

Process Modeling

MOMNA ZANEB





Process Modeling

- Rework & Repetition
- Information Artifacts
- Resources

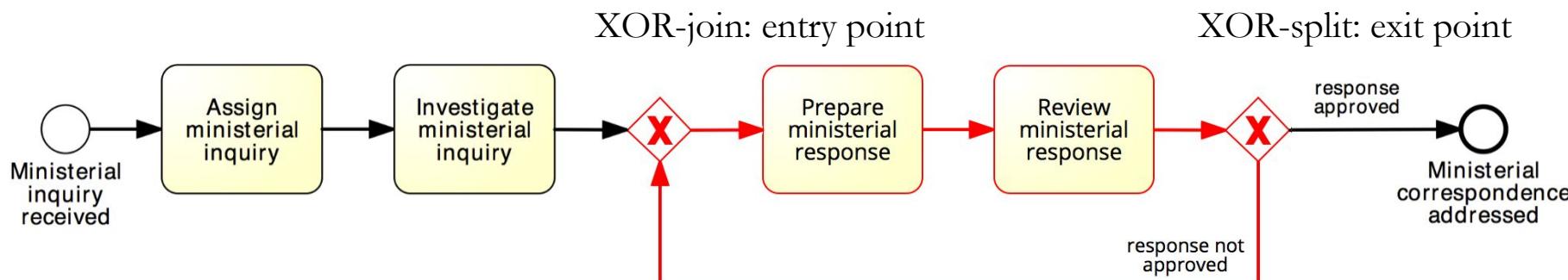
Rework and repetition

Address ministerial correspondence

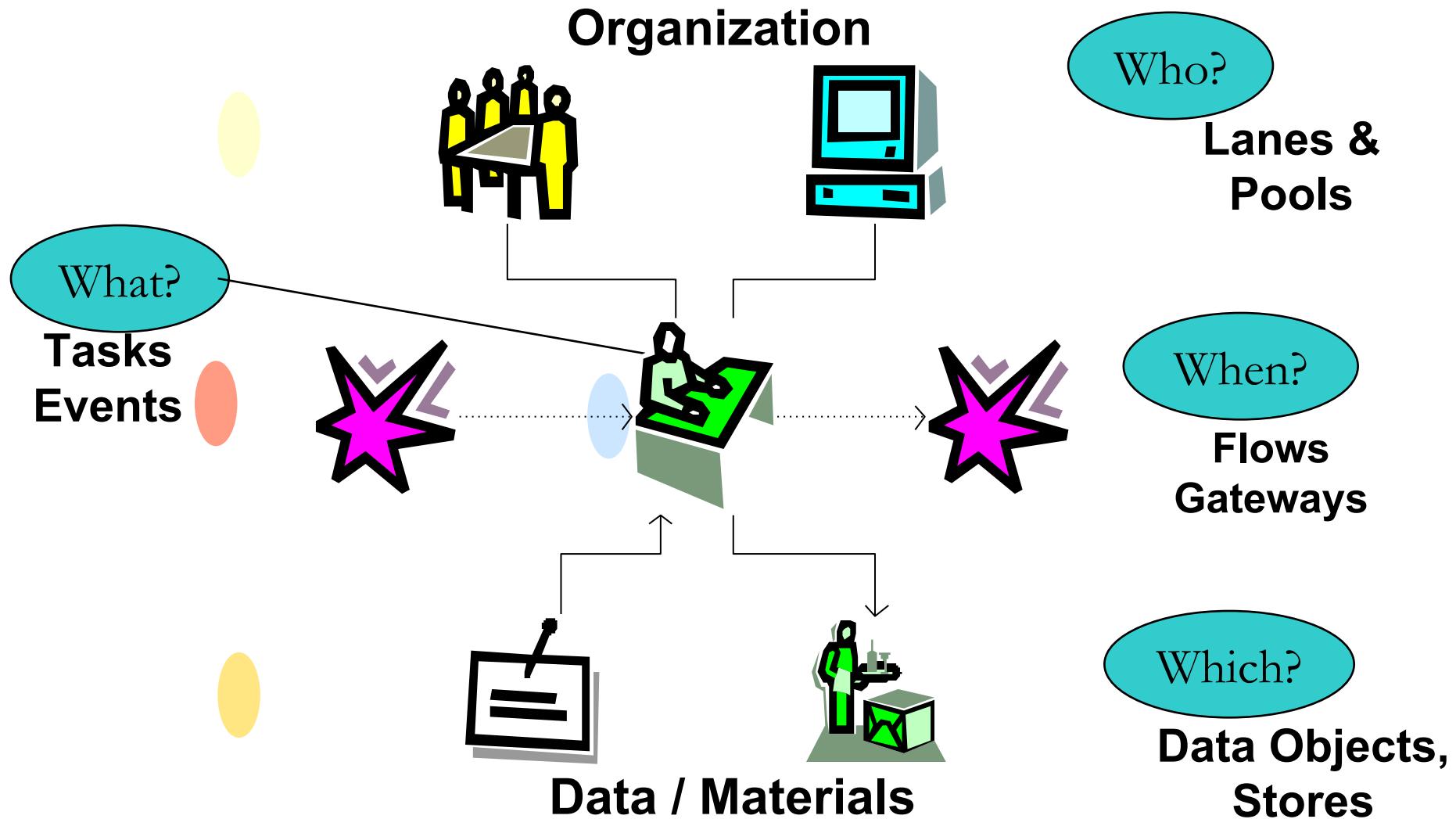
In the minister's office, when a ministerial inquiry has been received, it is registered into the system. Then the inquiry is investigated so that a ministerial response can be prepared.

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.

SESE = Single Entry Single Exit fragment, i.e. a fragment delimited by a single entry node and a single exit node (there are no other incoming arcs into the fragment or outgoing arcs from the fragment)



Process Modelling Viewpoints



Resources(Organizational Elements in BPMN – Pools & Lanes)

Resource or Organizational Perspective

Who or what performs which activity?

Resource is anything or anyone who is involved in the performance of the activity.

- A process participant (employee john smith)
- A software system (a server or website or phone application)
- An equipment , (a printer or manufacturing plant)

Active Resource (who can autonomously perform an activity , person using photocopying machine)

Passive Resource (resources merely involved in performance of activity , a photocopying machine)

We will focus on active resources.

Resources(Organizational Elements in BPMN – Pools & Lanes)

- Refer to a group of resources (Resource Class) , rather then referring to one resource at a time.
- In a process model never explicitly refer to one resource at a time (for example , An employee John Smith) , Refer to a group of these resources (for example if John Smith is a Clerk , refer to class Clerk)
- Resource group can refer to a whole organization , an organization unit , or a role

Resources(Organizational Elements in BPMN – Pools & Lanes)

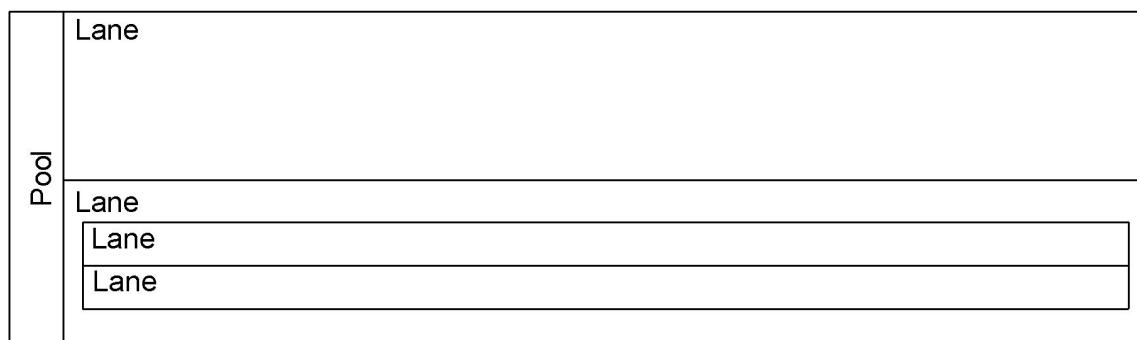
Pool – A very big box

Captures a resource class. Generally used to model a business party (e.g. a whole company)



Lane

A resource sub-class within a pool. Generally used to model departments (e.g. shipping, finance), internal roles (e.g. Manager, Associate), software systems (e.g. ERP, CRM)



Resources(Organizational Elements in BPMN – Pools & Lanes)

In practice, **each pool** represents a **distinct process** and **each set of resource classes** has its own pool.

Pools and Lanes \2



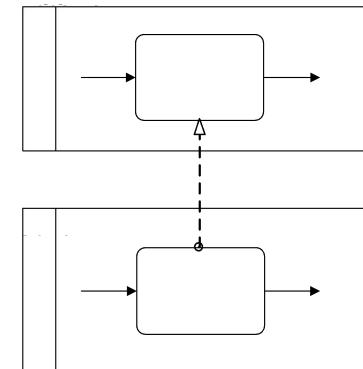
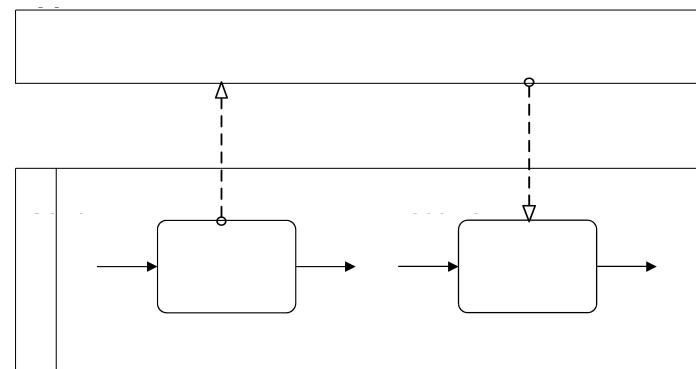
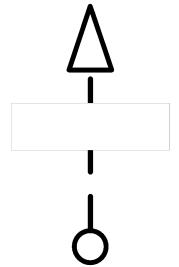
In each pool we may have multiple lanes, often assumed to represent **internal business roles** within a process. Lanes actually provide a generic mechanism for partitioning the objects within a pool.

