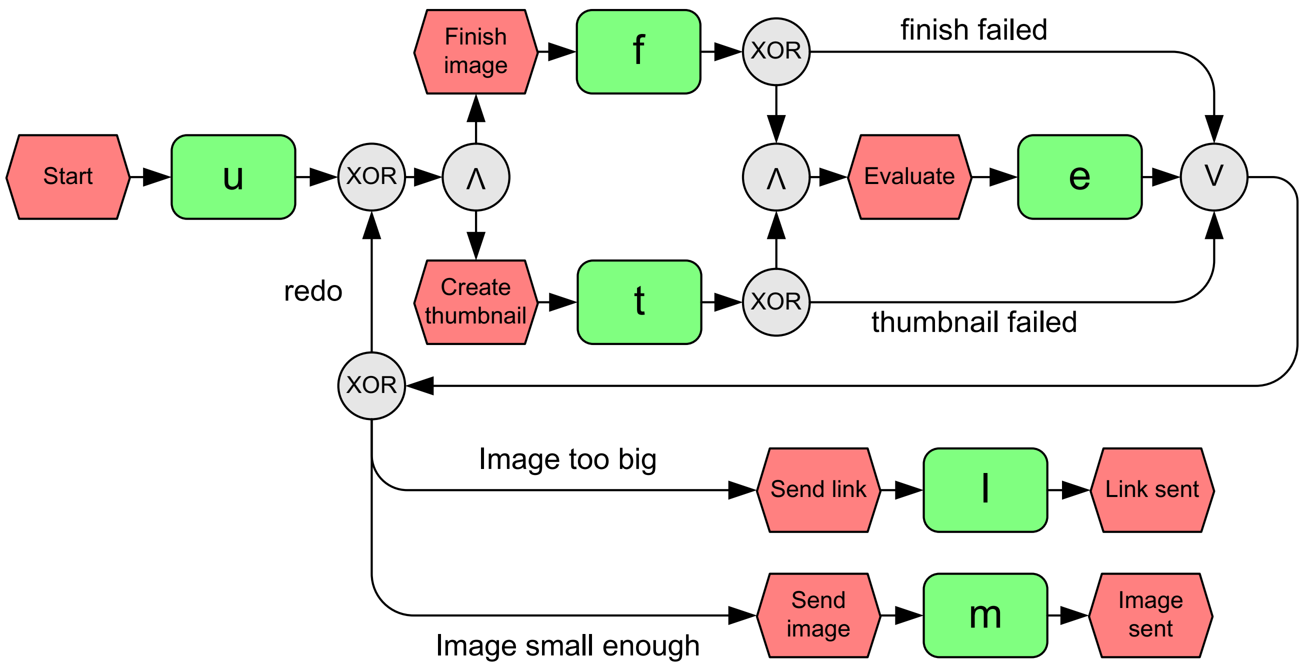
Business Processes Modelling

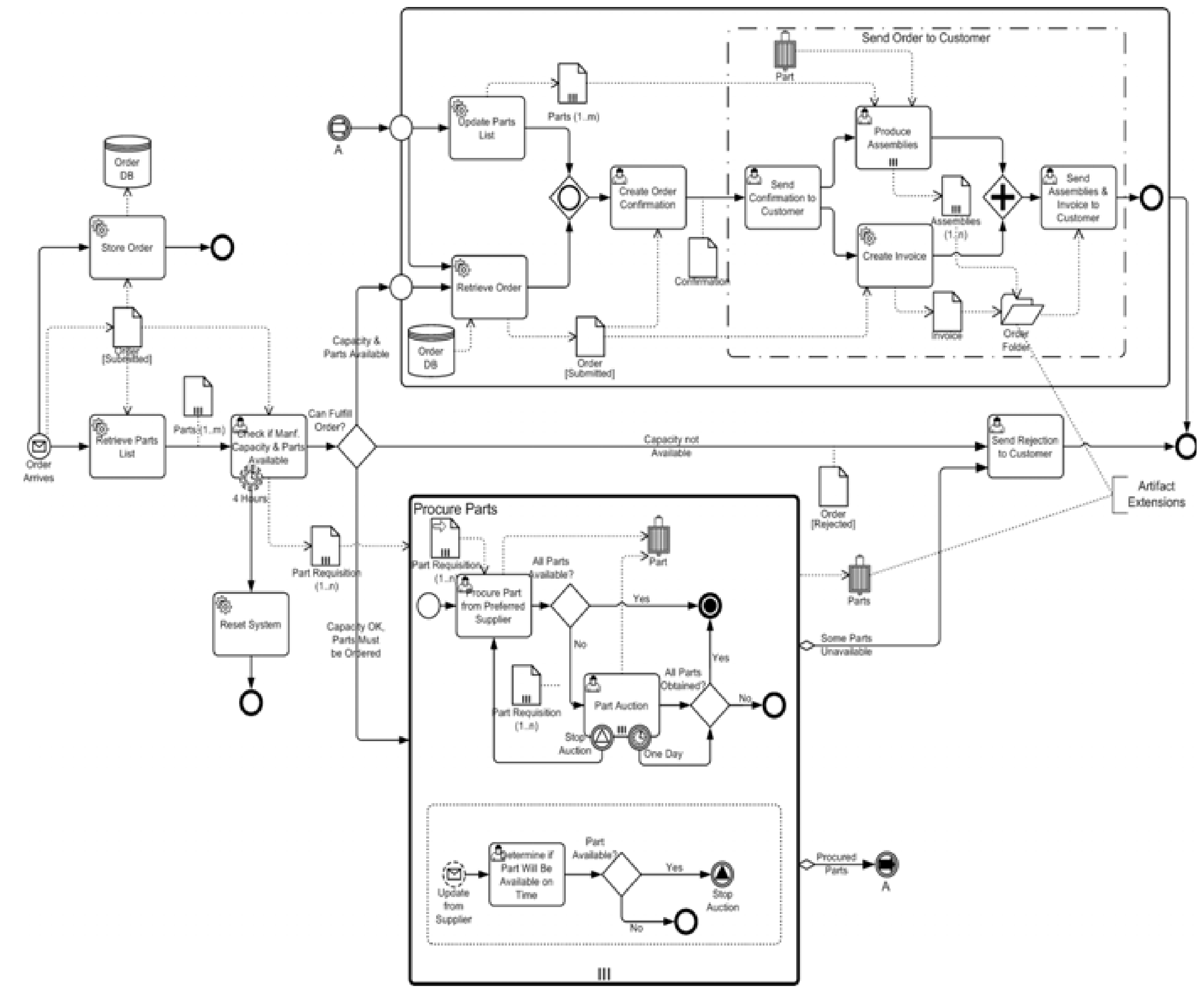
MPB (6 cfu, 295AA)

Roberto Bruni
http://www.di.unipi.it/~bruni

07 - EPC and BPMN

Object

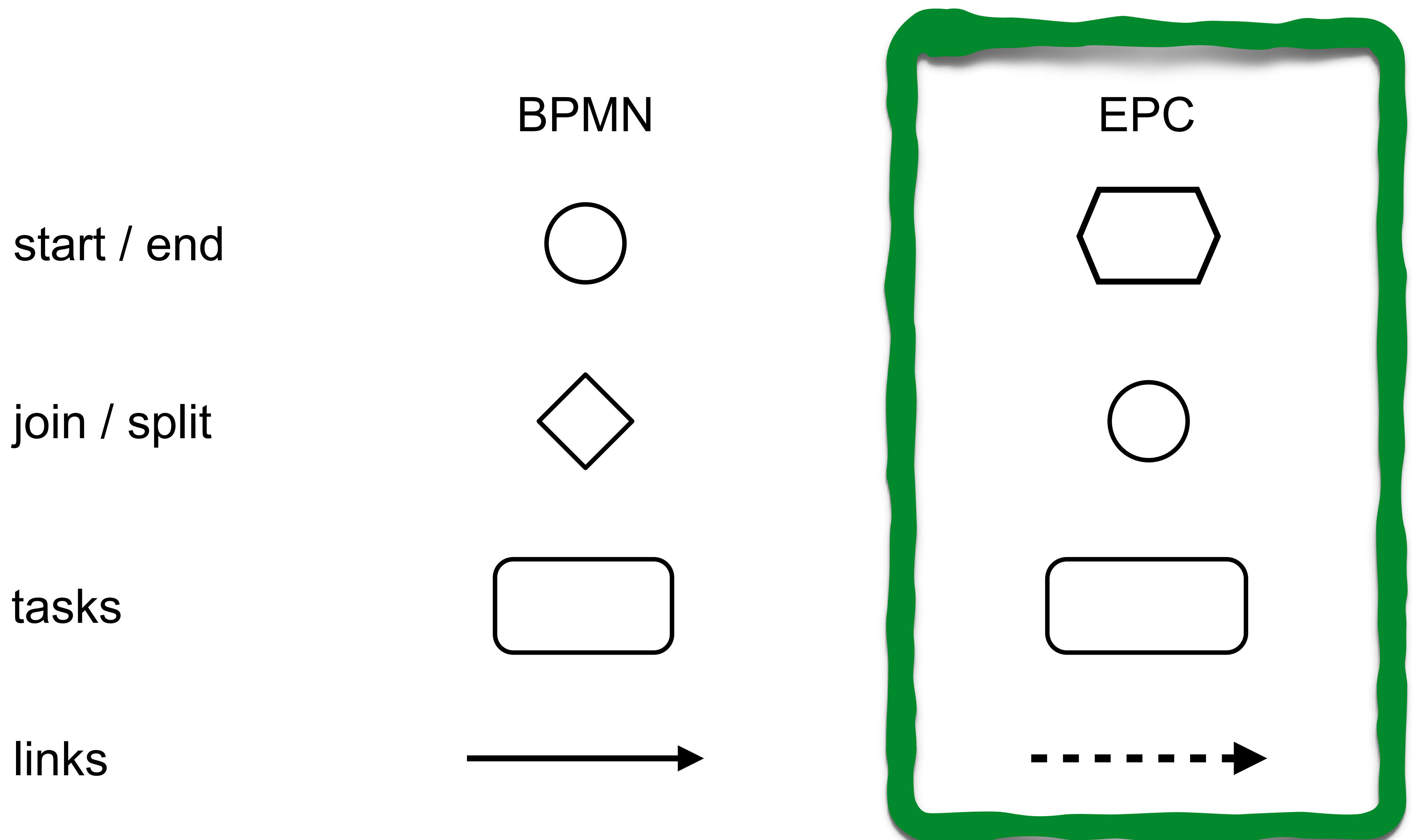




We overview high-level diagrammatic notation

Ch.4.3,4.7, 5.7 of Business Process Management: Concepts, Languages, Architectures
Ch.3, 4 of Fundamental of Business Process Management. M. Dumas et al.

