (HR)
 - Record order payment and issue receipt (accounting)
 - ♦ Primary
 - Receive order
 - Manage production of order
 - Manage warehouse

Definition

- **BPMN Business Process Modeling Notation:**
A graphical representation for specifying business processes
- **Perspectives**
 - ♦ Functional (what is done and when)
 - ♦ Organizational (who does what)
 - ♦ Data

-
- **BPMN is meant to represent processes at**
 - ♦ Medium level of detail
 - process analysis phase
 - ♦ High level of detail
 - Process implementation phase

BPMN, Functional perspective

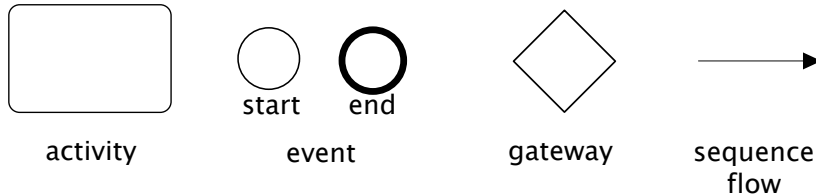
Business Process Model and Notation

- OMG standard (nowadays BPMN 2.0)
- Both for conceptual and executable models
- Supported by numerous tools:
bpmn.org lists over 70 tools, incl.
 - ♦ Signavio
 - ♦ Bizagi Process Modeler
 - ♦ Cameo Business Analyst
 - ♦ Camunda



BPMN from 10,000 miles...

A BPMN process model is a graph consisting of four types of **core elements**:



Let's start modeling

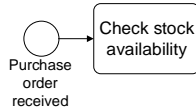
Order-to-cash

An order-to-cash process is triggered by the receipt of a purchase order from a customer. Upon receipt, the purchase order has to be checked against the stock to determine if the requested item(s) are available. Depending on stock availability the purchase order may be confirmed or rejected.

If the purchase order is confirmed, an invoice is emitted and the goods requested are shipped. The process completes by archiving the order.

BPMN Model

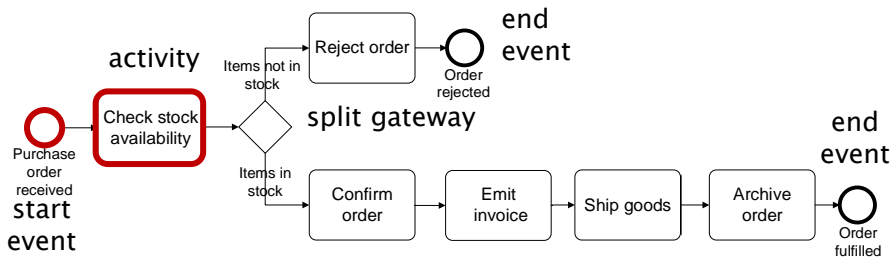
Order-to-cash



17

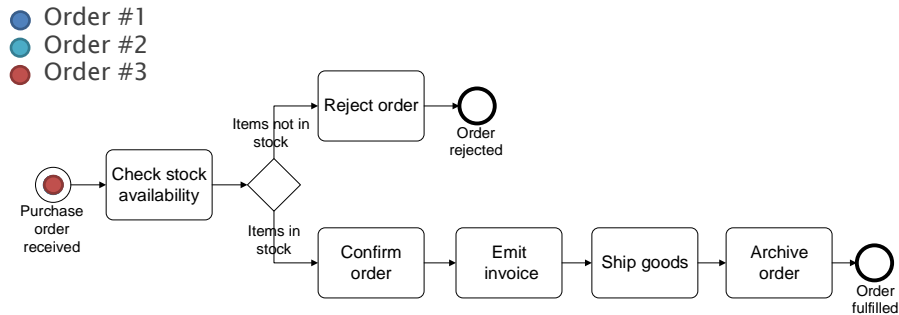
BPMN Model

Order-to-cash



18

Execution of a process model: tokens



19

Process model vs process instances

- Process model
 - ◆ Aka UML class diagram
 - ◆ Aka class X {} in Java
- Process instance, tokens
 - ◆ Aka UML instance diagram, instances
 - ◆ Aka new X; in Java

A little bit more on events...

A *start event* triggers a new process instance by generating a token that traverses the sequence flow (“tokens source”)

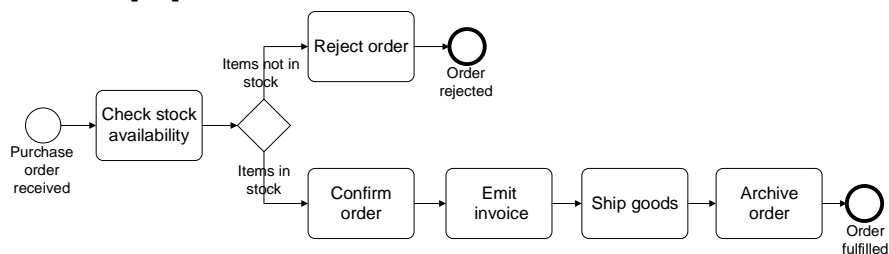


An *end event* signals that a process instance has completed with a given outcome by consuming a token (“tokens sink”)

21

Order-to-cash example revisited...

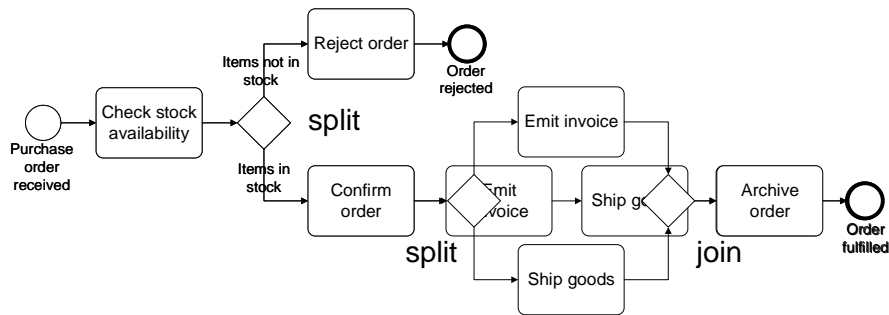
[...] If the purchase order is confirmed, an invoice is emitted and the goods requested are shipped (in any order). The process completes by archiving the order. [...]



22

First try

Order-to-cash

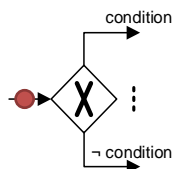


23

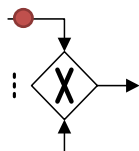
A little more on gateways: XOR Gateway



An *XOR Gateway* captures decision points (XOR-split) and points where alternative flows are merged (XOR-join)



XOR-split → takes **one** outgoing branch

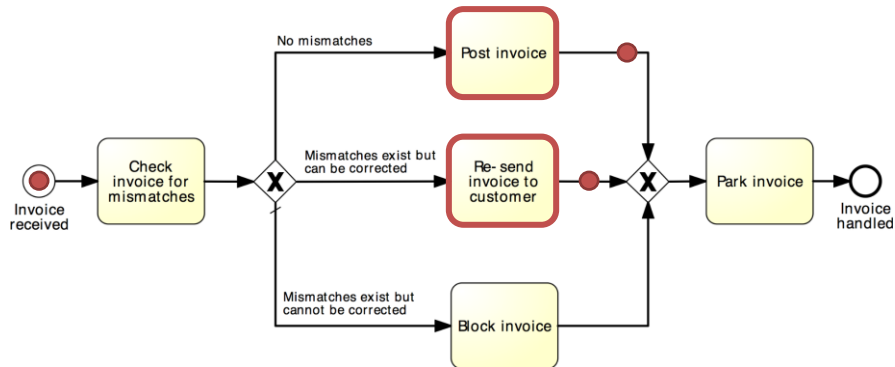


XOR-join → proceeds when **one** incoming branch has completed

24

Example: XOR Gateway

Invoice checking process

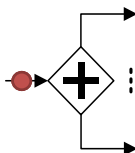


5

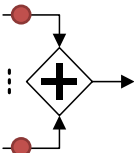
A little more on gateways: AND Gateway



An *AND Gateway* provides a mechanism to create and synchronize “parallel” flows.