Message Flow

A *Message Flow* represents a flow of information between two process parties (Pools)

A Message Flow can connect:

- directly to the boundary of a Pool captures an *informative* message to/from that party
- to a specific activity or event within that Pool captures a message that triggers a specific activity/event within that party

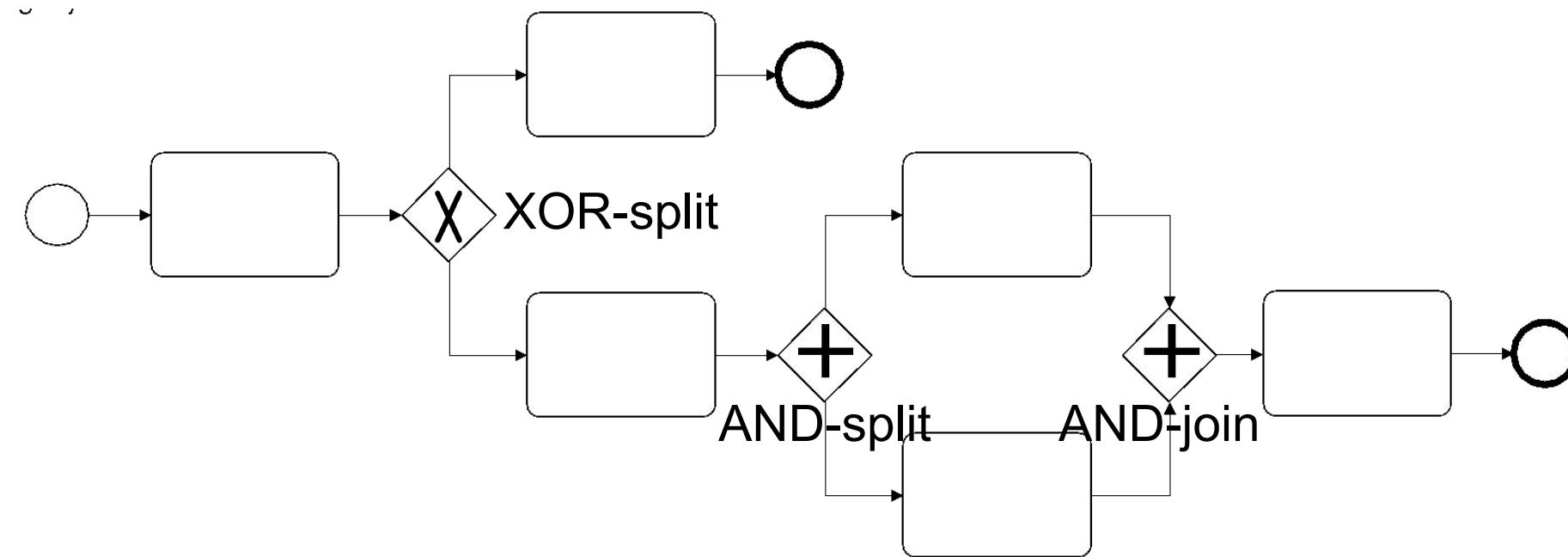


Order-to-Cash

- 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.

Revised solution

Order-to-cash



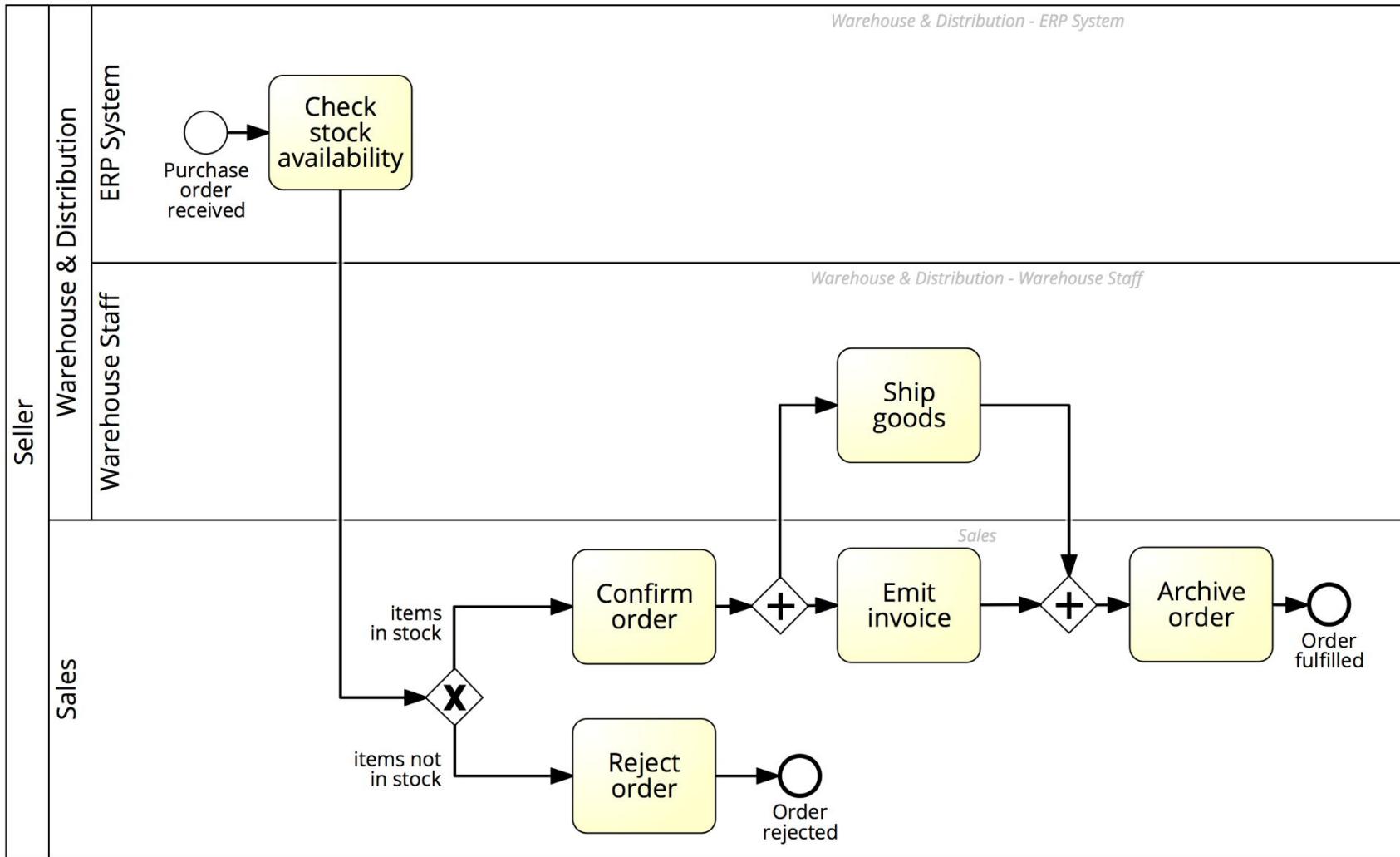
Resources in the order-to-cash example

The order-to-cash process is carried out by a seller's organization which includes two departments: the **Sales department** and the **Warehouse & Distribution department**.

The purchase order received by the Seller has to be checked against the stock. This is done via an **ERP** module within the **Warehouse & Distribution department**.

If the purchase order is confirmed, the **Warehouse & Distribution department** ships the goods. Meantime, the **Sales department** emits the invoice. The process concludes with the order being archived by the **Sales department**.

Order-to-cash process with lanes

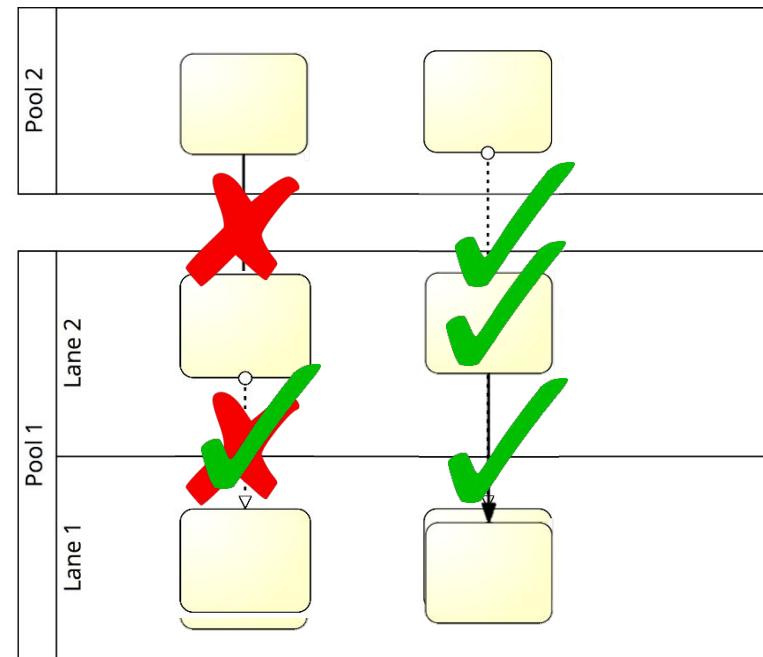


One more guideline...