A closer look at standards: EPC and BPMN



EPC Diagrams: Requirements

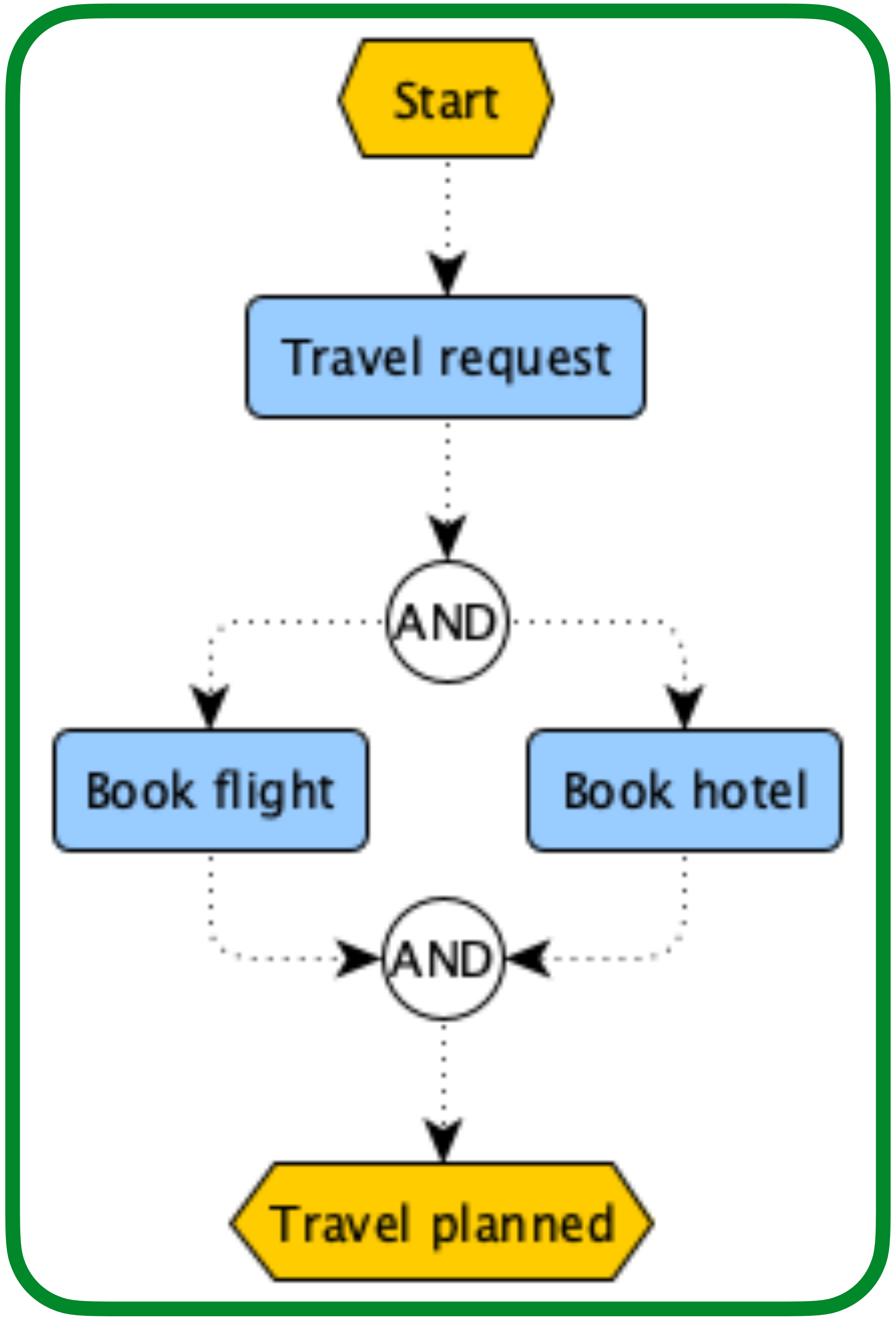
EPC diagrams: requirements

EPC elements can be combined in a fairly free manner
(possibly including cycles)

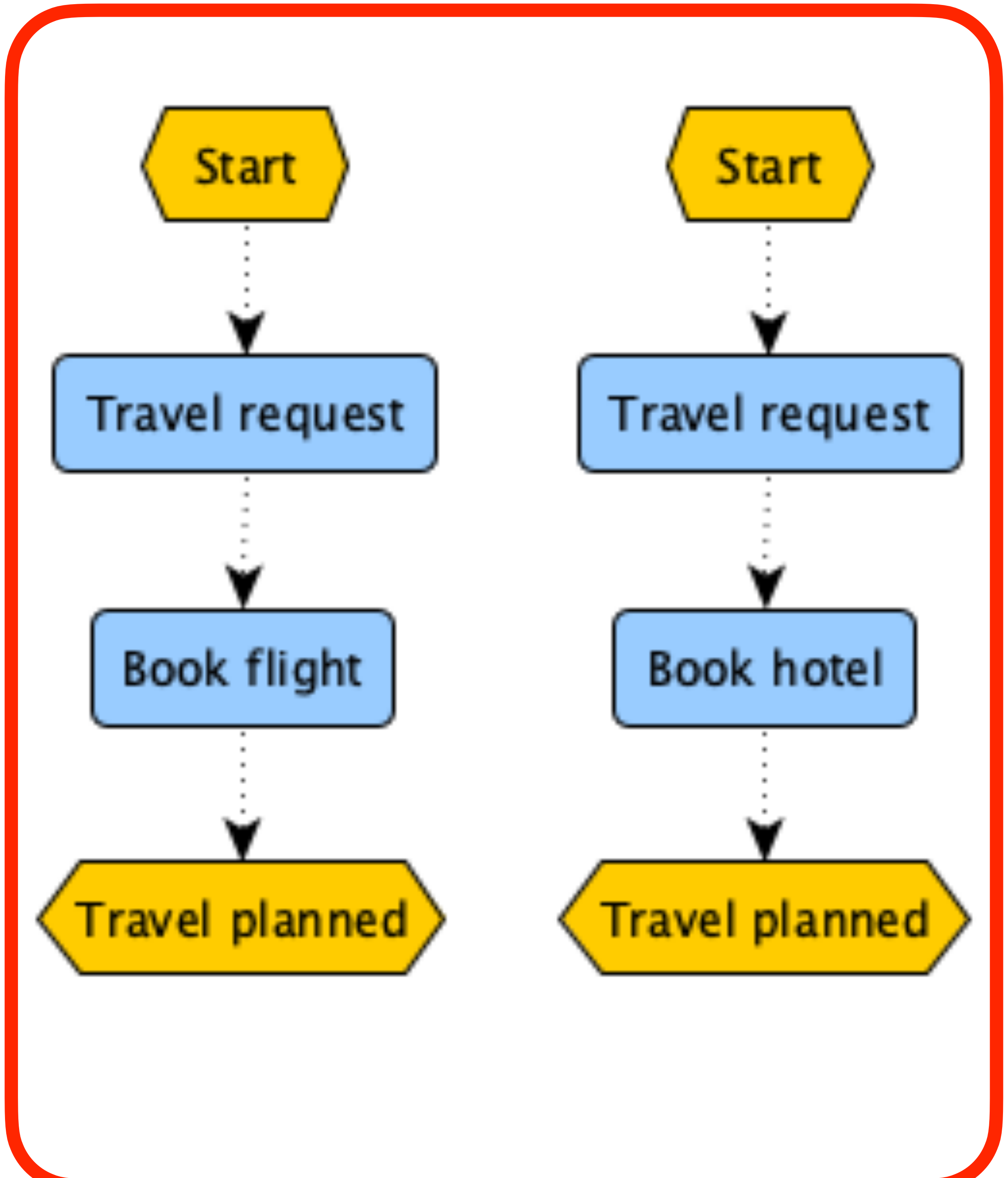
The graph must be **weakly connected** (e.g., no isolated nodes)