AND-split → takes all outgoing branches

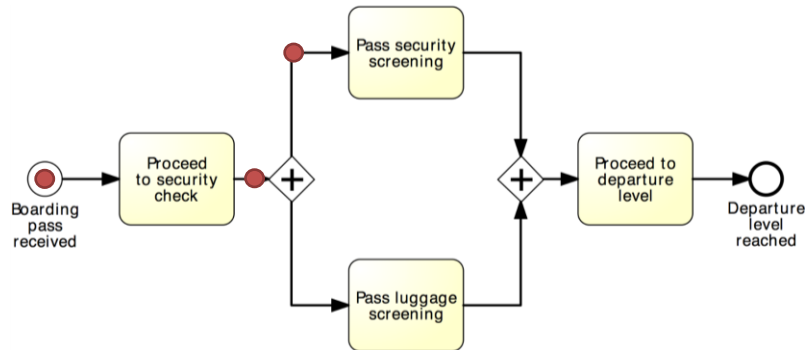


AND-join → proceeds when all incoming branches have completed

26

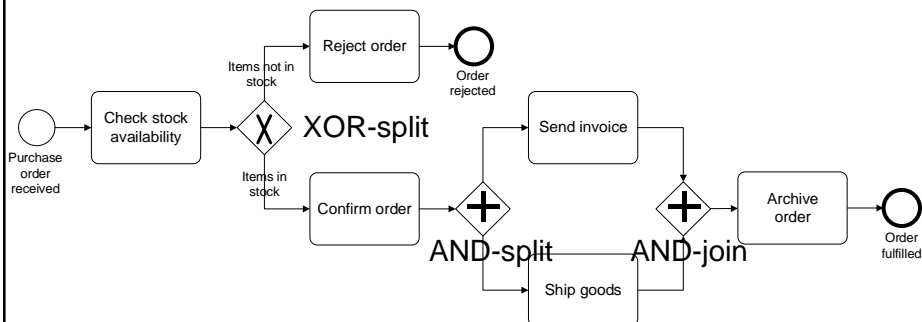
Example: AND Gateway

Airport security check



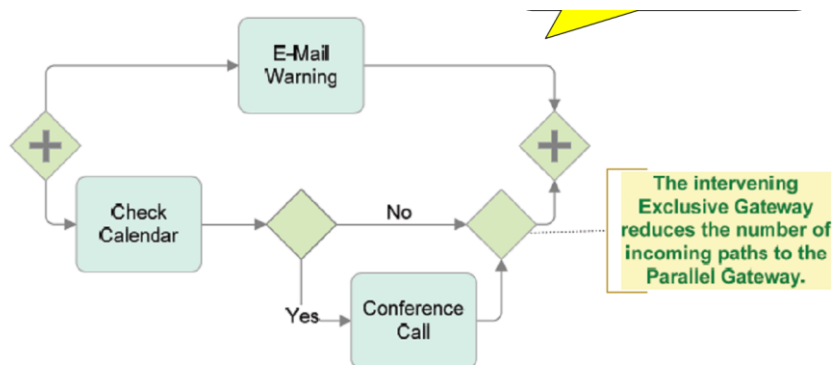
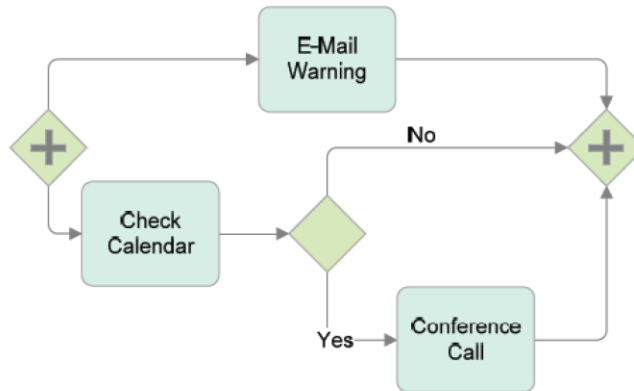
27

Revised order-to-cash process model



28

Result?



Between XOR and AND

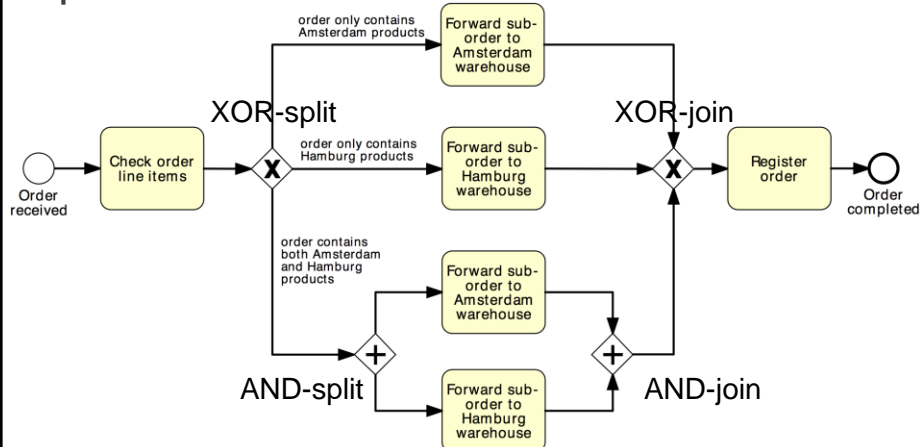
Order distribution process

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.

31

Solution 1

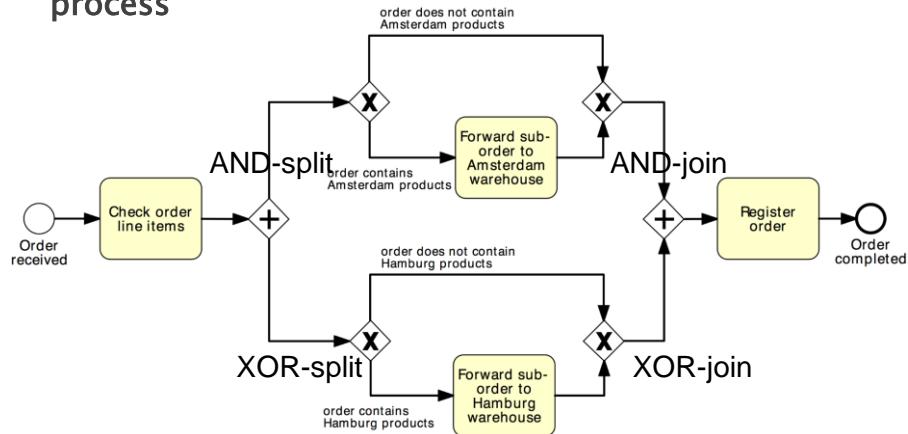
Order distribution process



32

Solution 2

Order distribution process

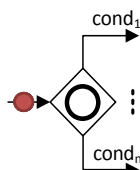


33

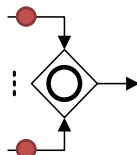
OR Gateway



An *OR Gateway* provides a mechanism to create and synchronize n out of m parallel flows.



OR-split → takes one or more branches depending on conditions

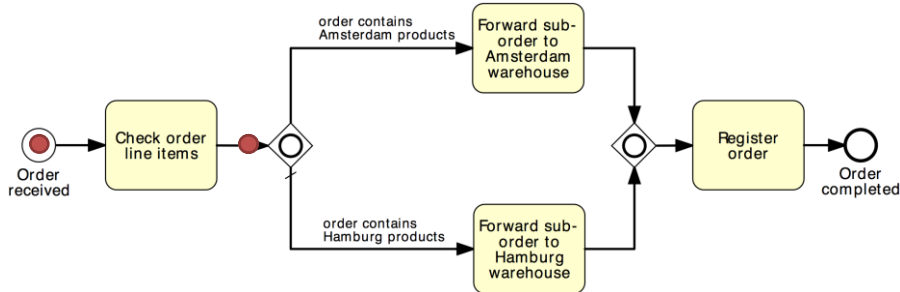


OR-join → proceeds when all active incoming branches have completed

34

Solution using OR Gateway

Order distribution process



35

BPMN Gateways: sum up

Exclusive (XOR)

- Exclusive decision
take one branch
- Exclusive merge
Proceed when one branch has completed

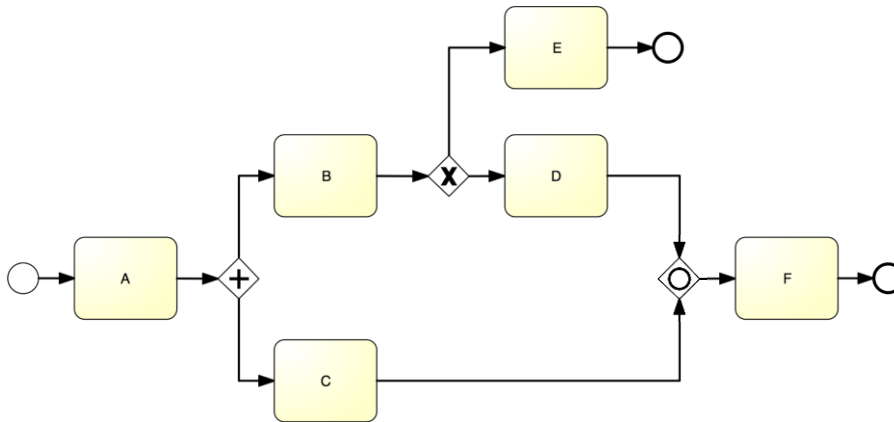
Parallel (AND)