- Start modeling with one single “white-box” pool
 - Initially, put the events and tasks in only one pool – the pool of the party who is running the process
 - Leave all other pools “black-boxed”
 - Once you have modeled this way, and once the process diagram inside the white-box pool is complete, you can model the details (events and tasks) in the other pools if that is useful.
 - In this course we will only model processes with one single white-box pool – all other pools are black-box
 - BPMN diagram with 2 or more pools is called collaboration diagram

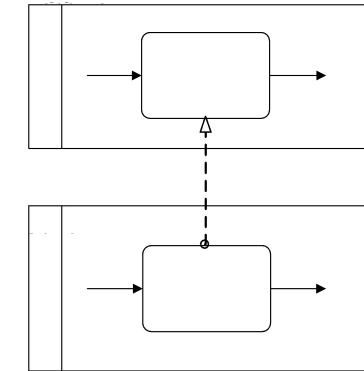
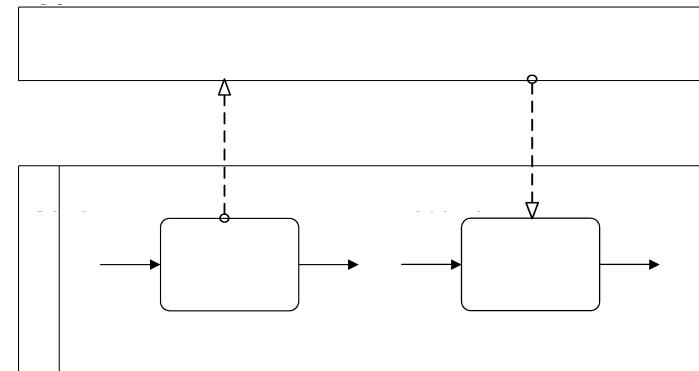
Pools, Lanes and Flows: syntactic rules

1. A Sequence Flow **cannot** cross the boundaries of a Pool (message flows can)
2. Both Sequence Flow and Message Flow **can cross** the boundaries of Lanes
3. A Message Flow **cannot connect** two flow elements within the same pool



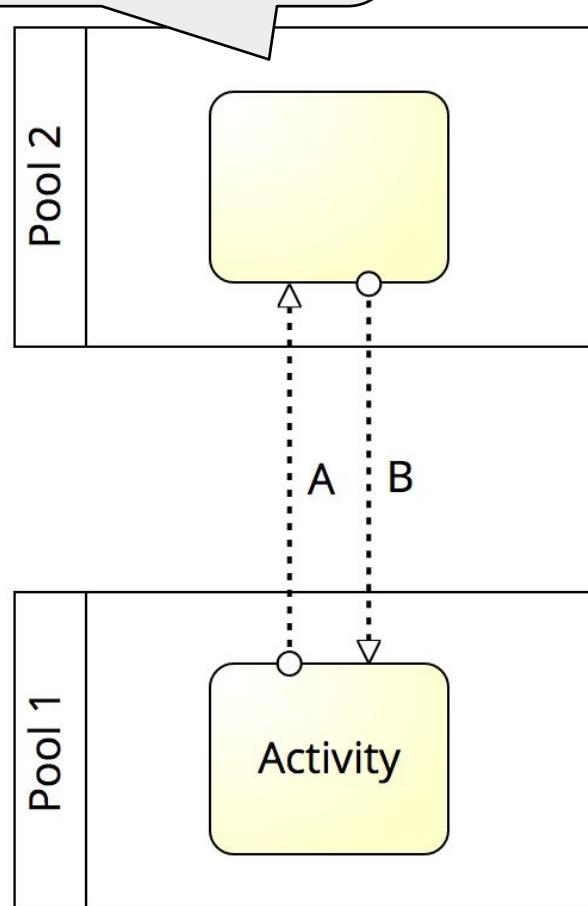
When are messages sent or received?

- A **Send** activity will send the outgoing message upon activity completion
- A **Receive** activity won't start until the incoming message has been received

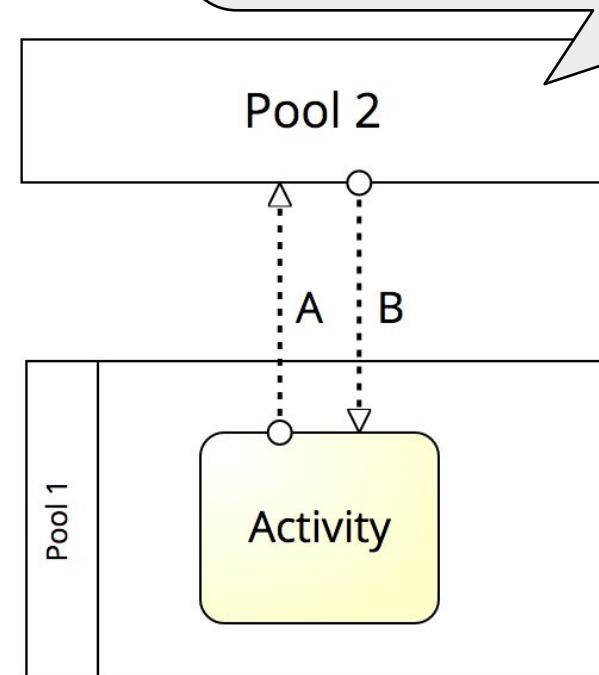


When are messages sent or received?

- Message B is first received before Activity can start.
- Message A is sent after, upon Activity's completion



- First, message B is received, before Activity can start.
- Then, message A is sent, upon Activity's completion



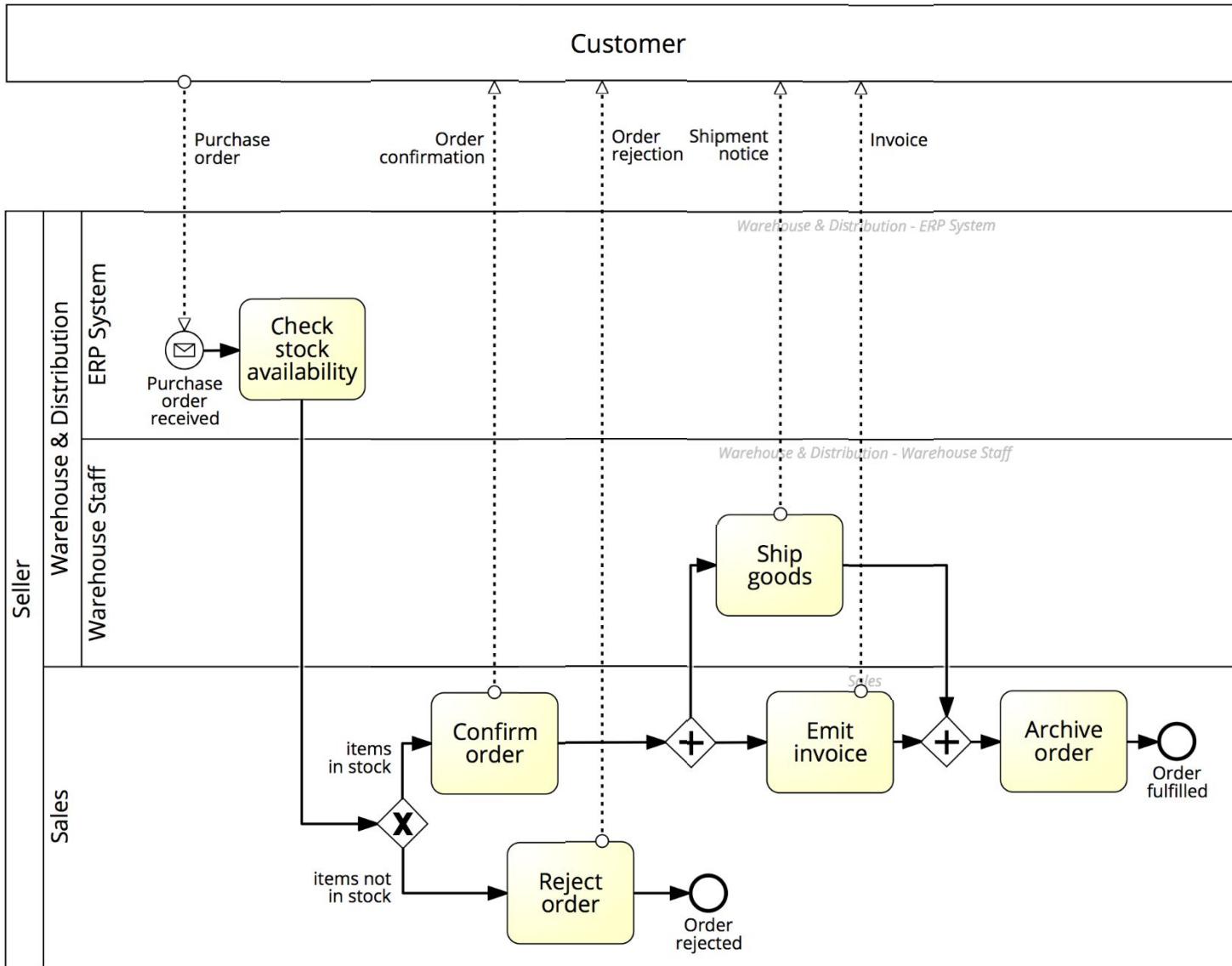
Exchanging information between business parties