Weak connectivity

Weakly connected



Non weakly connected



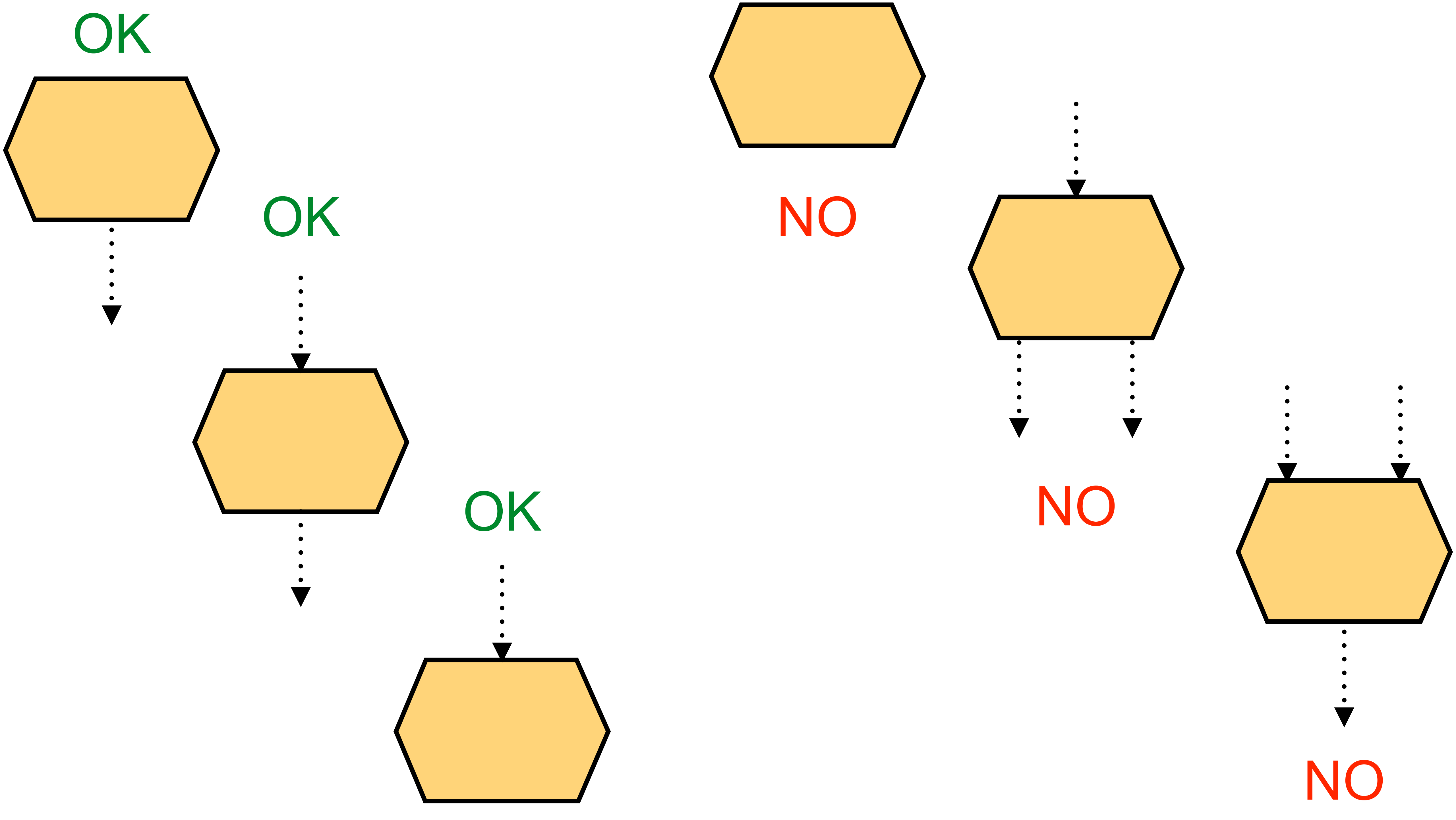
EPC diagrams: requirements

EPC elements can be combined in a fairly free manner
(possibly including cycles)

The graph must be **weakly connected** (e.g., no isolated nodes)

Events have at most one incoming and one outgoing arc
Events have at least one incident arc
There must be at least one start event and one end event

Event connectivity



EPC diagrams: requirements

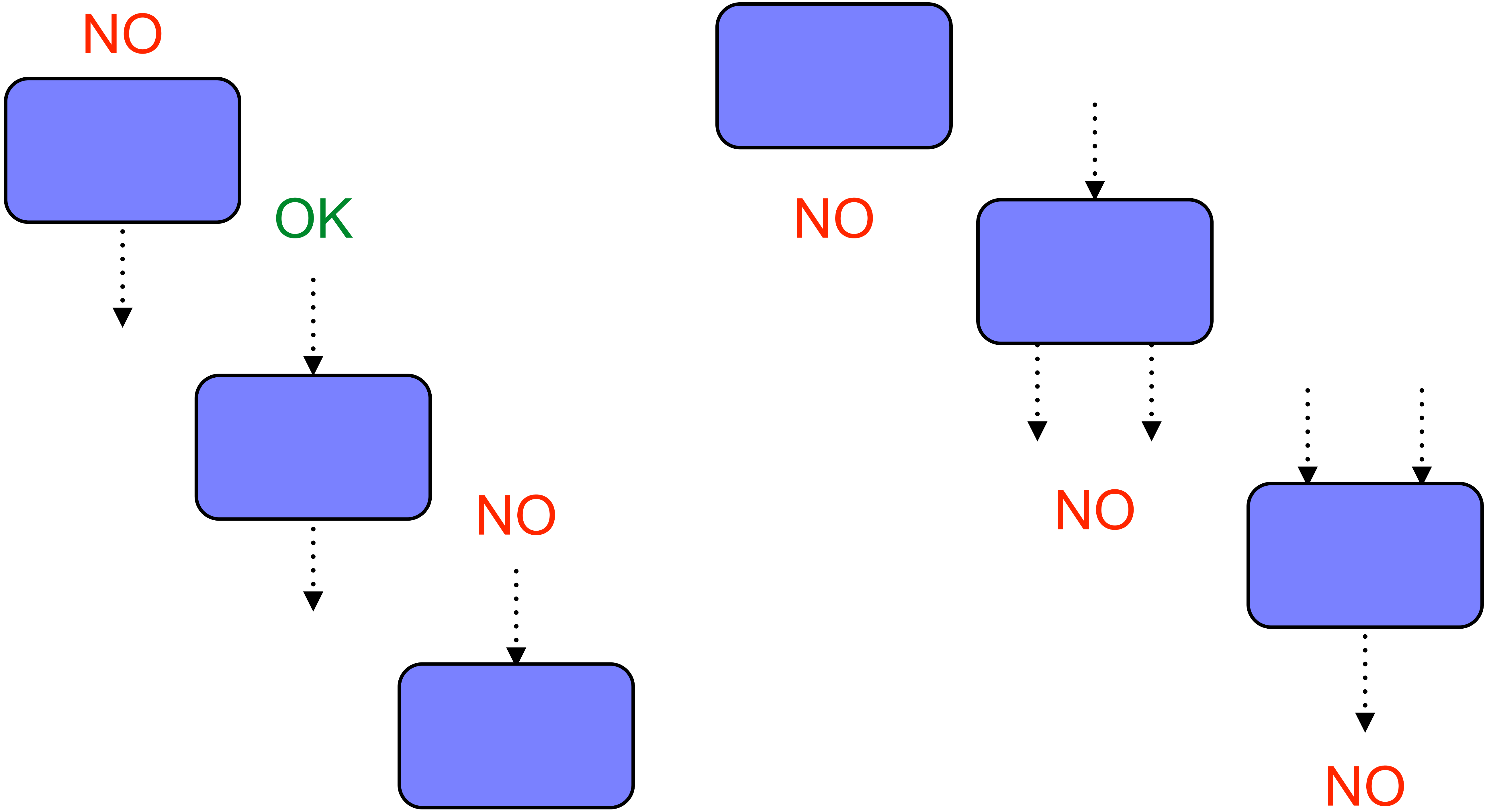
EPC elements can be combined in a fairly free manner
(possibly including cycles)

The graph must be **weakly connected** (e.g., no isolated nodes)

Events have at most one incoming and one outgoing arc
Events have at least one incident arc
There must be at least one start event and one end event

Functions have exactly one incoming and one outgoing arc

Function connectivity



EPC diagrams: requirements

EPC elements can be combined in a fairly free manner
(possibly including cycles)