- Parallel split
take all branches
- Parallel join
proceed when all incoming branches have completed

Inclusive (OR)

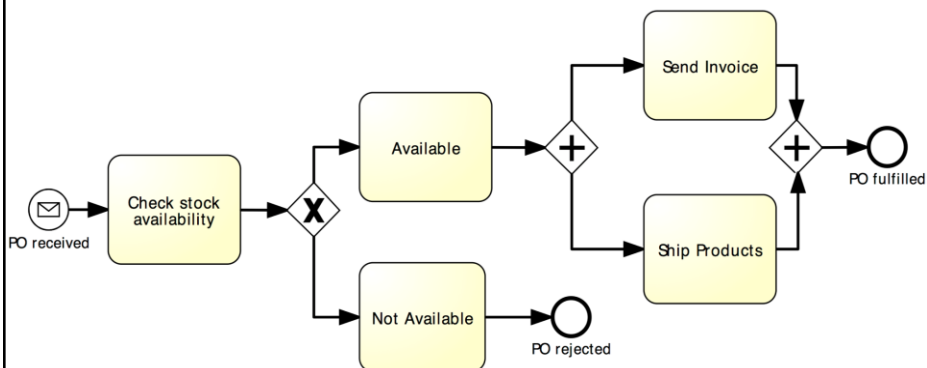
- Inclusive decision
take one or several branches depending on conditions
- Inclusive merge
proceed when all active incoming branches have completed

What join type do we need here?



37

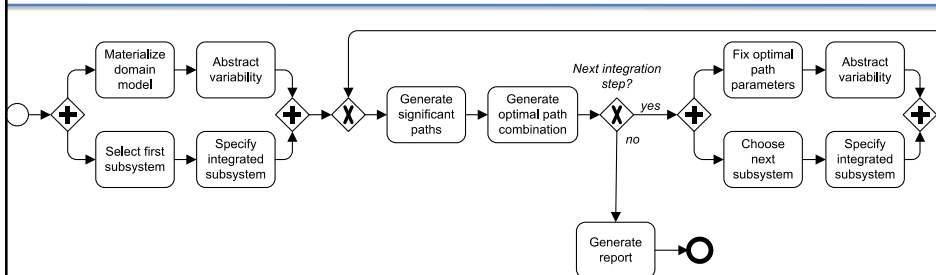
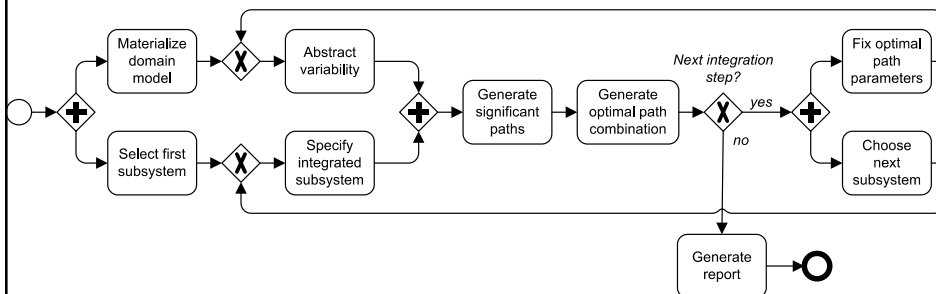
Beware: Beginner's Mistake ...



Guidelines: Naming Conventions

1. Give a name to every event and task
2. For tasks: verb followed by business object name and possibly complement
 - Issue Driver Licence, Renew Licence via Agency
3. For message events: object + past participle
 - Invoice received, Claim settled
4. Avoid generic verbs such as Handle, Record...
5. Label each XOR-split with a condition
 - Policy is invalid, Claim is inadmissible

Poll: Which model do you prefer?



One more guideline...

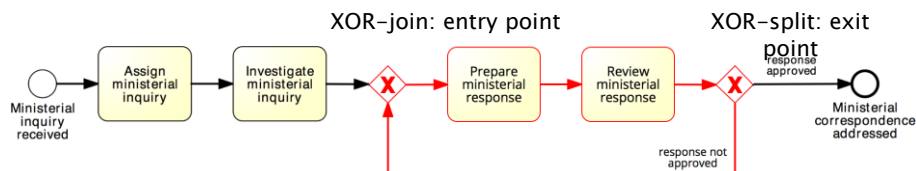
- Model in blocks
 - ♦ Pair up each AND-split with an AND-join and each XOR-split with a XOR-join, whenever possible
 - ♦ Exception: sometimes a XOR-split leads to two end events – different outcomes (cf. order management example)

Rework and repetition

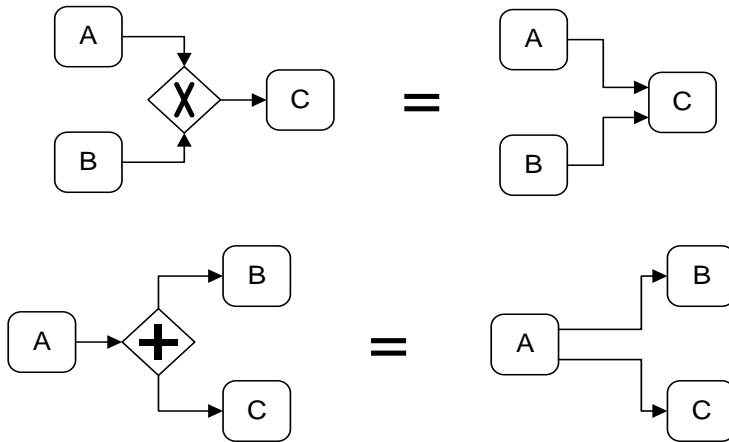
Address ministerial correspondence

In the minister's office, when a ministerial inquiry has been received, it is registered into the system. Then the inquiry is investigated so that a ministerial response can be prepared.

The finalization of a response includes the preparation of the response itself by the cabinet officer and the review of the response by the principal registrar. If the registrar does not approve the response, the latter needs to be prepared again by the cabinet officer for review. The process finishes only once the response has been approved.

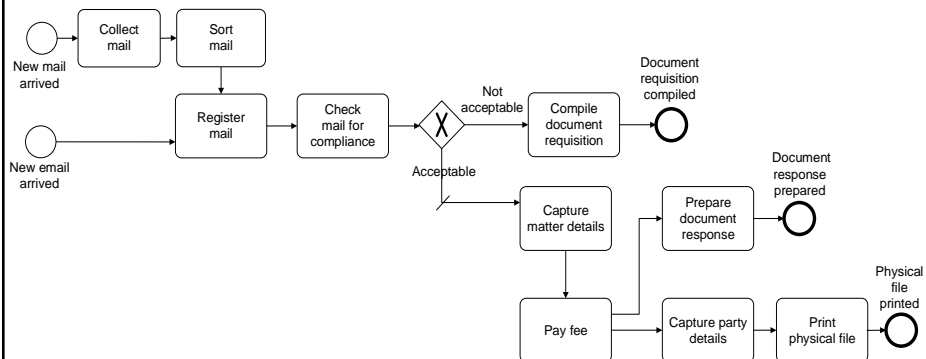


Quick Note: Implicit vs. explicit gateways



43

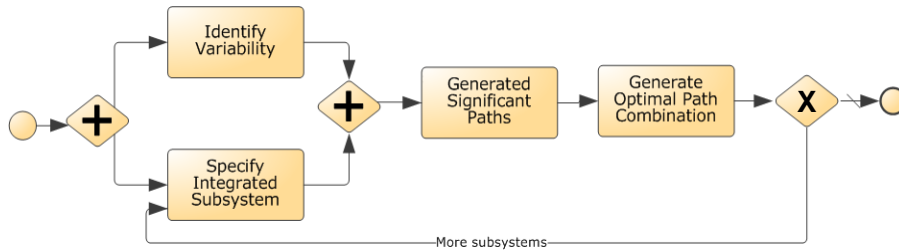
How this process starts? ~~How it ends?~~



44

What's wrong with this model?

How to fix it?