Order-to-cash

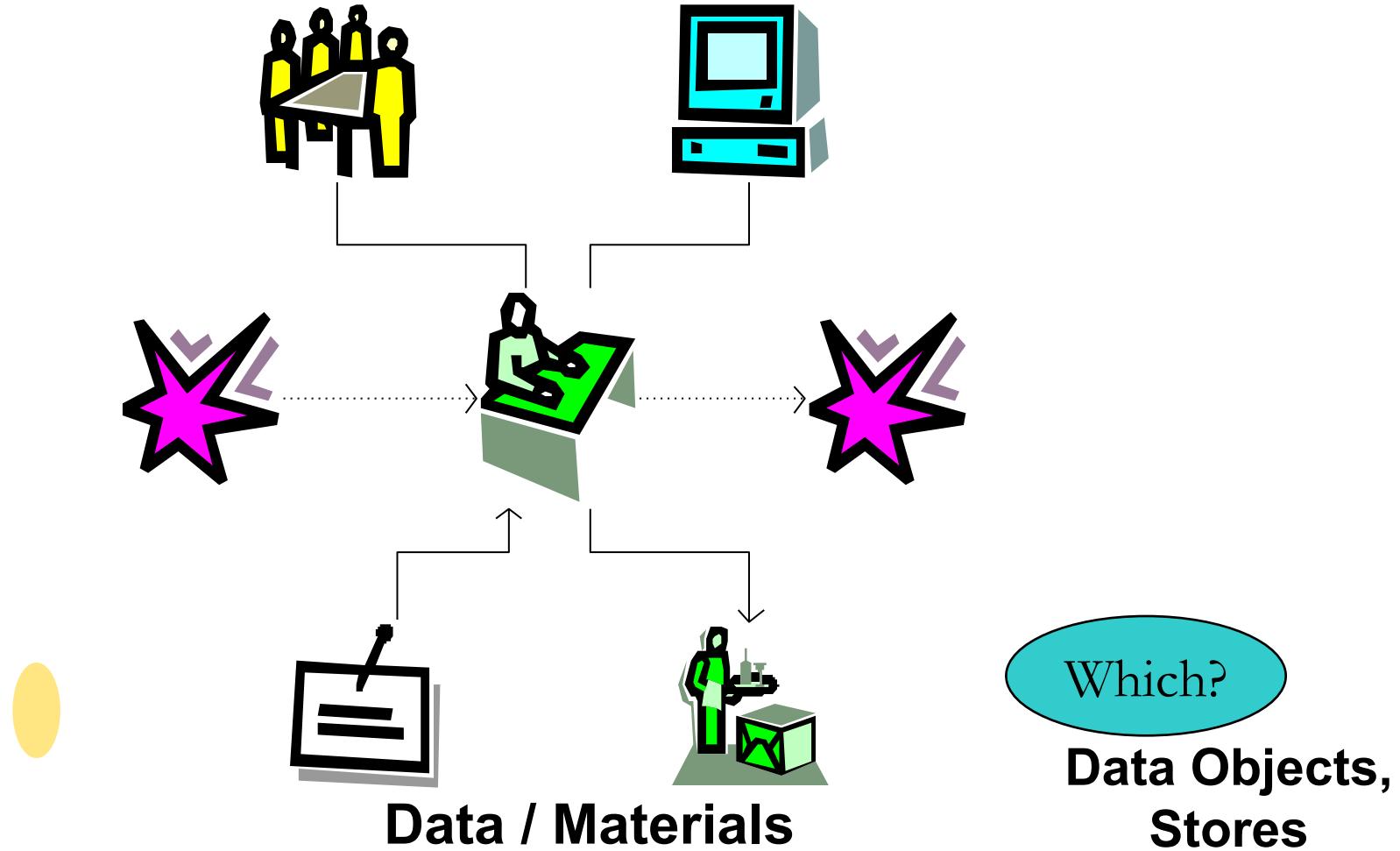
The purchase order **sent by the Customer** is received by the Seller and checked against the stock. This is done via an ERP module within the Warehouse & Distribution department. If the purchase order is not confirmed, the Sales department **sends an order rejection to the Customer**, otherwise it **sends an order confirmation**.

Next, the Warehouse & Distribution department ships the goods and **sends a shipment notification to the Customer**. Meantime, the Sales department emits the invoice and **sends it to the Customer**. The process concludes with the order being archived by the Sales department.

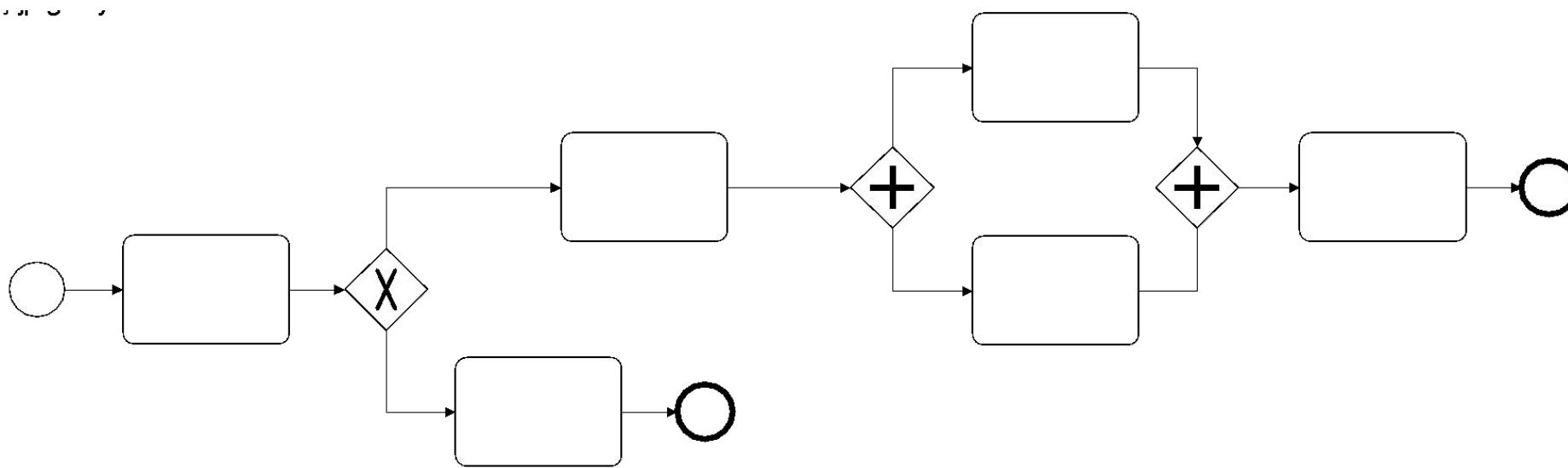
Order-to-cash process with a black-box customer pool



Process Modelling Viewpoints

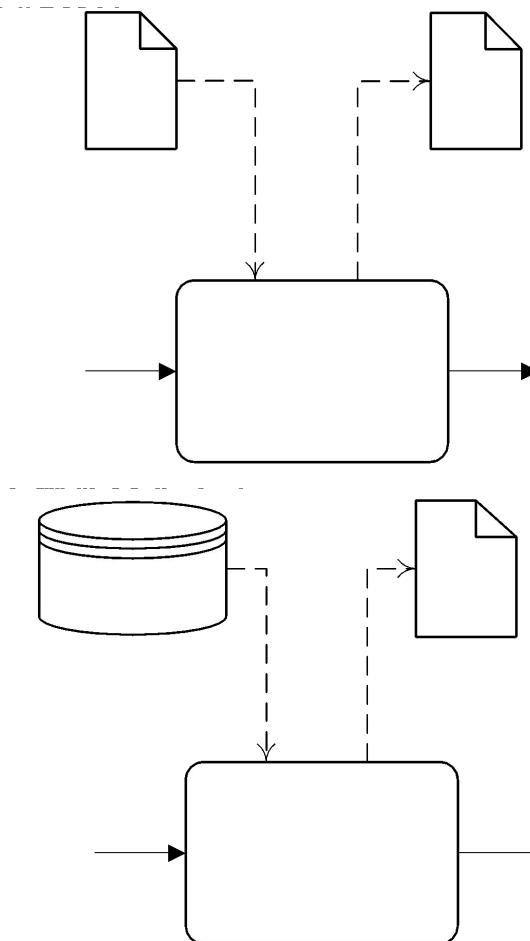


Order-to-cash process, again



The purchase order document serves as an input to the stock availability check. Based on the outcome of this check, the status of the document is updated, either to “approved” or “rejected”. If the order is approved, an invoice and a shipment notice are produced.