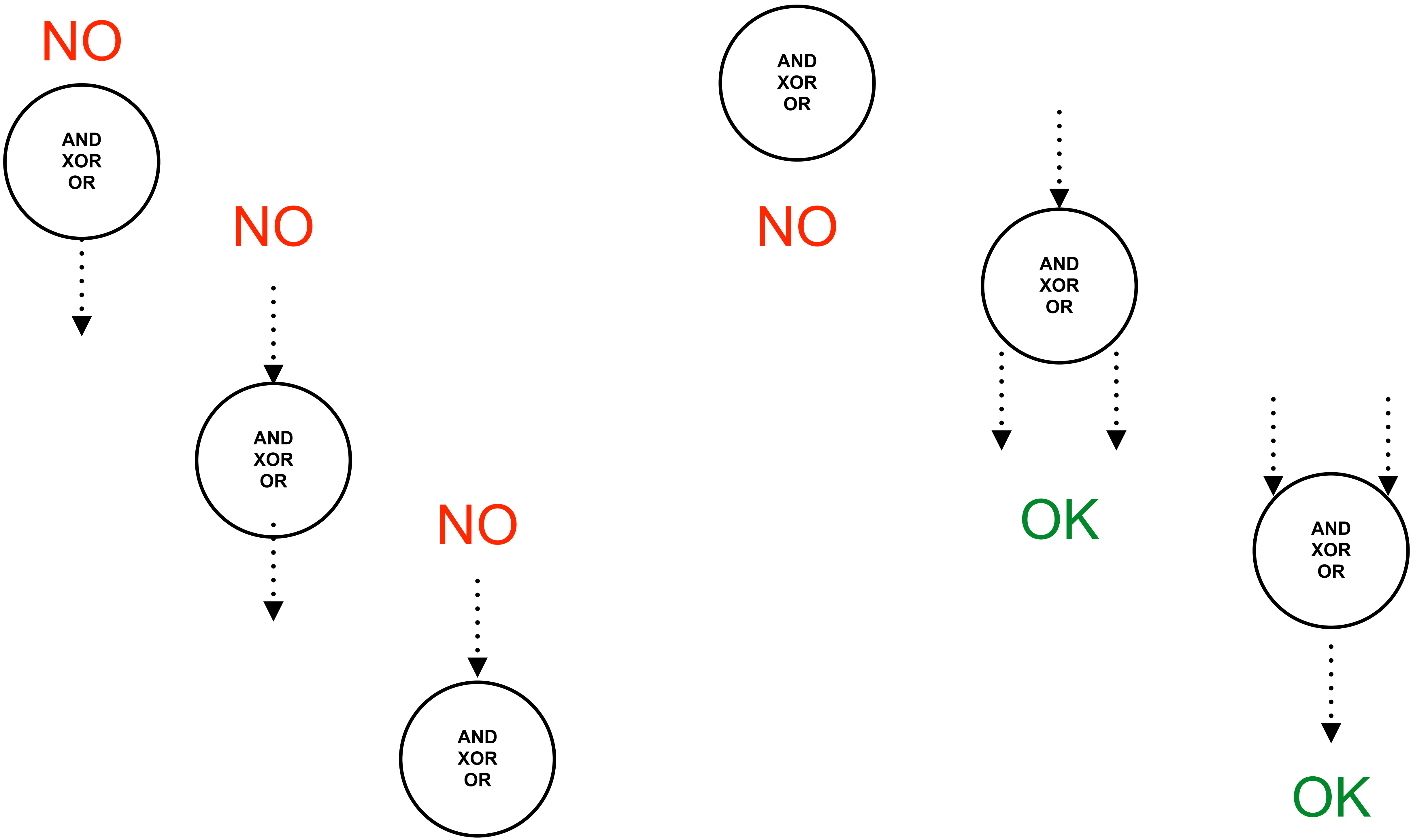
The graph must be **weakly connected** (e.g., no isolated nodes)

Events have at most one incoming and one outgoing arc
Events have at least one incident arc
There must be at least one start event and one end event

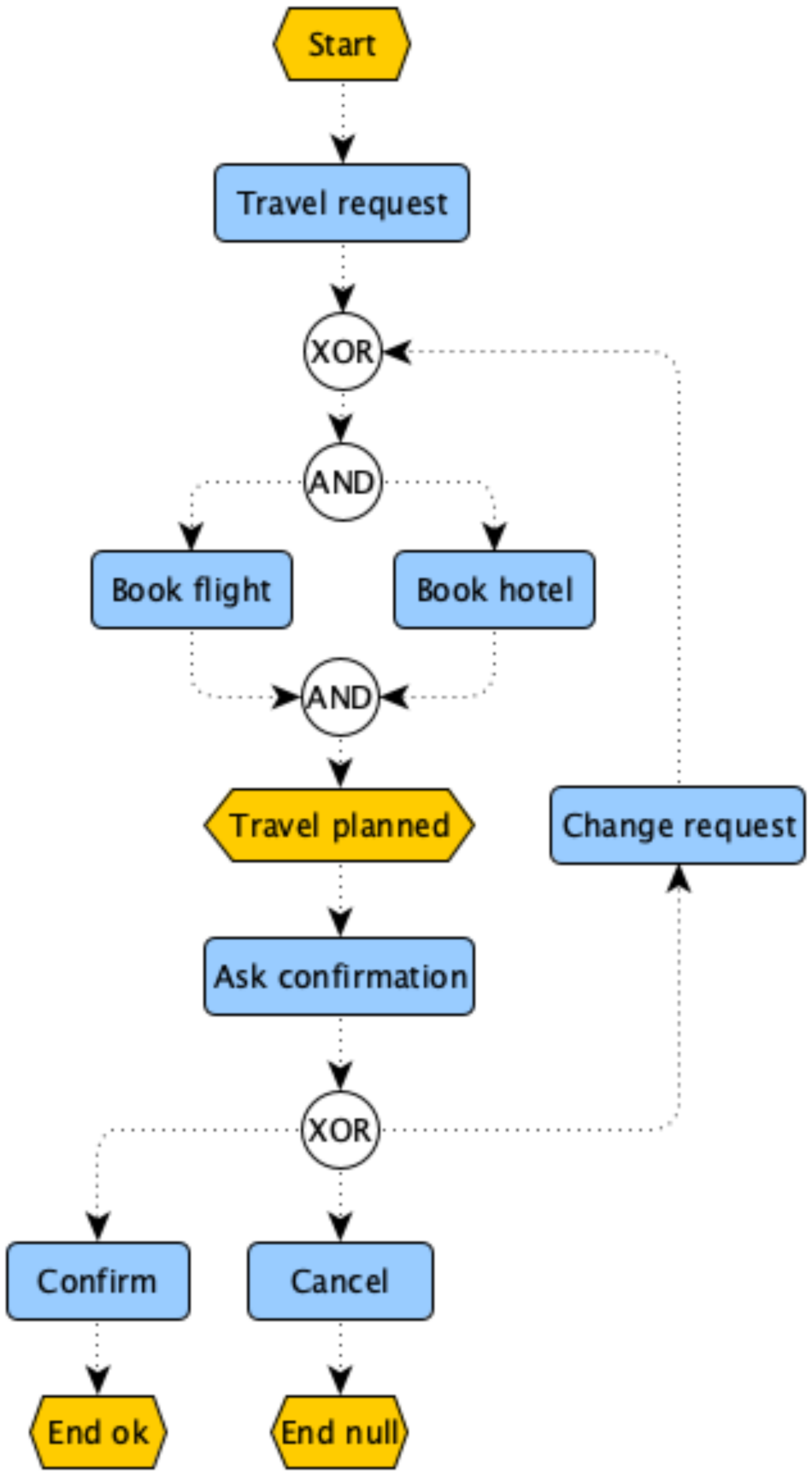
Functions have exactly one incoming and one outgoing arc

Connectors have either one incoming arc and multiple outgoing arcs
or viceversa (multiple incoming arcs and one outgoing arc)

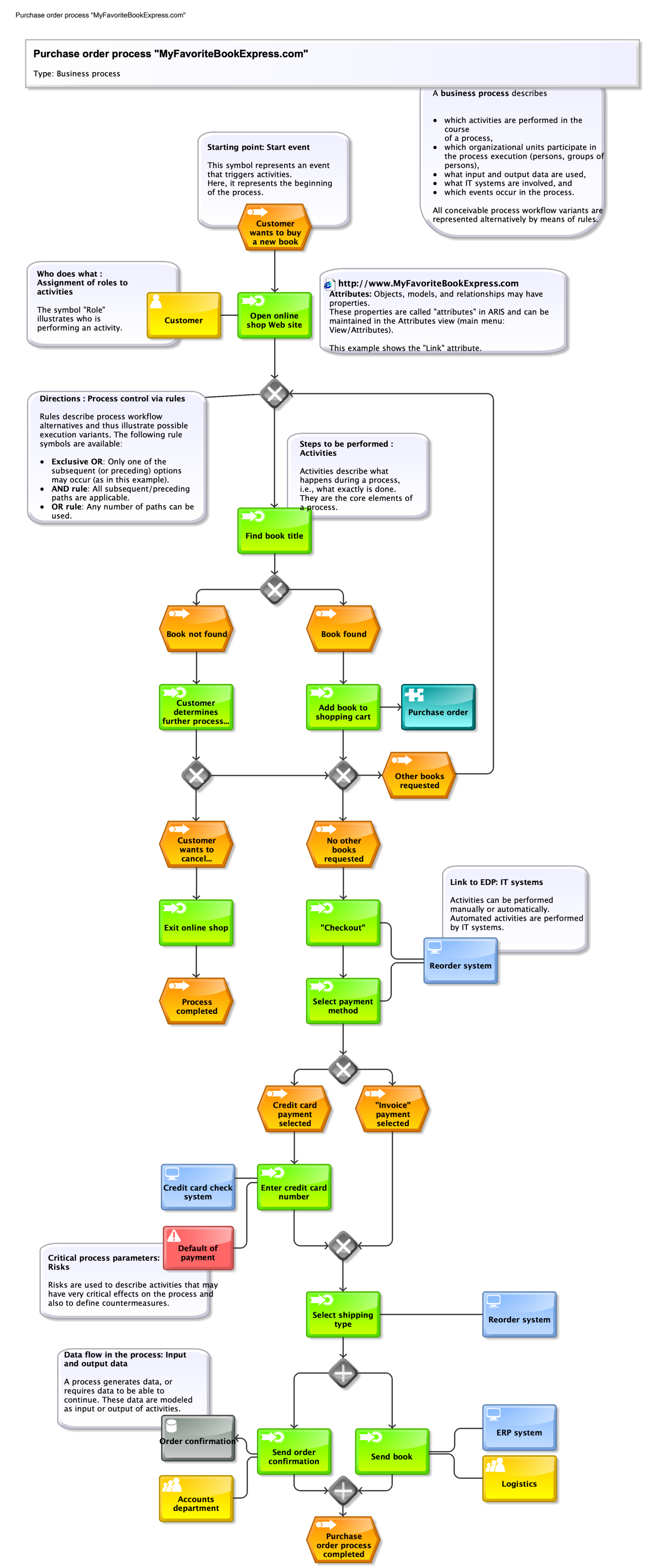
Split/Join connectivity



EPC: Example (yEd)