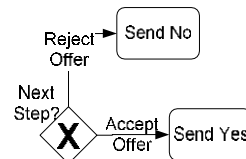
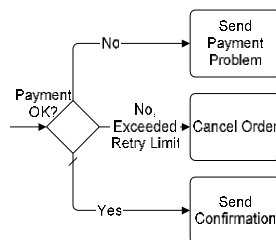


More on Exclusive Gateway

- The flow, after split, takes only one flow (aka XOR)
 - ♦ Based on data
 - ♦ Based on event

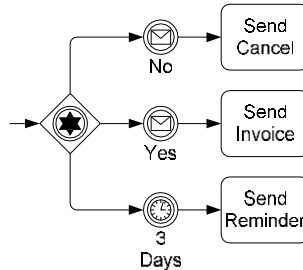
Exclusive, based on data

- ♦ Dash indicates default flow
- ♦ Diamond can also be drawn without X



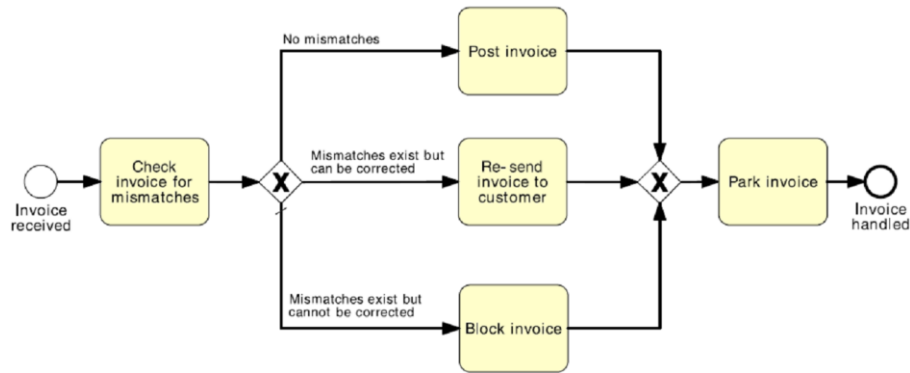
Exclusive, based on event

- First event that happens decides the next flow



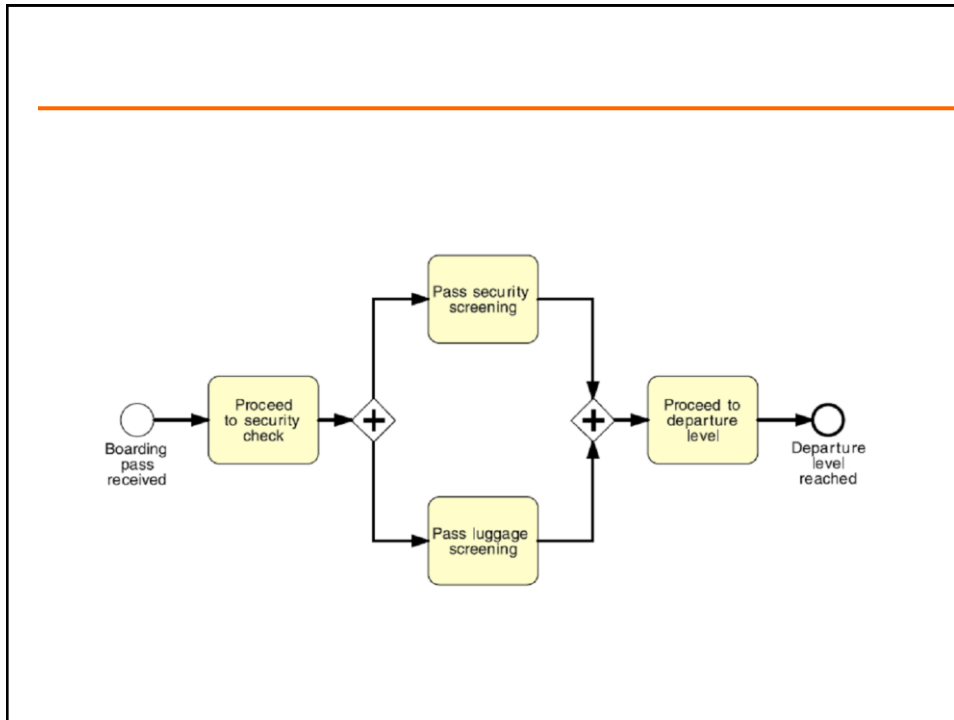
Exercise

- 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:
 - ♦ there are no mismatches, in which case the invoice is posted;
 - ♦ there are mismatches but these can be corrected, in which case the invoice is re-sent to the customer;
 - ♦ 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.



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.



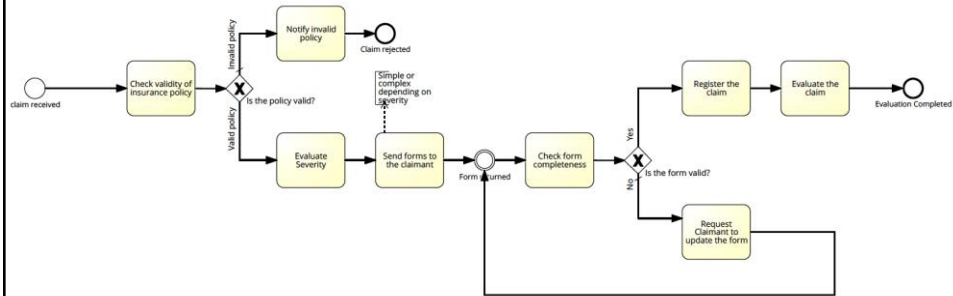
Exercise

When a claim is received, it is first checked whether the claimant has a valid insurance policy. If not, the claimant is informed that the claim is rejected due to an invalid policy.

Otherwise, the severity of the claim is evaluated. Based on the outcome (simple or complex claims), relevant forms are sent to the claimant. Once the forms are returned, they are checked for completeness.

If the forms are complete, the claim is registered in the Claims Management system and the evaluation of the claim may start. Otherwise, the claimant is asked to update the forms. Upon reception of the updated forms, they are checked again.

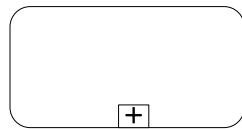
Solution



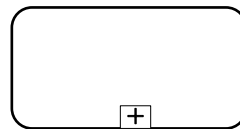
Sub processes

Sub process

- An activity can invoke a sub process.
Should be used to
 - ♦ Decompose large models
 - ♦ Share subprocesses

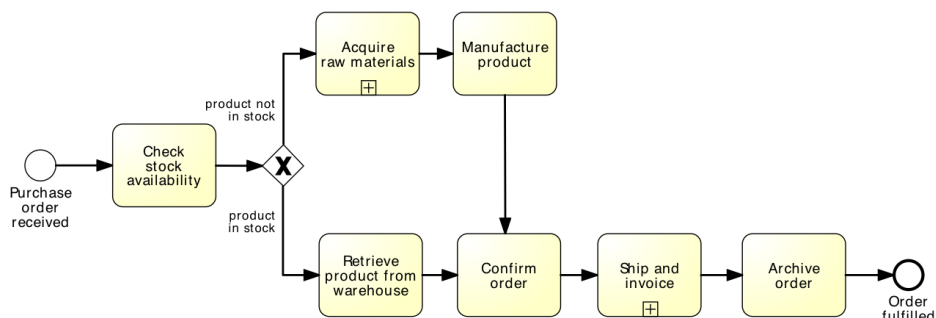


decomposition

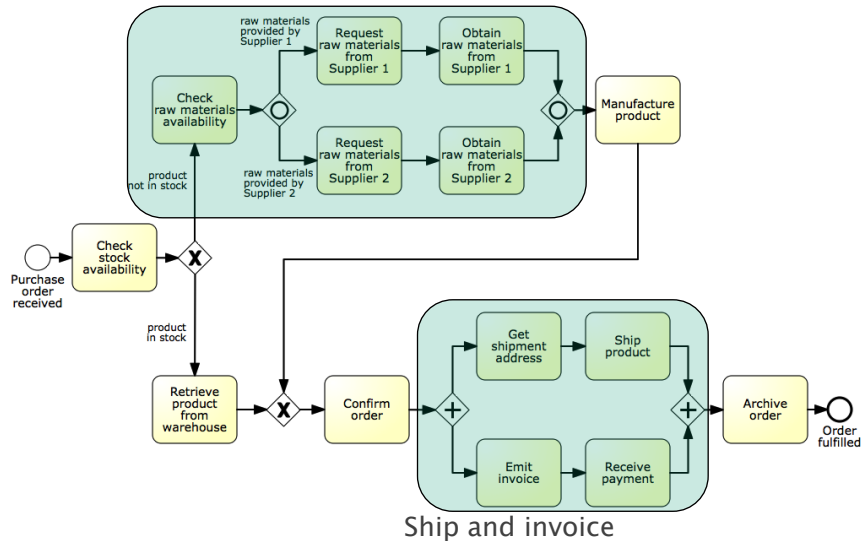


shared process (call activity)
(note bold border)