BPMN Information Artifacts

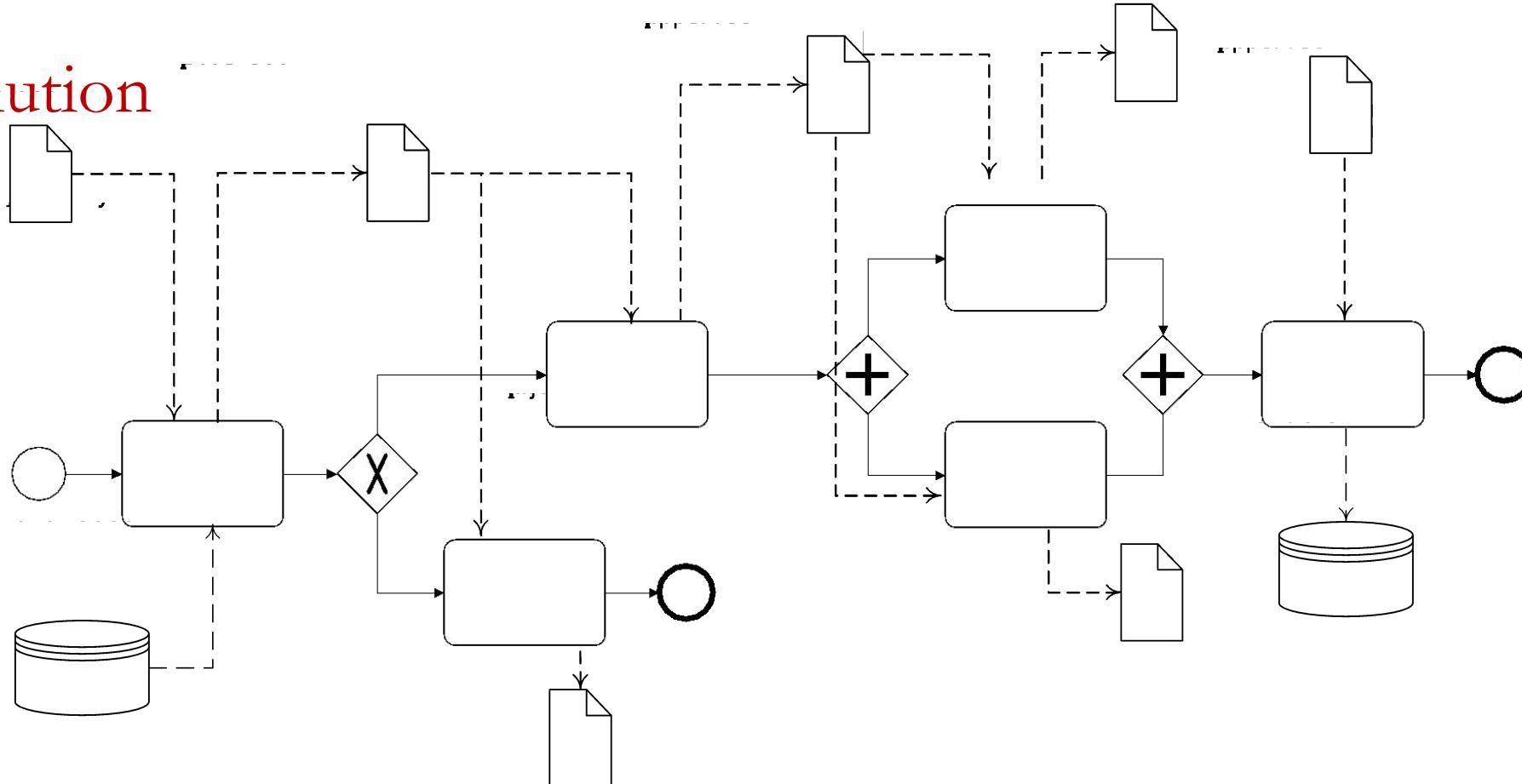


A *Data Object* captures an artifact required (input) or produced (output) by an activity.

- Can be physical or electronic

A *Data Store* is a place containing data objects that must be persisted beyond the duration of a process instance.
It is used by an activity to store (as output) or retrieve (as input) data objects.

Solution

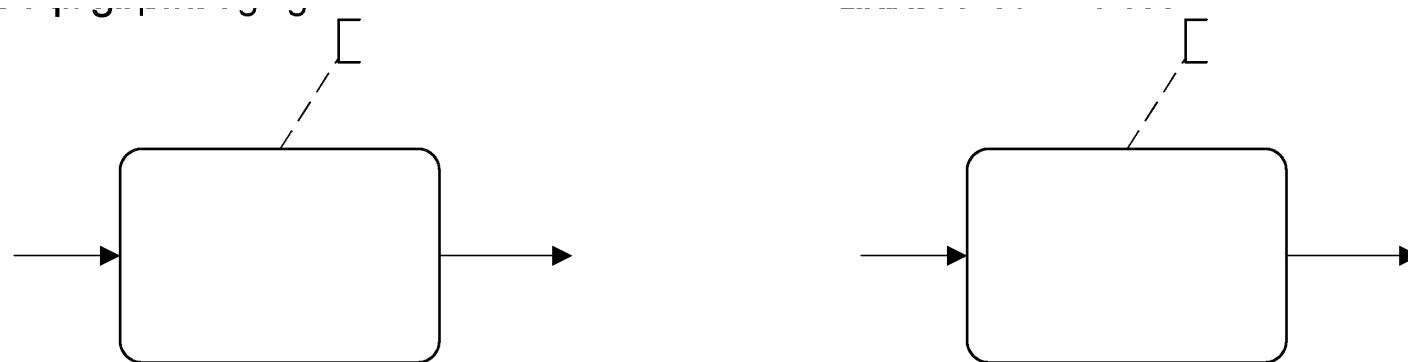


The purchase order document serves as an input to the stock availability check against the Warehouse DB. Based on the outcome of this check, the status of the document is updated, either to “approved” or “rejected”. If the order is approved, an invoice and a shipment notice are produced. The order is then archived on the Orders DB.

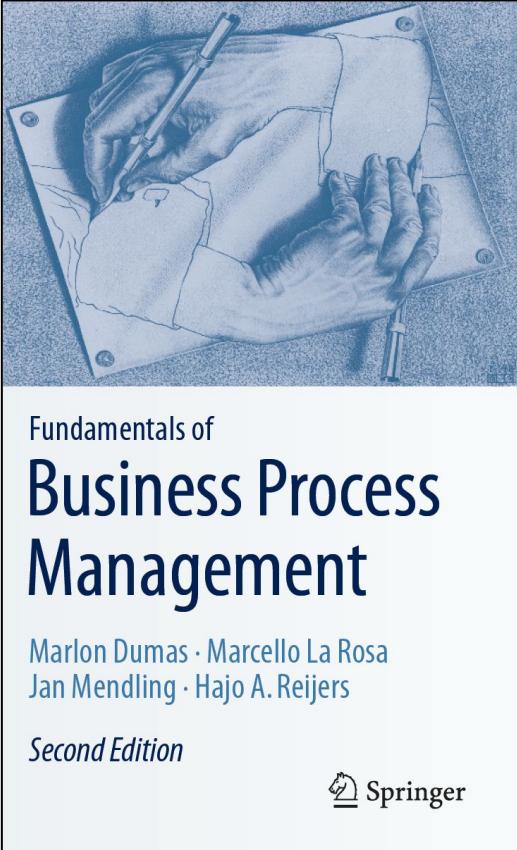
BPMN Text Annotations

A *Text Annotation* is a mechanism to provide additional text information to the model reader