EPC: Example (ARIS Express)



EPC Diagrams: Guidelines

EPC Diagrams: guidelines

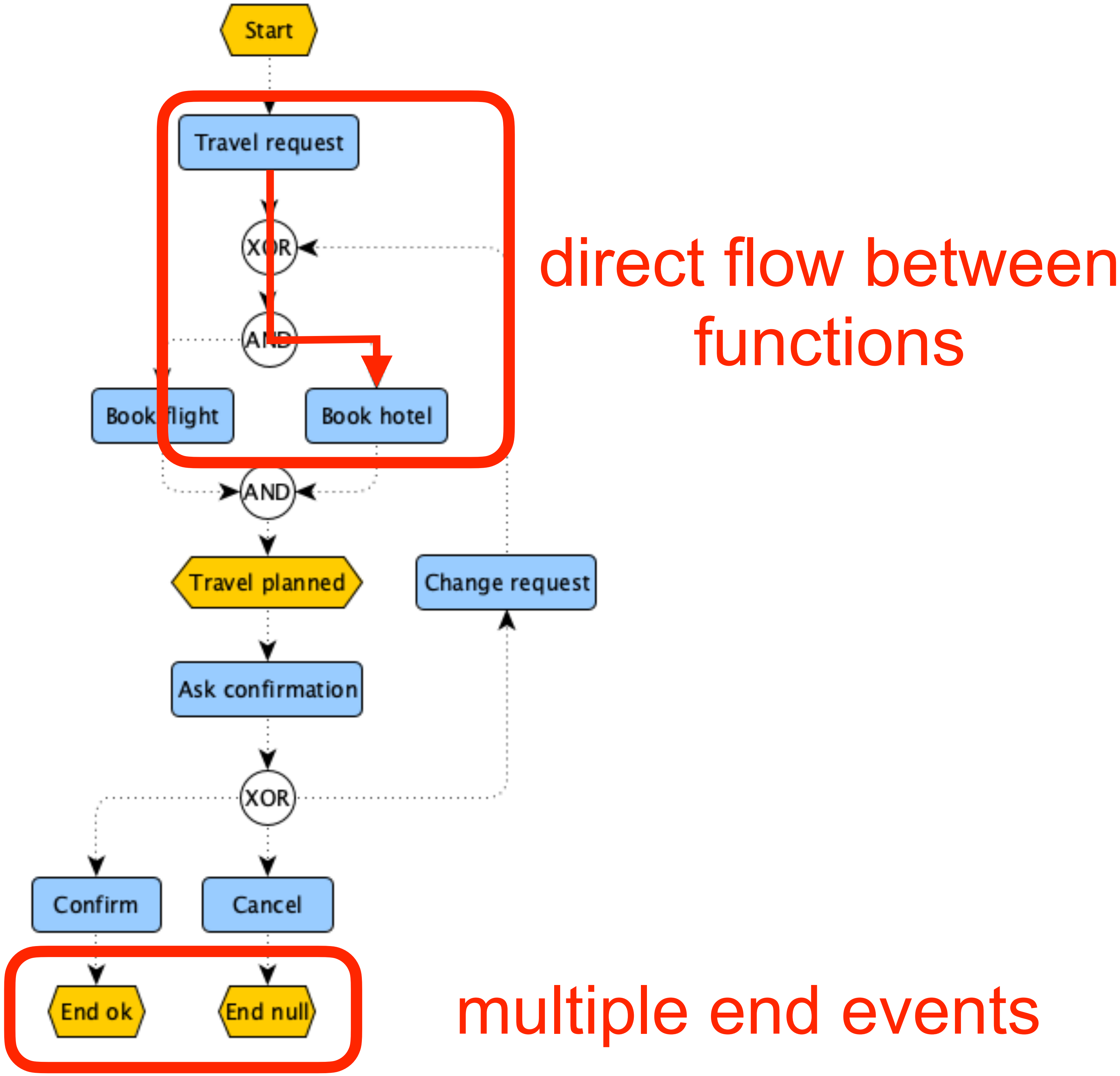
Other constraints are sometimes imposed

Unique start / end event

No direct flow between two events
No direct flow between two functions

No event is followed by a decision node
(i.e. (X)OR-split)

EPC guidelines: Example

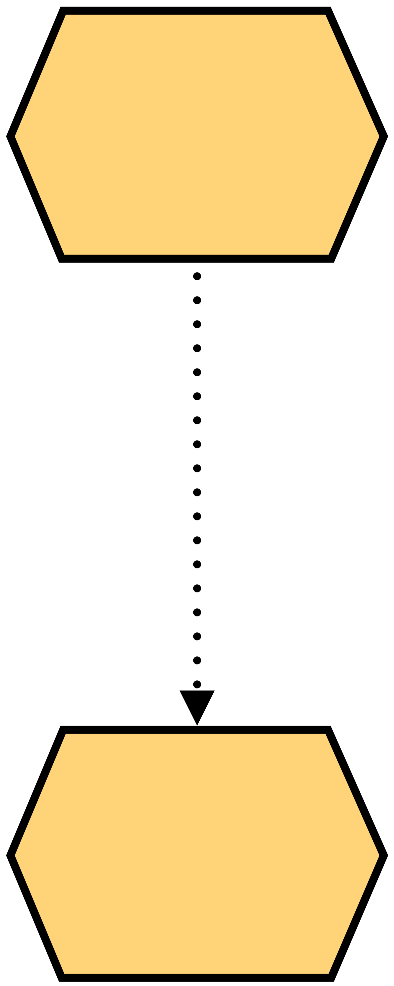


Problem with guidelines

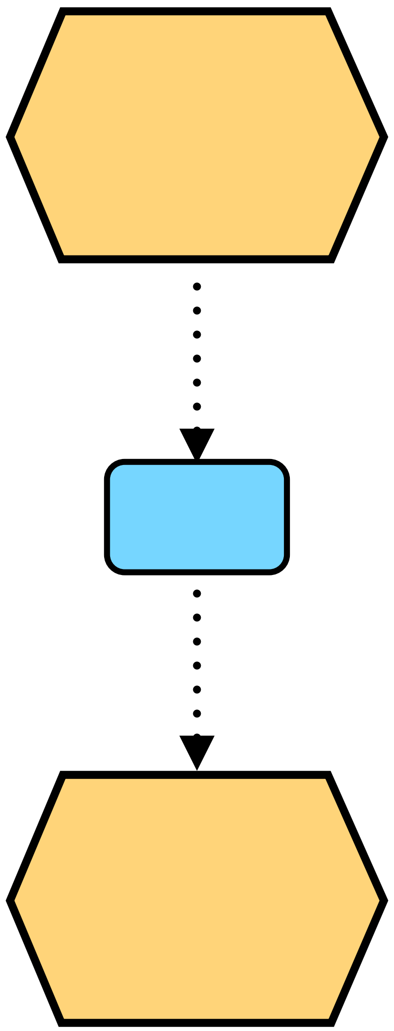
From empirical studies:
guidelines are too restrictive and people ignore them
(following rigid guidelines can lead to unnecessarily complicated diagrams, more difficult to read and understand)

Solution:
It is safe to drop most constraints
(e.g., implicit dummy nodes might always be added later)