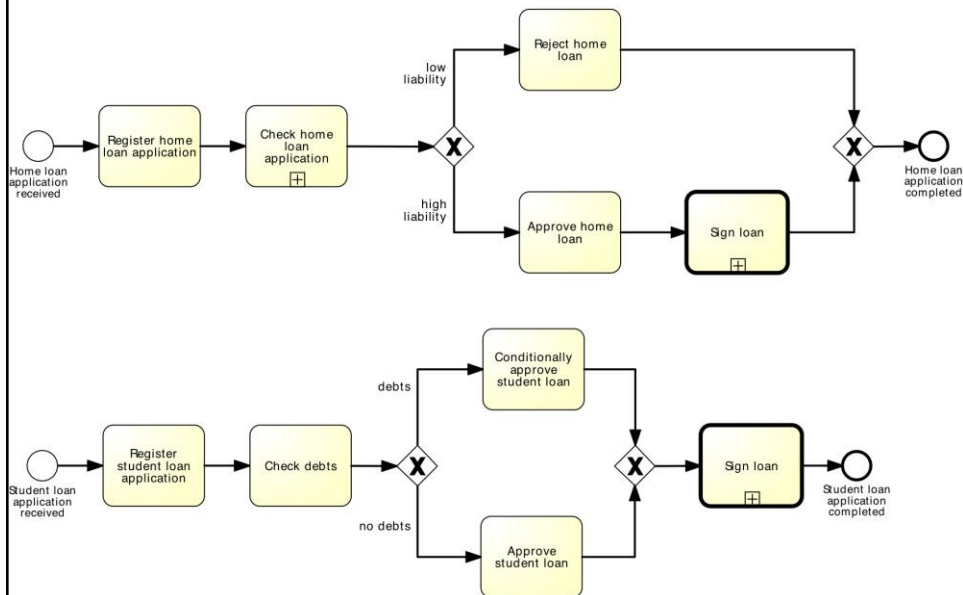
Ex: organize model



Acquire raw materials

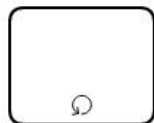


Ex: shared subprocesses

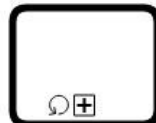


Repetitions

-
- Either a task, or a subprocess, can be repeated



repeated task



repeated sub process

-
- Further, it is possible to state if the repetition is made in sequence or in parallel

In parallel:

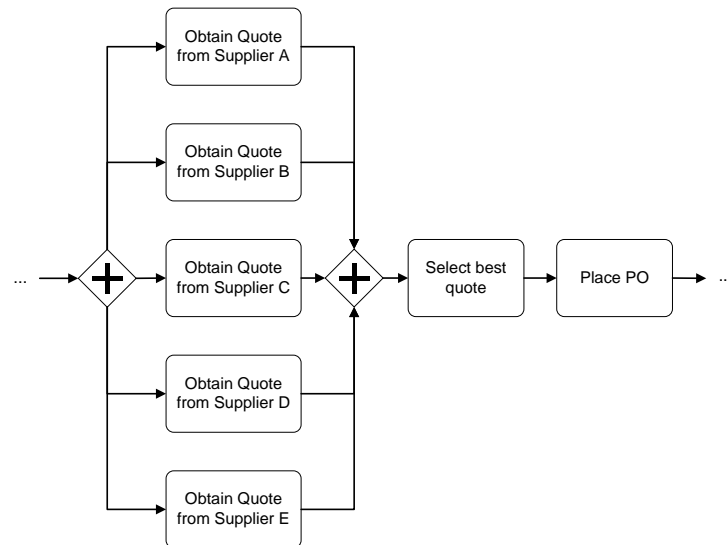


In sequence:

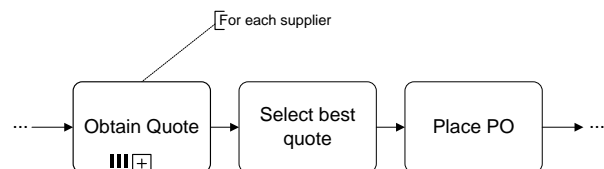


Ex.: 采购

In procurement, typically a quote is to be obtained from all preferred suppliers (assumption: five preferred suppliers exist). After all quotes are received, they are evaluated and the best quote is selected. A corresponding purchase order is then placed.








With parallel subtask



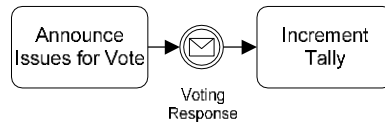
Events – advanced

Start events

- Have an associated trigger (condition that describes when event happens)
 - ♦ None has no trigger (for subprocesses)
 - ♦ Multiple has more than one

None	
Message	
Timer	
Rule	
Multiple	

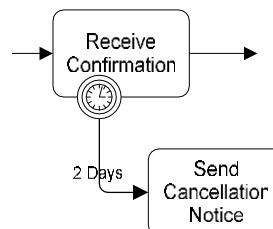
Intermediate events



None	
Message	
Timer	
Error	
Compensation	
Rule	
Multiple	




Intermediate events

- If attached to boundary of activity interrupt the activity



End events

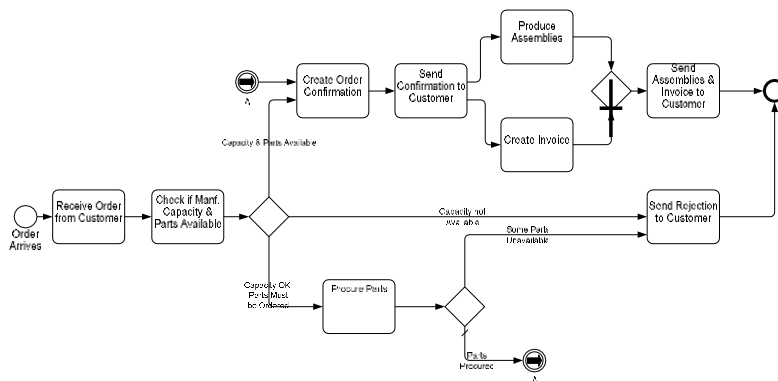
- Indicate end of process
- None for subprocesses

None	
Message	
Error	
Compensation	
Terminate	
Multiple	

Link events

Used to cut a process in parts

Deprecated



Tasks, advanced

User, manual, service



- Manual task: executed by person, with no software tool



- User task: executed by person, with software tool



- Service task: completely automated

Message task



Send Task



Receive Task

- Message send / receive activities that can be interrupted
- (if not interruptable use message events)

BPMN, Organizational
perspective

Elements

Resource

- ♦ Human actor or equipment (e.g. printer) that is required to perform an activity

Resource class: Set of resources with shared characteristics, e.g. Clerk, Manager, Insurance Officer

- ♦ **Role** (skill, competence, qualification)
This classification is based on what a resource can do or is expected to do.
- ♦ **Group** (department, team, office, organizational unit). This classification is based on the organization's structure.

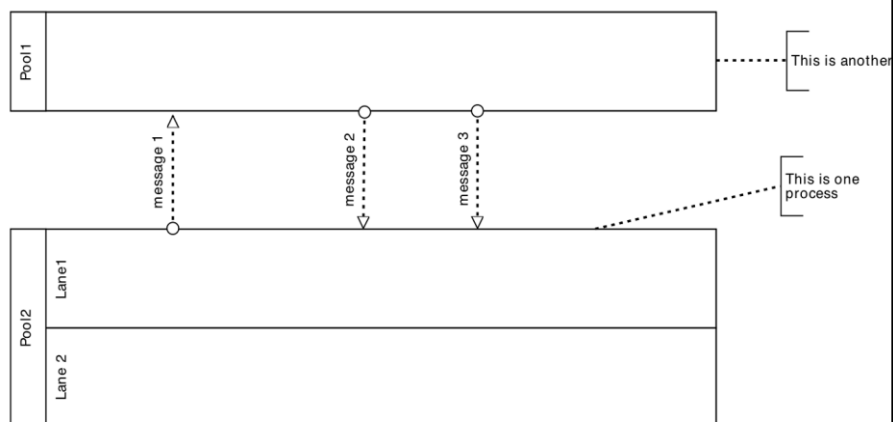
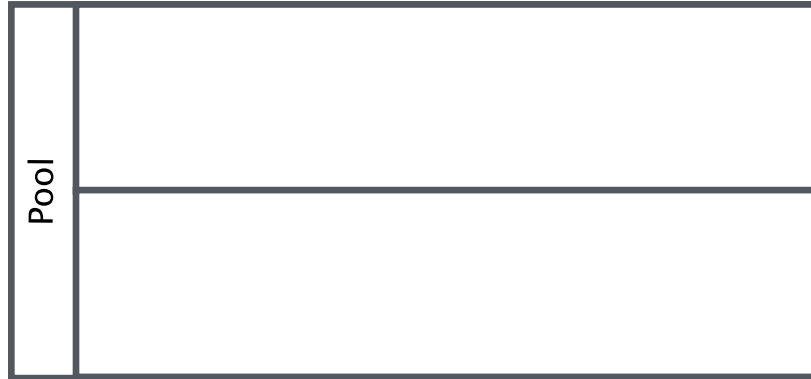
Organisational Elements

- Pool;
- Swimlane;
- Group.