- **Doesn't affect** the flow of tokens through the process



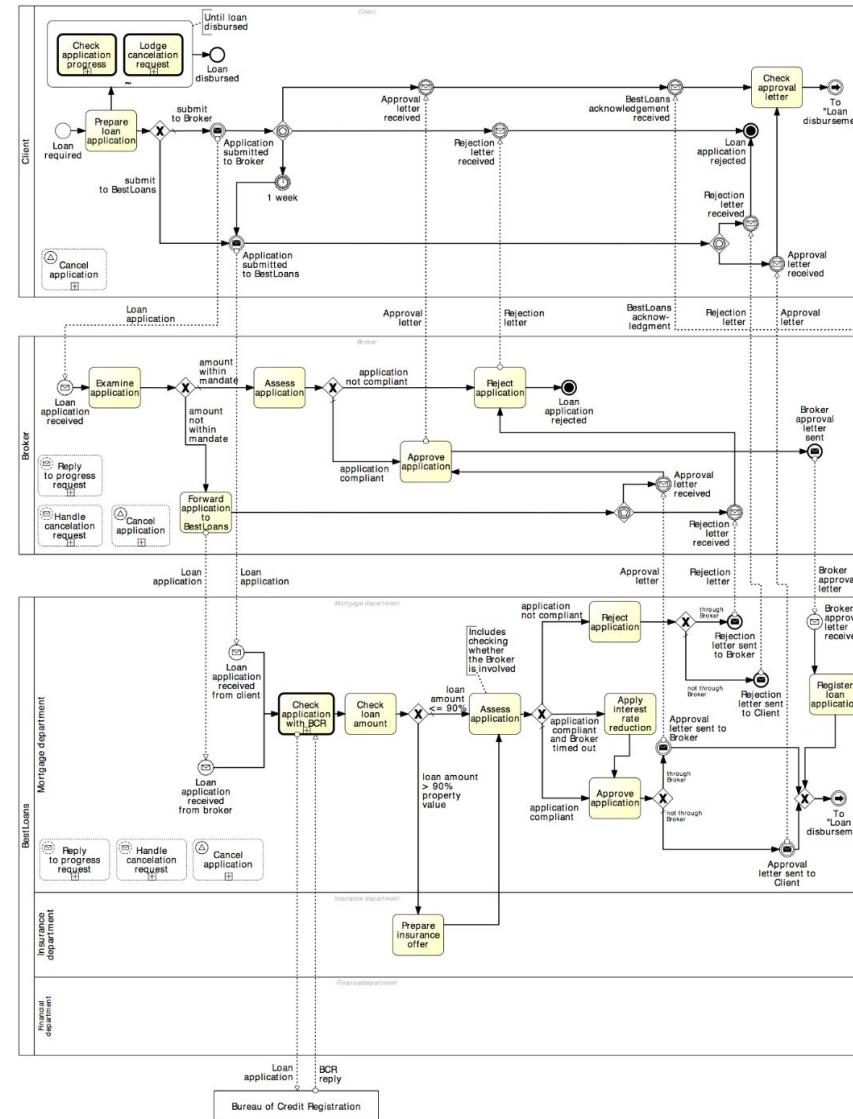
Chapter 3: Essential Process Modeling



Contents

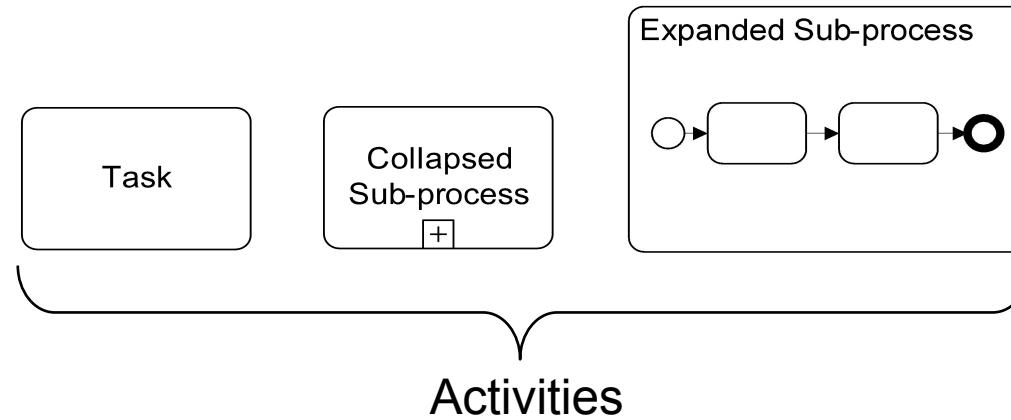
1. First Steps with BPMN
2. Branching and Merging
3. Business Objects
4. Resources
5. Process Decomposition
6. Process Model Reuse
7. Recap

Process decomposition



Process decomposition

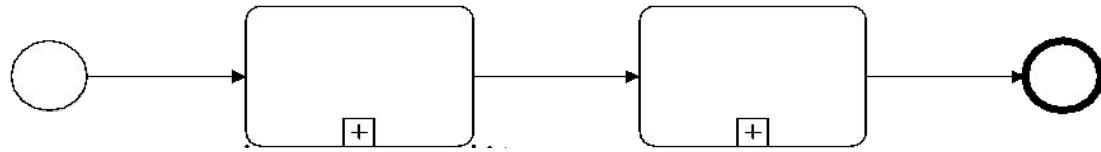
An activity in a process can be decomposed into a “sub-process”



Use this feature to:

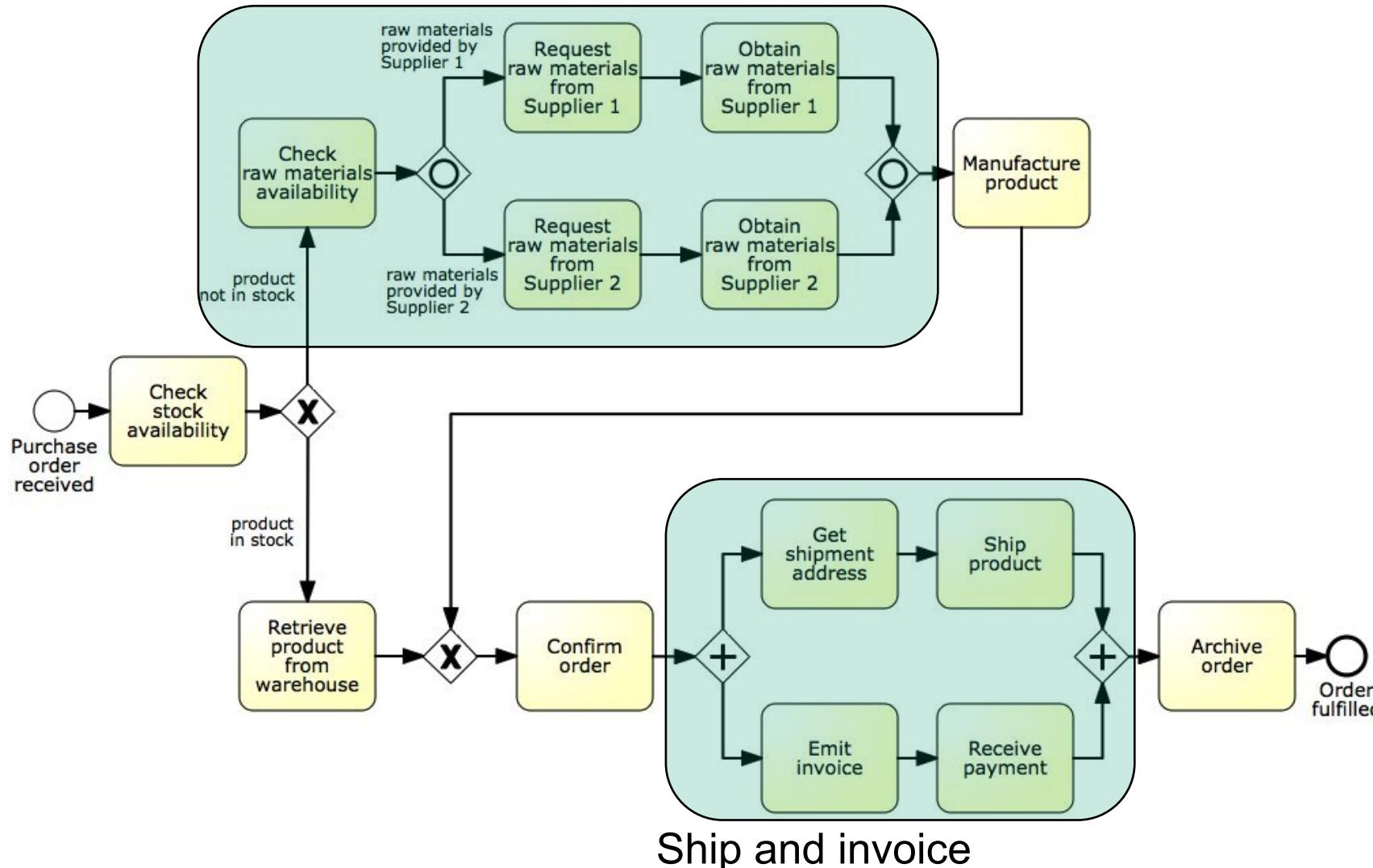
1. Improve understanding by breaking down large models
2. Identify parts that should be:
 - repeated
 - executed multiple times in parallel
 - interrupted, or
 - compensated