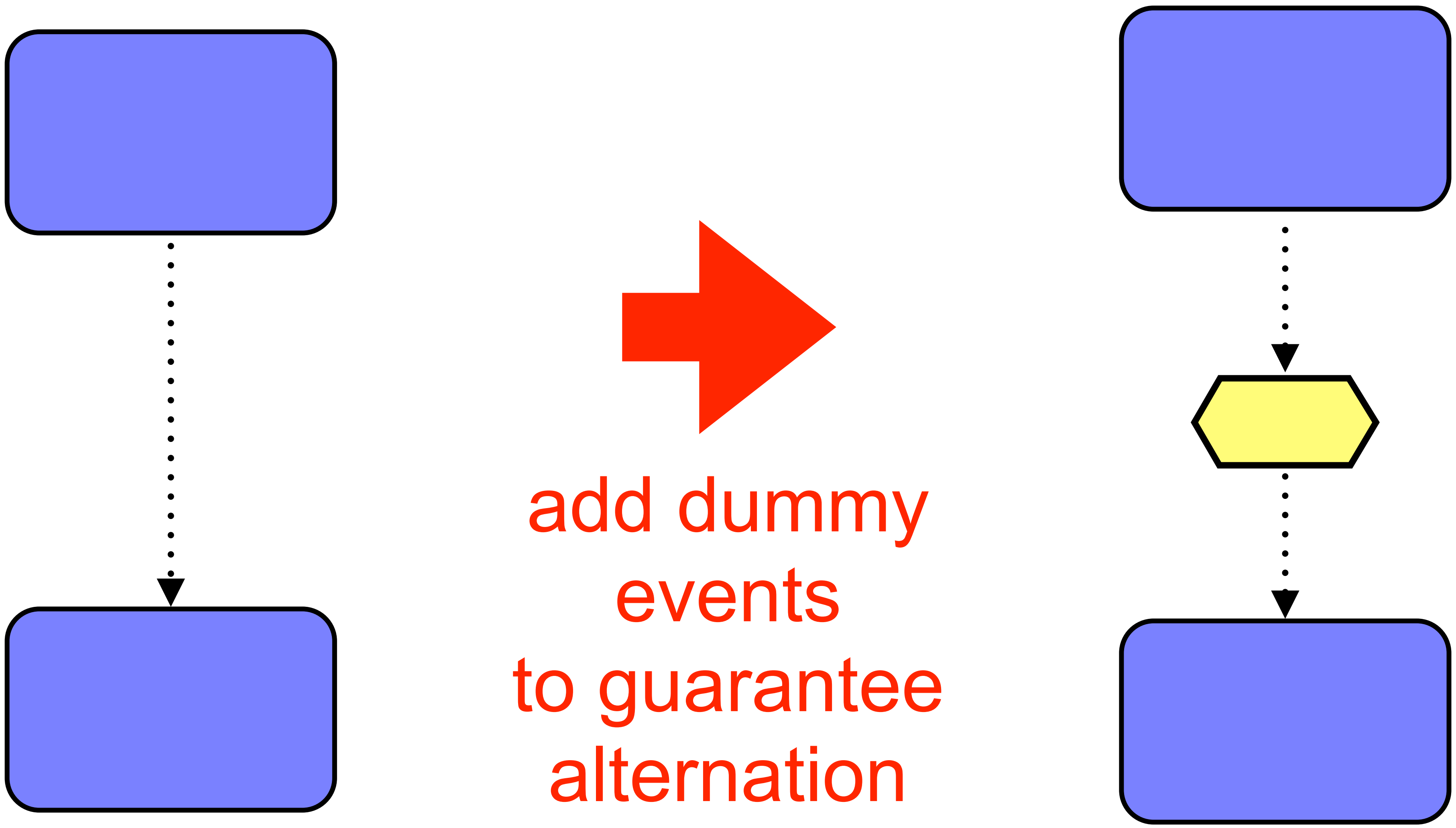
EPC: repairing alternation



add dummy functions to guarantee alternation

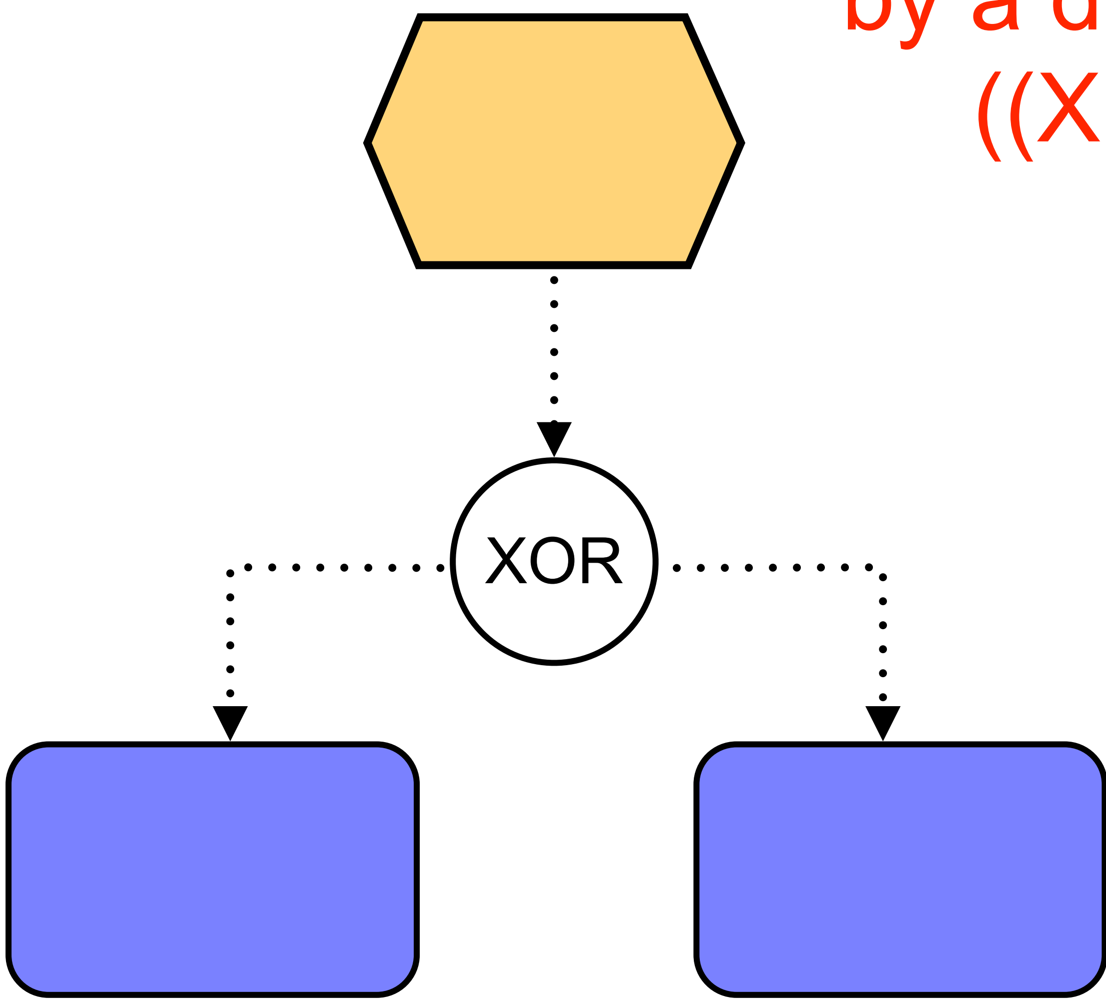


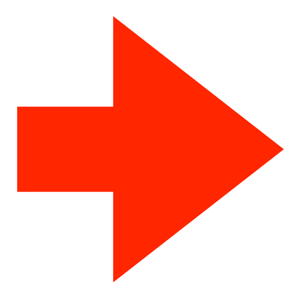
EPC: repairing alternation

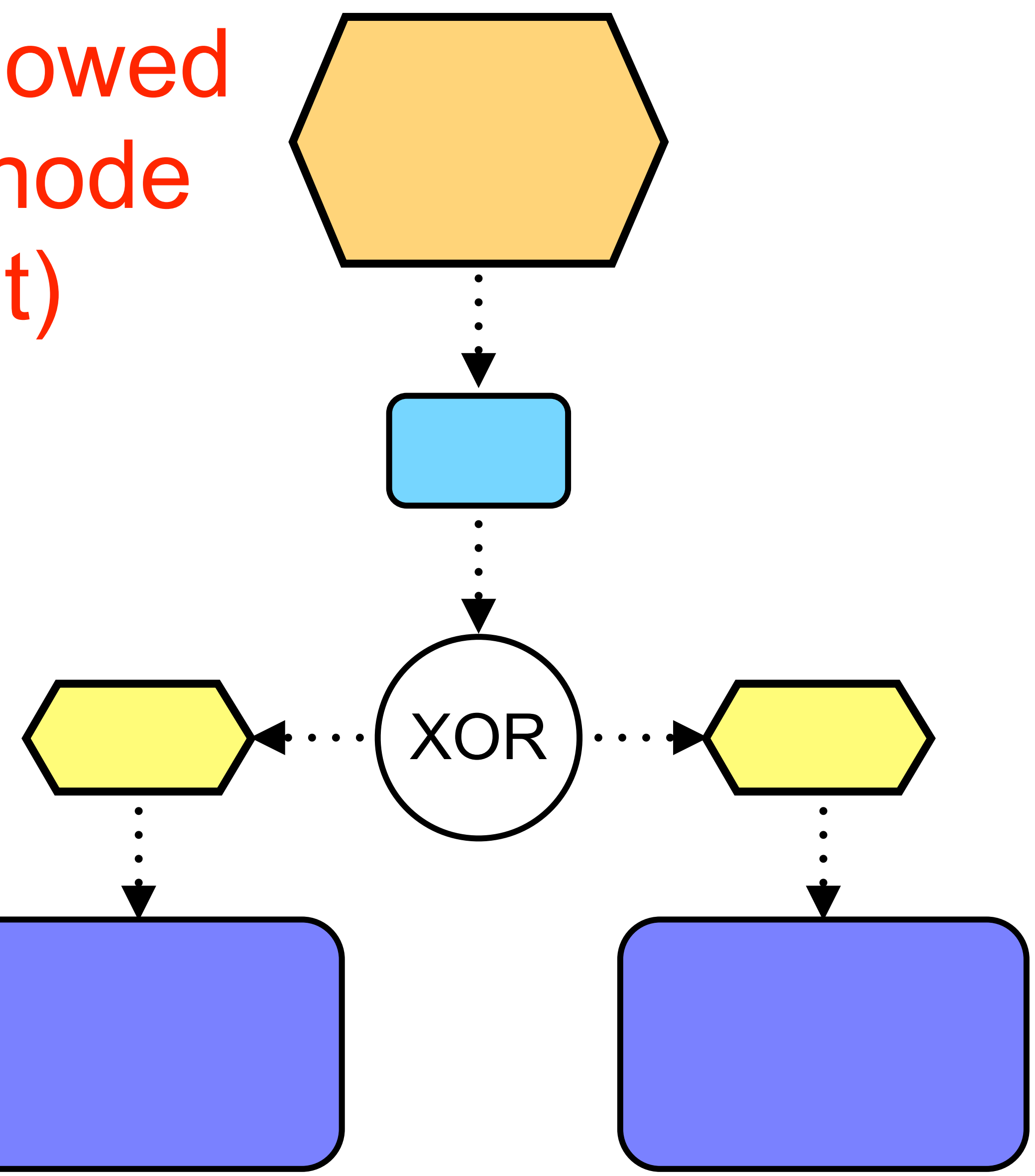


EPC: repairing decisions

add dummy nodes
to guarantee
no event be followed
by a decision node
((X)OR-split)