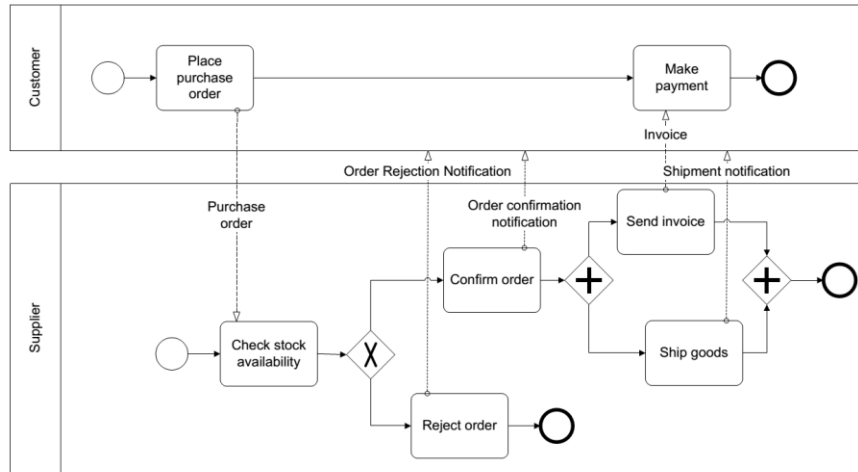
区隔线

Pool

- Contains a single, complete process.
- Flow cannot cross pool
- Processes in different pools communicate via messages



Order management w pools



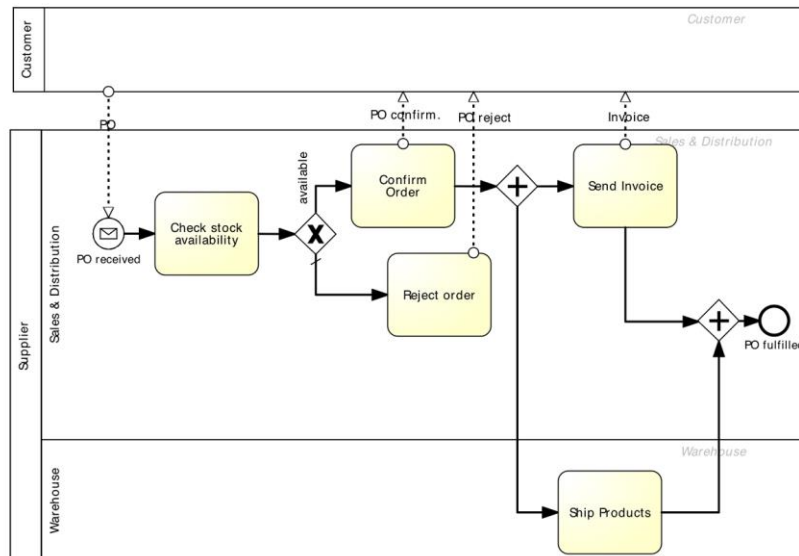
Lane

Inside a process (Pool) shows who does what.

Flow can cross lanes



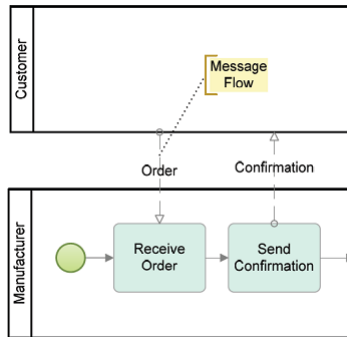
Order management w lanes



Pools and lanes

- Pool: independent organisational entity
 - ♦ Ex: customer == pool
supplier == pool
- Two pools do not share a common system to communicate
 - ♦ Must use messages /events to communicate
- Lane
 - ♦ Units of same organisational entity
 - ♦ Share a common system to communicate

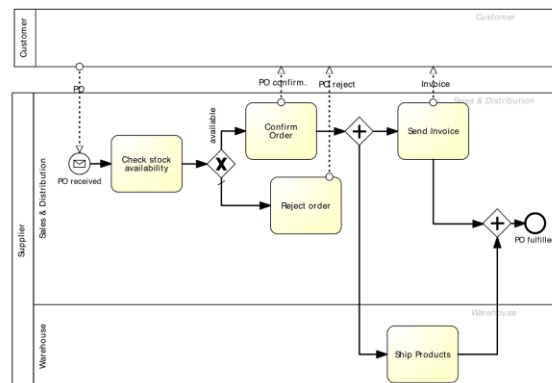
Message flow



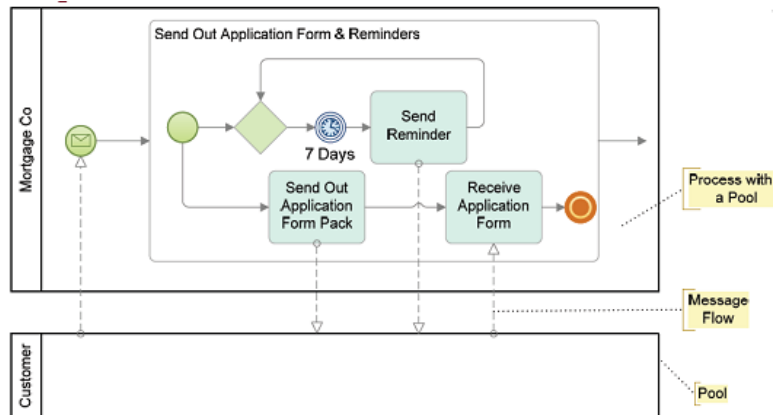
- Represents asynchronous communication between two pools (two processes)
 - ♦ Cannot happen within a pool / process

Black box pool

- Process in the pool is undefined
- Message flows connect to/from pool boundary

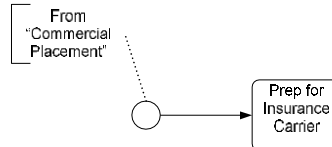


Black box pool



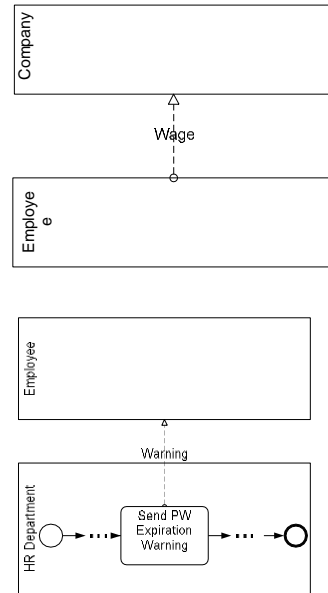
Text Annotation

To attach notes to a model entity with explanations for clarity.

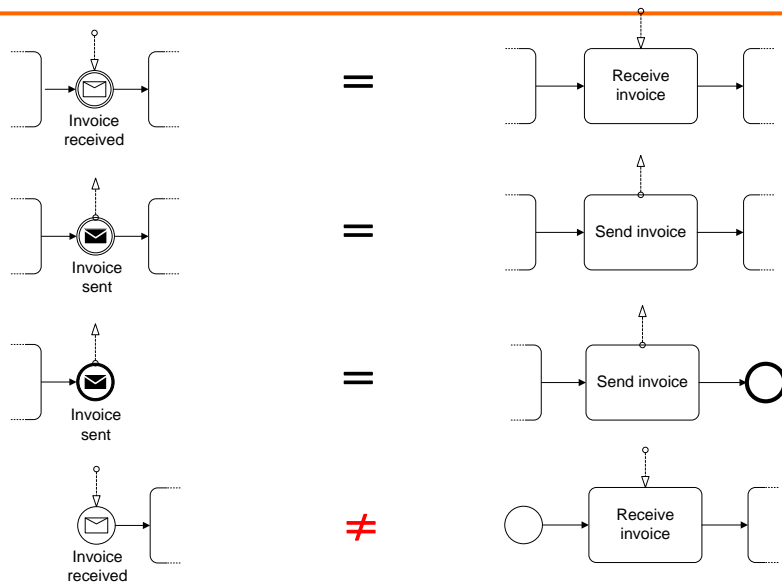


Message flow

- Shows flow of message
 - ♦ Between pools
 - ♦ Between activities in different pools