Example: Sub-Process

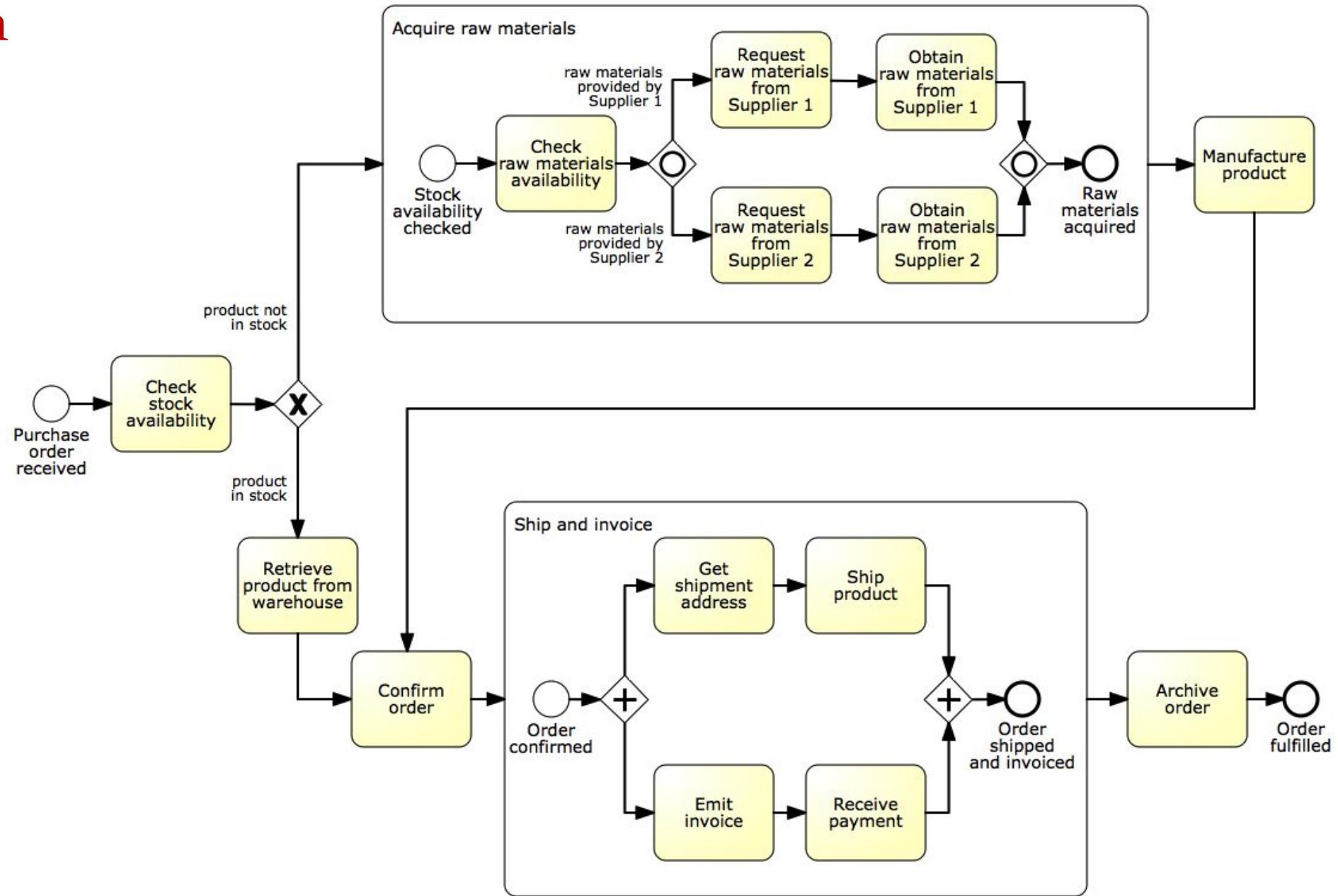


Identify possible sub-processes

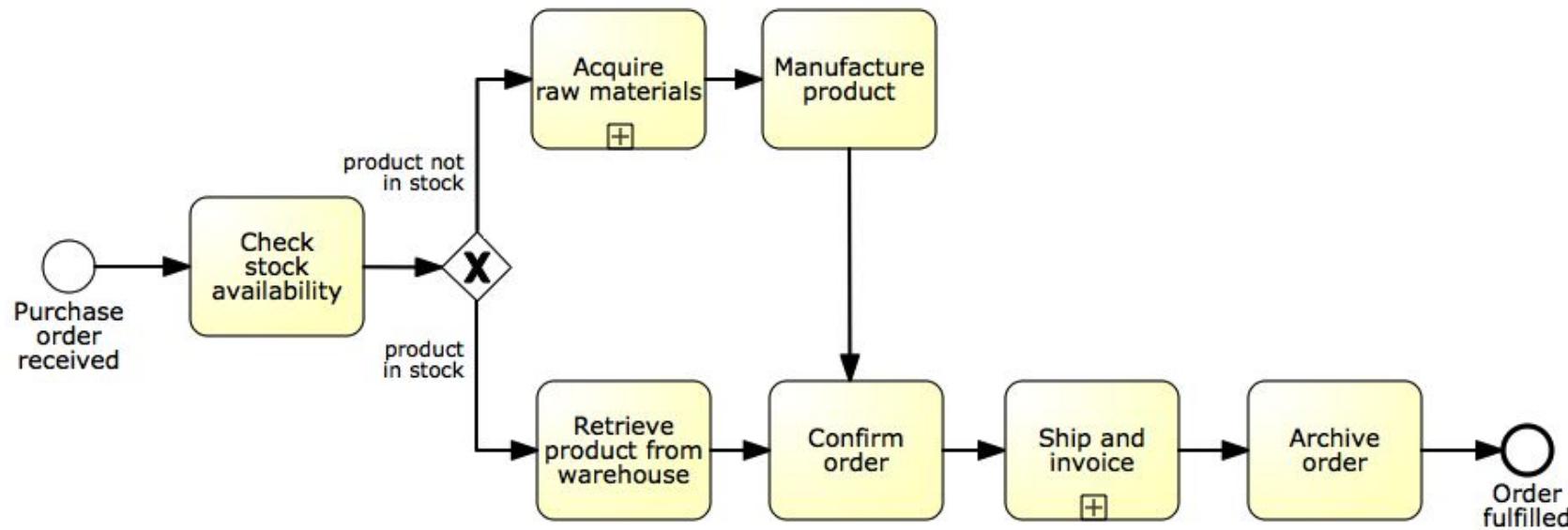
Acquire raw materials



Solution

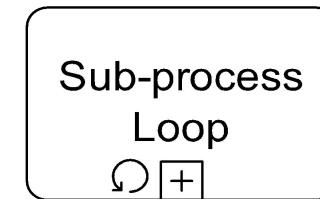
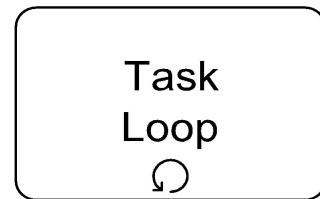


The refactored model

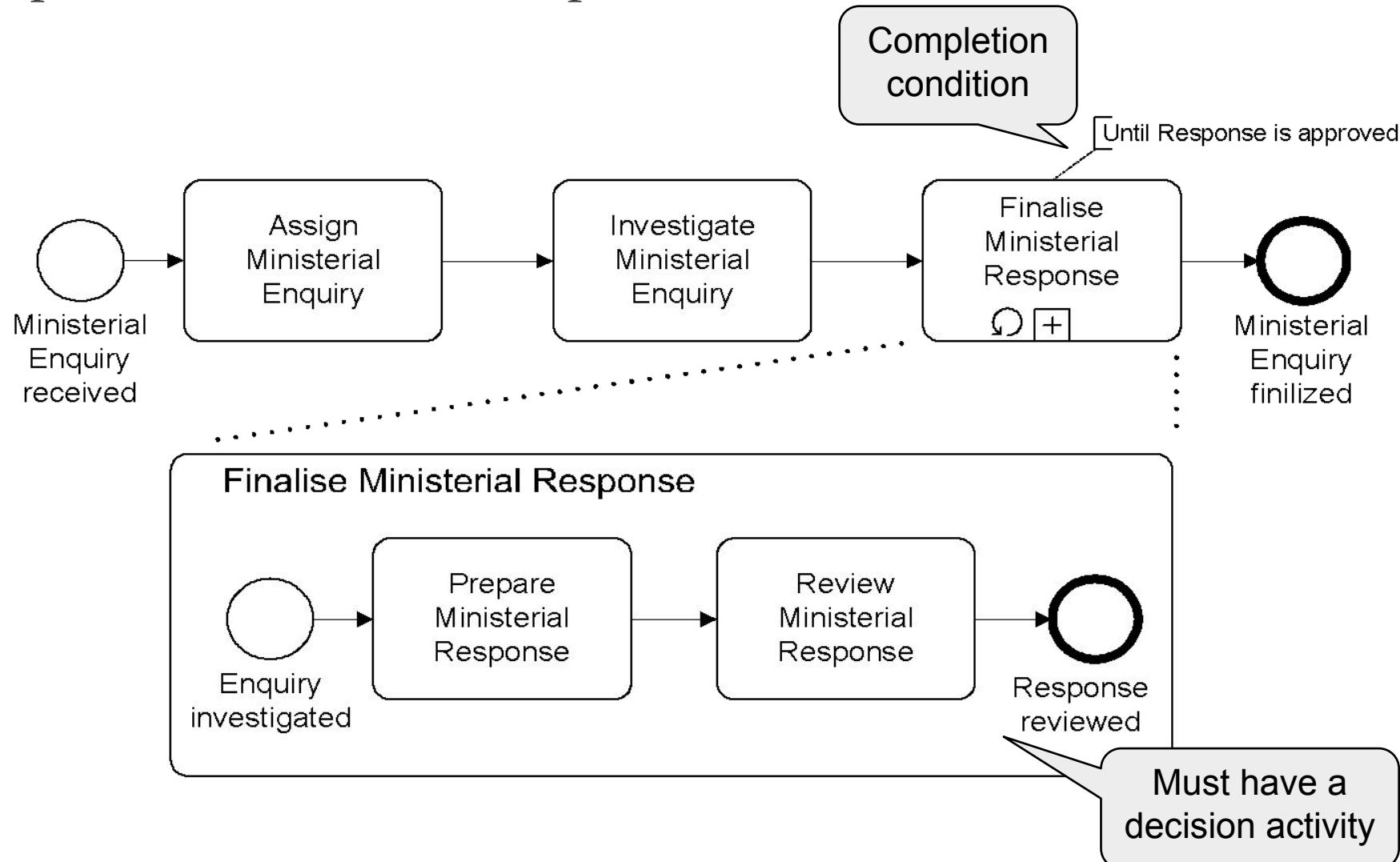


Block-structured repetition: Loop Activity

BPMN also provides the *loop activity* construct to allow the repetition of a task or sub-process