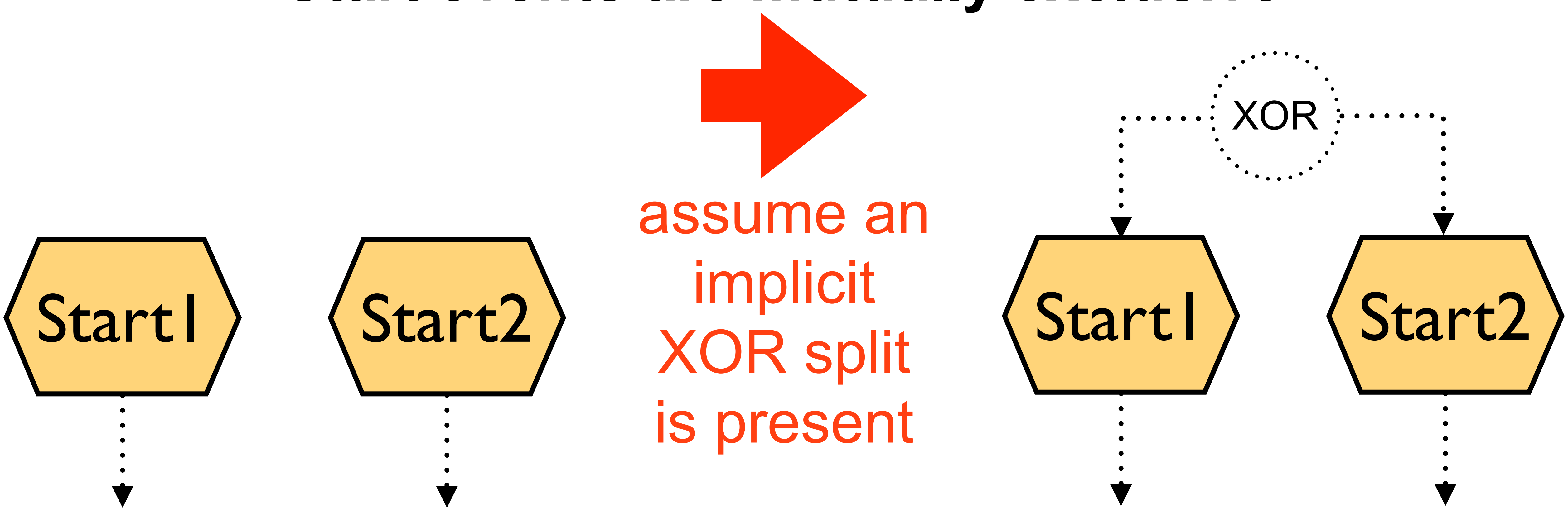




EPC: repairing multiple start events

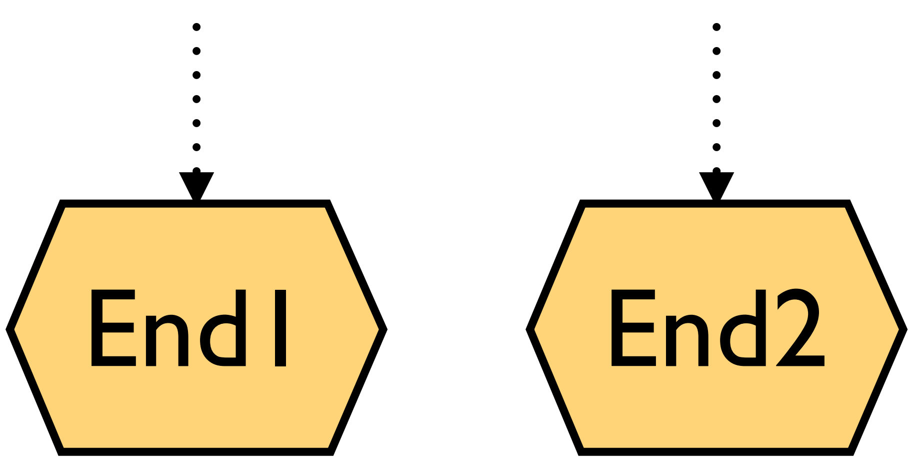
A start event is an event with no incoming arc
it invokes a new instance of the process template

Start events are mutually exclusive

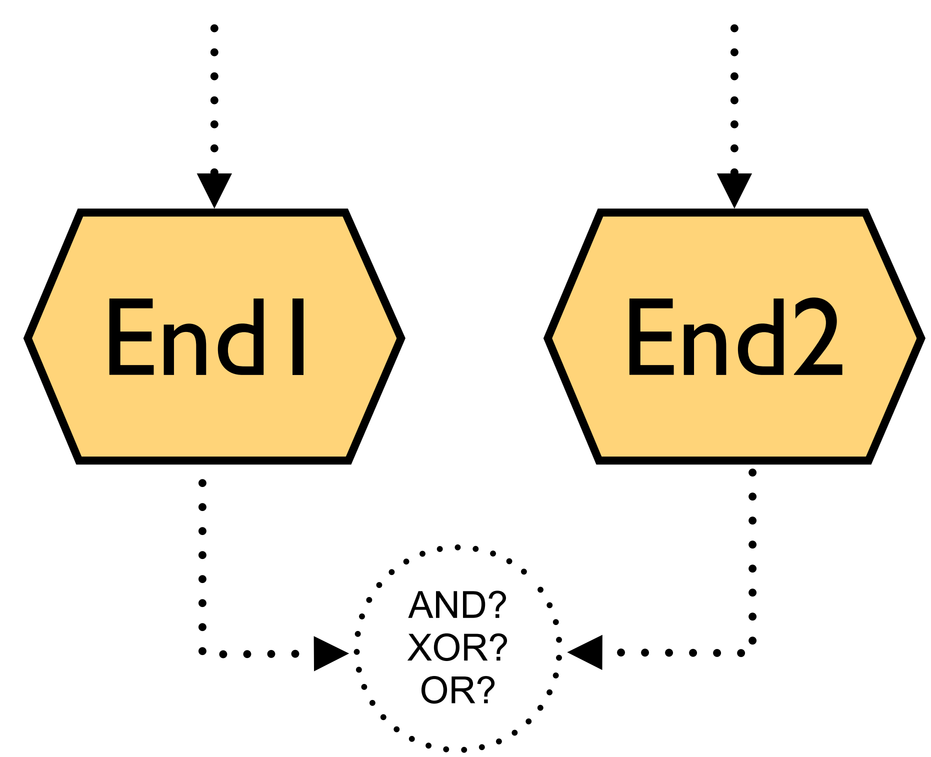


EPC: repairing multiple end events

An end event is an event with no outgoing arc
it indicates completion of some activities
What if multiple end events occur? No unanimity!
they are followed by an implicit join connector
(typically a XOR… but not necessarily so)

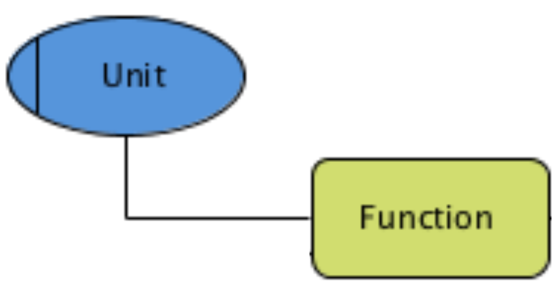


assume an implicit join is present

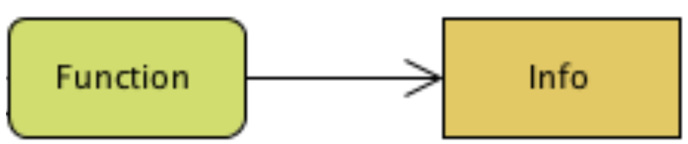


Other ingredients: function annotations

Organization unit:
determines the person or organization responsible for a specific function (ellipses with a vertical line)