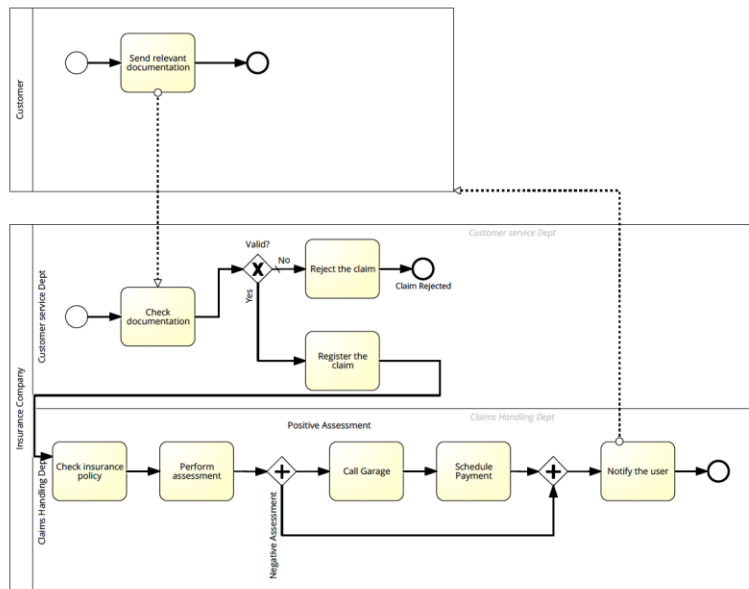
Equivalences



Exercise: Lanes, Pools

- Claims Handling process at a car insurer
- A customer submits a claim by sending in relevant documentation. The Customer Service department checks the documents for completeness and registers the claim. The Claims Handling department picks up the claim and first checks the insurance policy. Then, an assessment is performed. If the assessment is positive, a garage is phoned to authorise the repairs and the payment is scheduled (in this order). In any case (whether the outcome is positive or negative), an e-mail is sent to the customer to notify the outcome

Solution



Data perspective

Information Artifacts

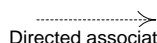


Data Objects show volatile data required or produced by activities.

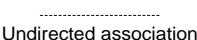
They represent input and output of a process activity.



Data stores are containers of data objects that need be persisted beyond the duration of a process instance



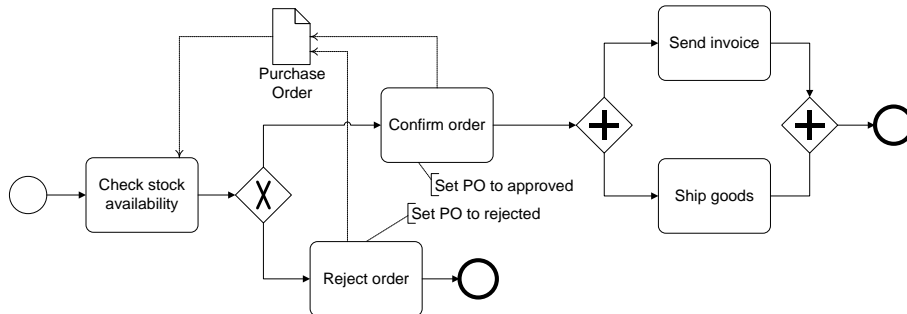
Directed association



Undirected association

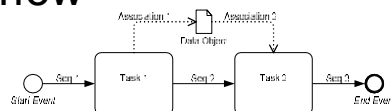
Associations are used to link artifacts such as data objects and data stores with flow objects (e.g. activities).

Order Processing Model with Artifacts

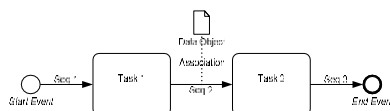


Artifacts – 2

- Data flow decoupled from sequence flow

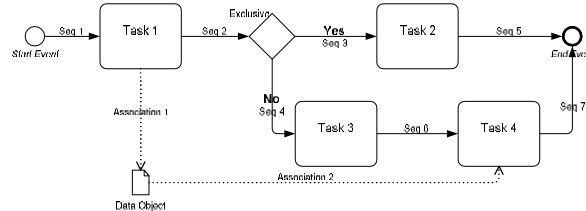


- coupled



Artifacts 3

- decoupled

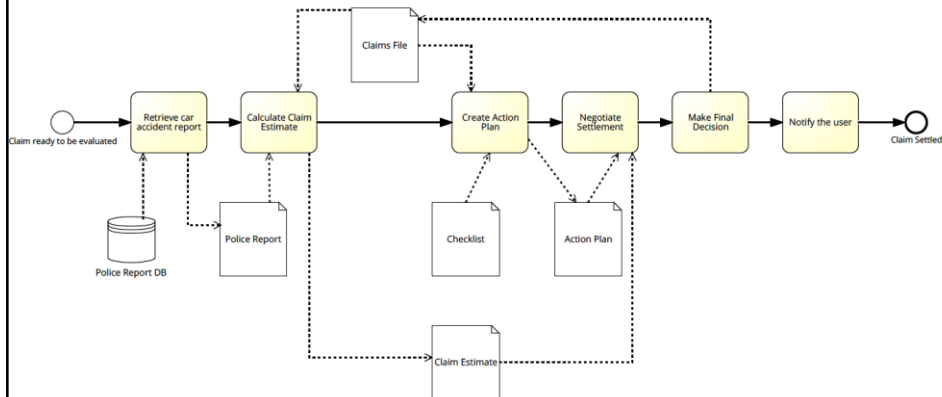


Exercise: Artifacts

职员

When a claim related to a major car accident is evaluated, a clerk first retrieves the corresponding car accident report in the Police Reports database. If the report is retrieved, it is attached to the claim file. The claim file and the police report serve as input to a claims handler who calculates an initial claim estimate. Then, an “action plan” is created based on a “checklist”. Based on the action plan and the initial claims estimate, a claims manager negotiates a settlement with the customer. After this negotiation, the claims manager makes a final decision, updates the claim file to record this decision, and sends a letter to the claimant to inform him/her of the decision.

Solution

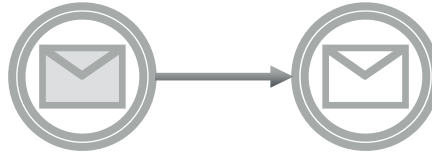


Special Behaviour Elements

- Message and Message Flows;
- Signals;
- Correlation;
- Timers;
- Errors;

Messages and Message Flow

Used to transfer actions or data from one pool or process to another and to correlate related processes.



Throw Message Catch Message

A message is a direct communication between two business participants. These participants must be in separate Pools (they cannot be sent from another Lane inside a single Pool)

Signals

Used to send data to multiple activities simultaneously.



Throw Signal Catch Signal

Signals are broadcast communications from a business participant or another Process. Signals have no specific target or recipient – i.e. all Processes and participants can see the signal and it is up to each of them to decide whether or not to react.

Timers

Used to launch periodic activities, or to ensure that an activity happens within a specified deadline



Timer

Error Events

- The Error Intermediate Event is used to handle the occurrence of an *error* that needs the interrupting of an Activity (to which it is attached).



Error end event

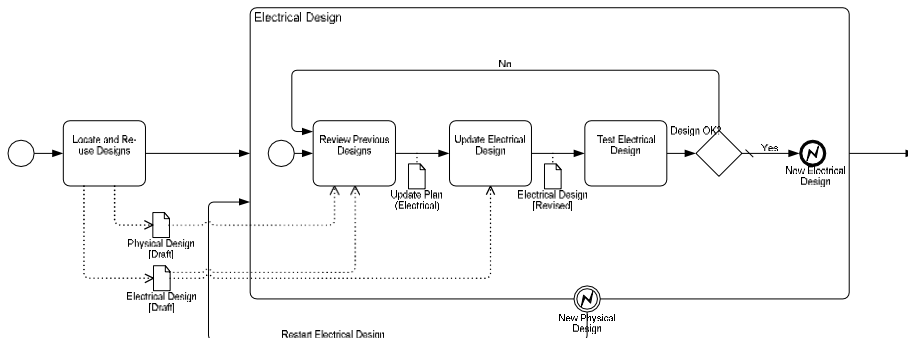
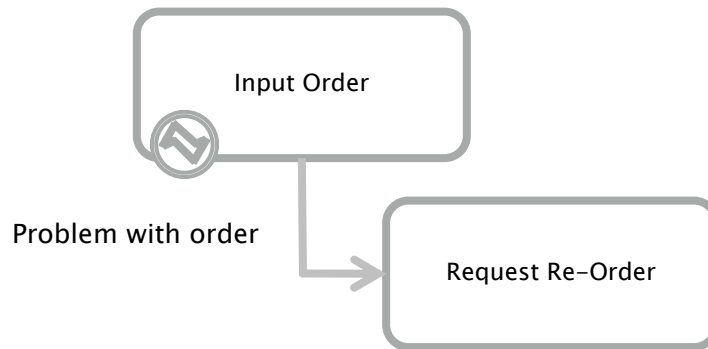


Error Intermediate Event

- The Error End Event is used to throw an error.
- The Error Intermediate Event can only be used when attached to the boundary of an Activity, thus it can only be used to catch an error.
- When an error occurs all work will stop for that Process.