Example: block-structured repetition



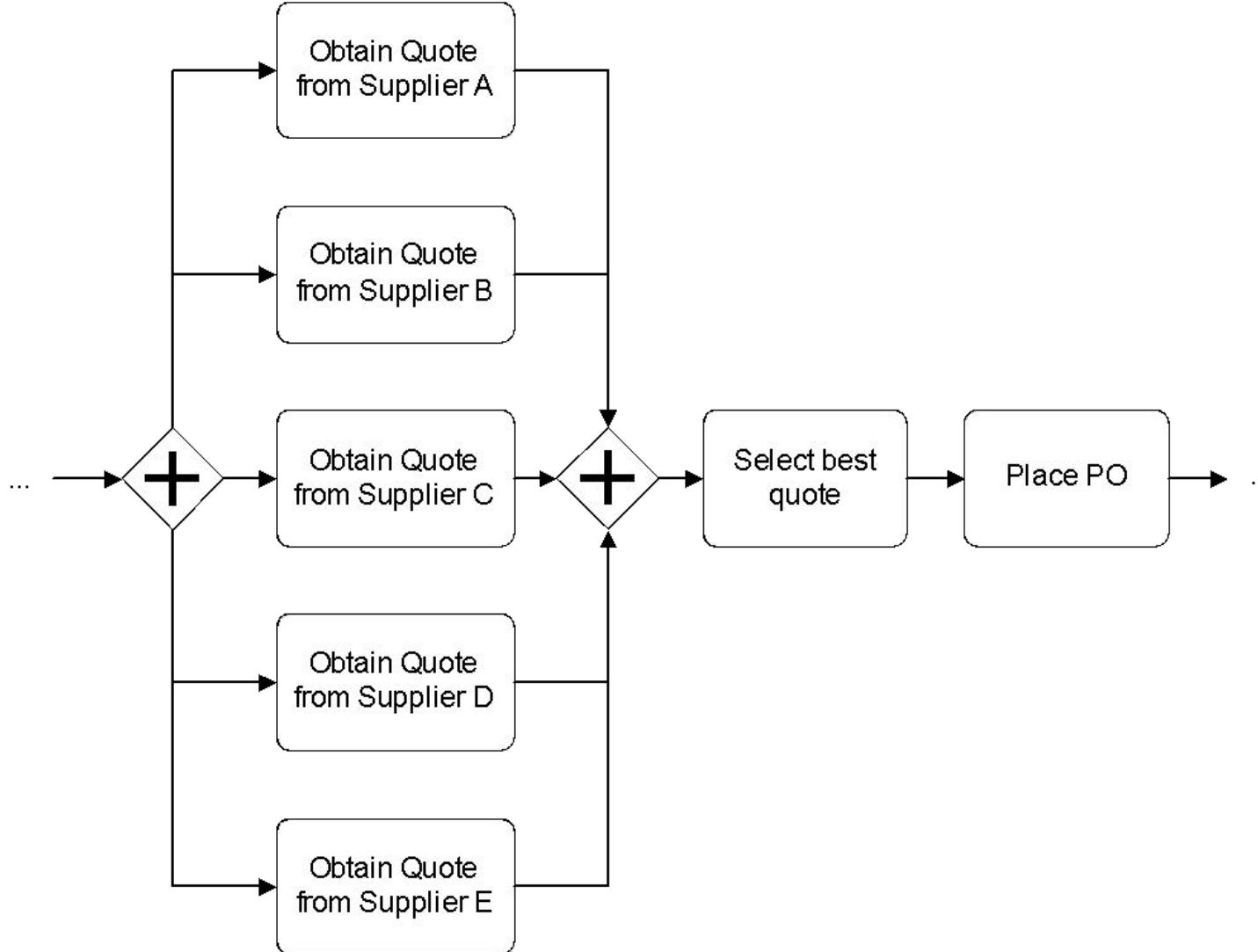
Example: multi-instance activity

Procurement

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.

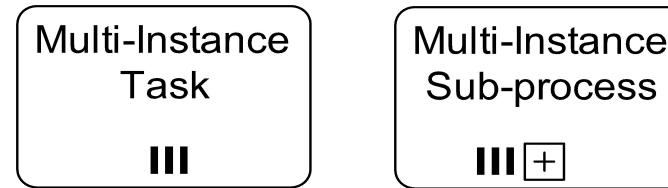
Solution: without multi-instance activity

Procurement



Parallel repetition: multi-instance activity

Provides a mechanism to indicate that an activity is executed *multiple times concurrently*

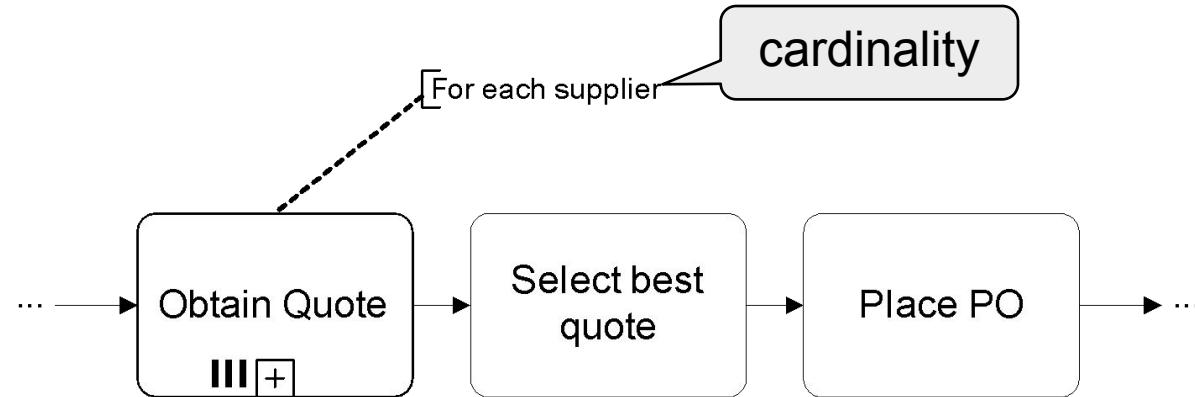


Useful when the same activity needs to be executed for multiple entities or data items, such as:

- Request quotes from multiple suppliers
- Check the availability for each line item in an order separately
- Send and gather questionnaires from multiple witnesses in the context of an insurance claim

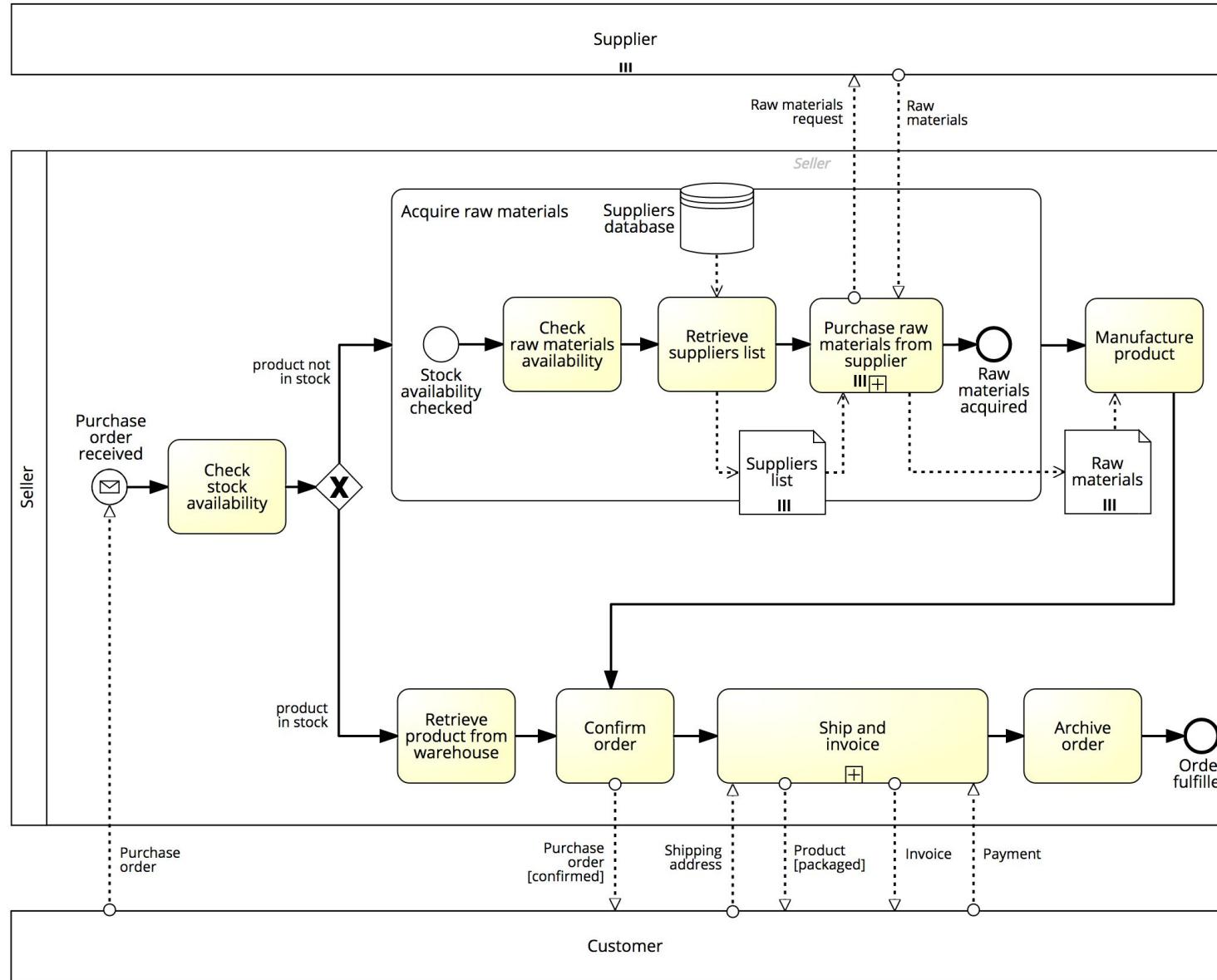
Solution: with multi-instance activity

Procurement



Our order-to-cash example...

now with pools, messages and MI markers



BPMN Poster (link in “Readings” page)