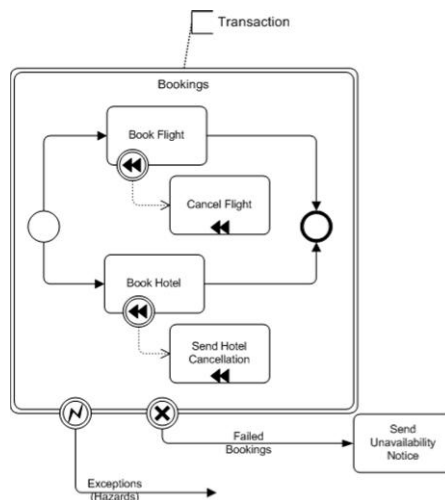
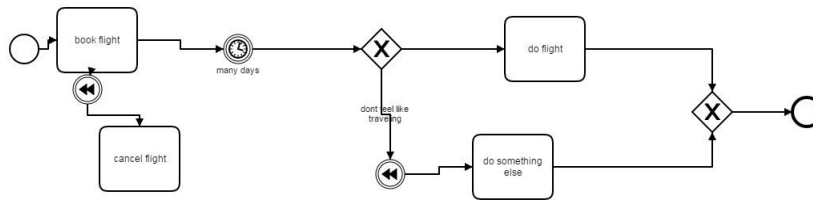
Exception

Used to define behaviour when the system encounters a technical error.



Compensation

- Event that cancels an activity already terminated in the past, triggering suitable compensation activities

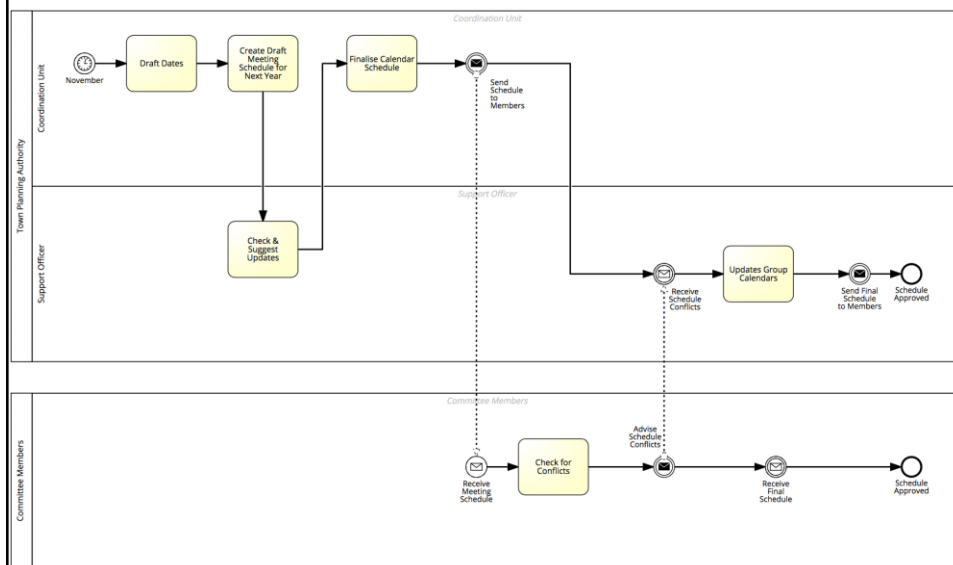


Exercise – Part 1

In November of each year, the Coordination Unit at the Town Planning Authority drafts a schedule of meetings for the next calendar year and adds draft dates to all calendars.

The Support Officer then checks the dates and suggests modifications. The Coordination Unit then rechecks all dates and looks for potential conflicts. The final schedule of meeting dates is sent to all the independent Committee Members by email, who then check their diaries and advise the Coordination Unit of any conflicts.

Solution – Part 1

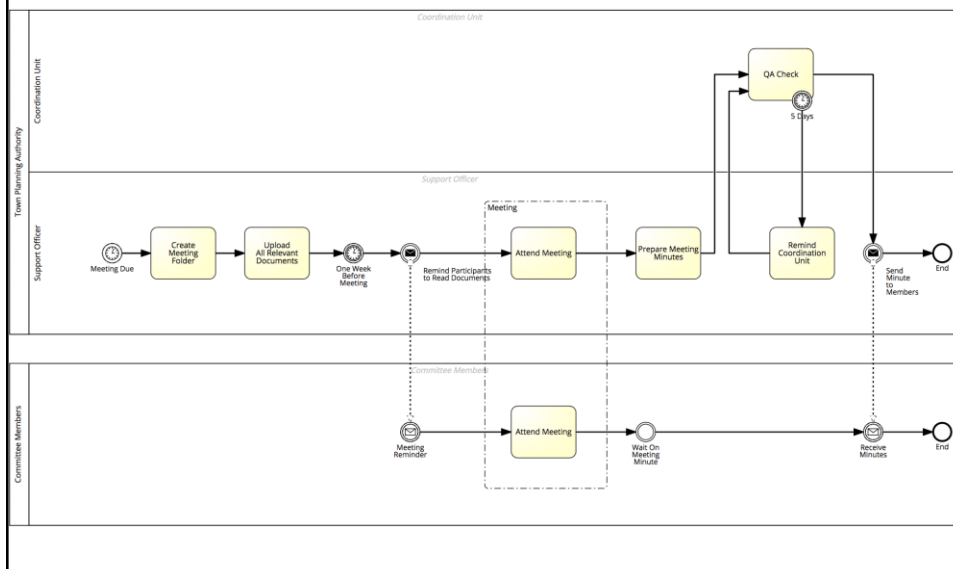


Exercise – Part 2

Once the dates are finalized (by the Coordination Unit), the Support Officer updates all group calendars and creates meeting folders for each meeting and ensures all appropriate documents are uploaded to system.

Committee Members are advised a week before each meeting to read all related documents. The Committee Members hold their meeting, and the Support Office then produces minutes including any Action Points for each Committee Member. Within 5 working days, the Coordination Unit must conduct a QA check on the minutes, which are then sent to all Committee Members. The Support Officer then updates all departmental records.

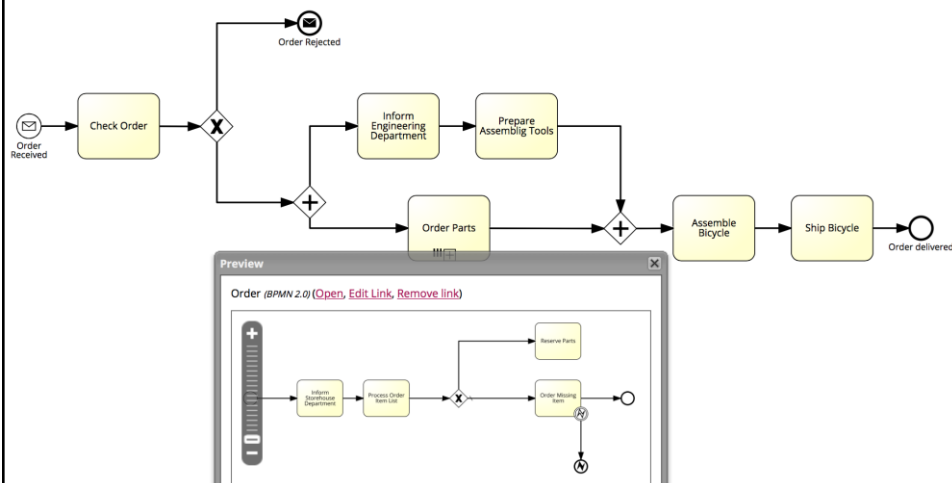
Solution – Part 2



Exercise

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

Solution



References

- Marlon Dumas – *Fundamentals of BPM*
- Andrea Marrella – *Modeling Business Processes with BPMN*
- OMG, *BPMN by example*