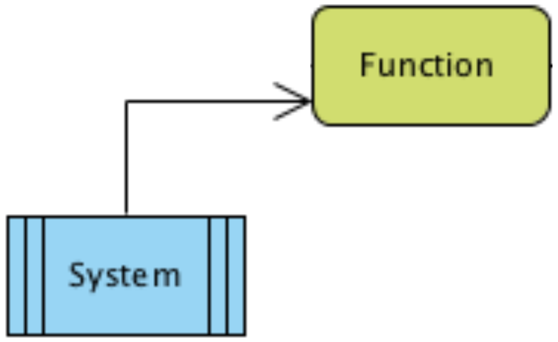


Other ingredients: function annotations



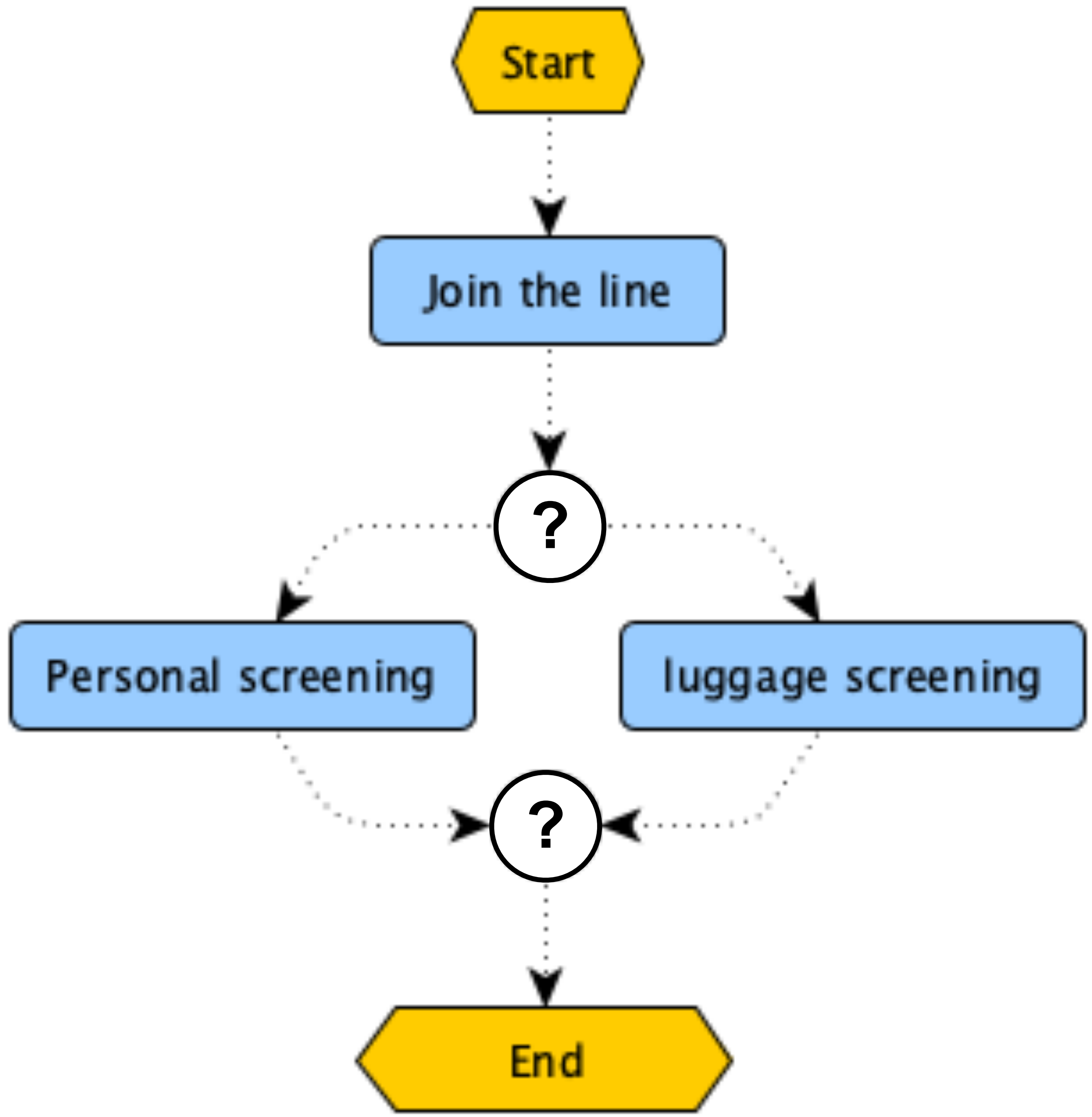
Information, material, resource object:
represents objects in the real world
e.g. input data or output data for a function
(rectangles linked to function boxes)
angles with vertical lines on its sides)

Other ingredients: function annotations

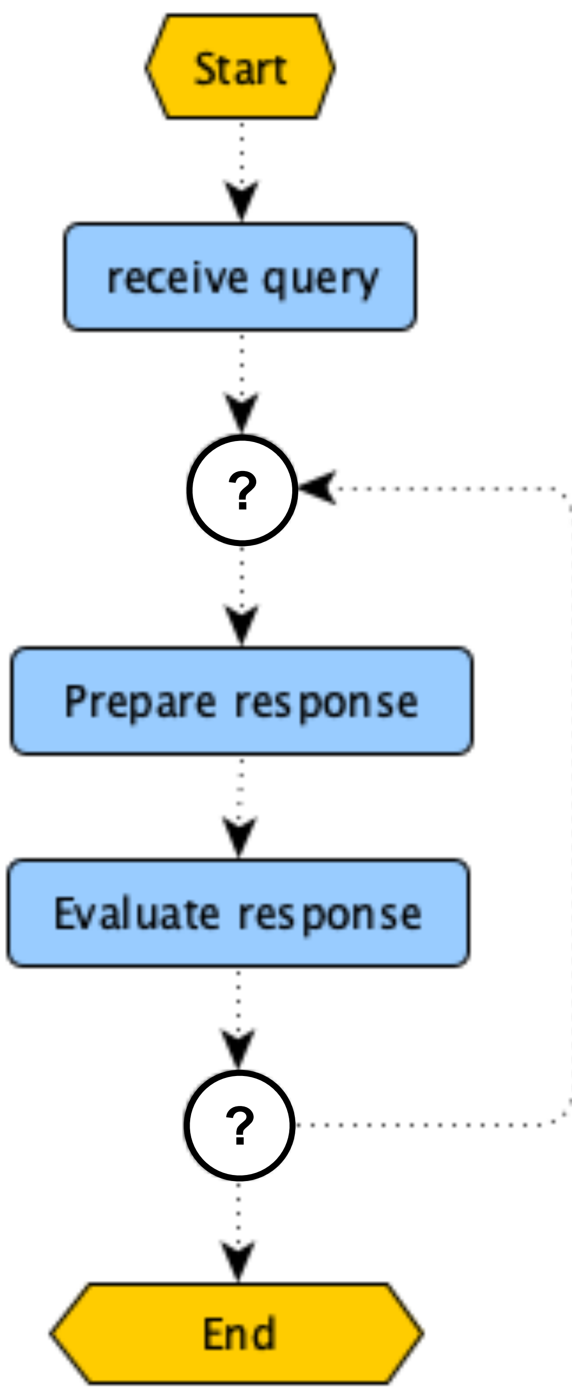


Supporting system: technical support (rectangles with vertical lines on its sides)

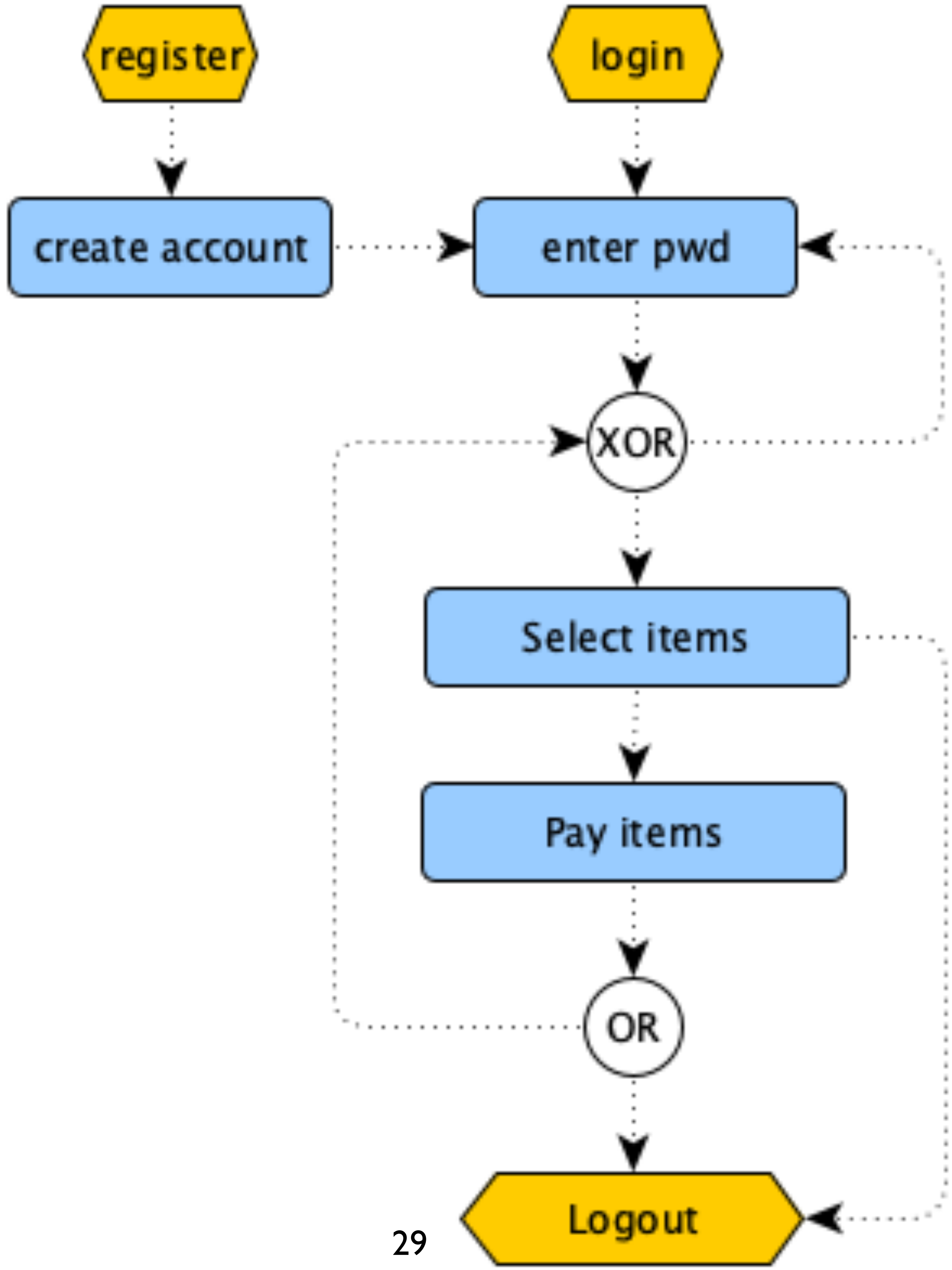
Question time: which connectors?



Question time: which connectors?



Question time: what's wrong?



EPC Sample Diagrams

Example: comments?

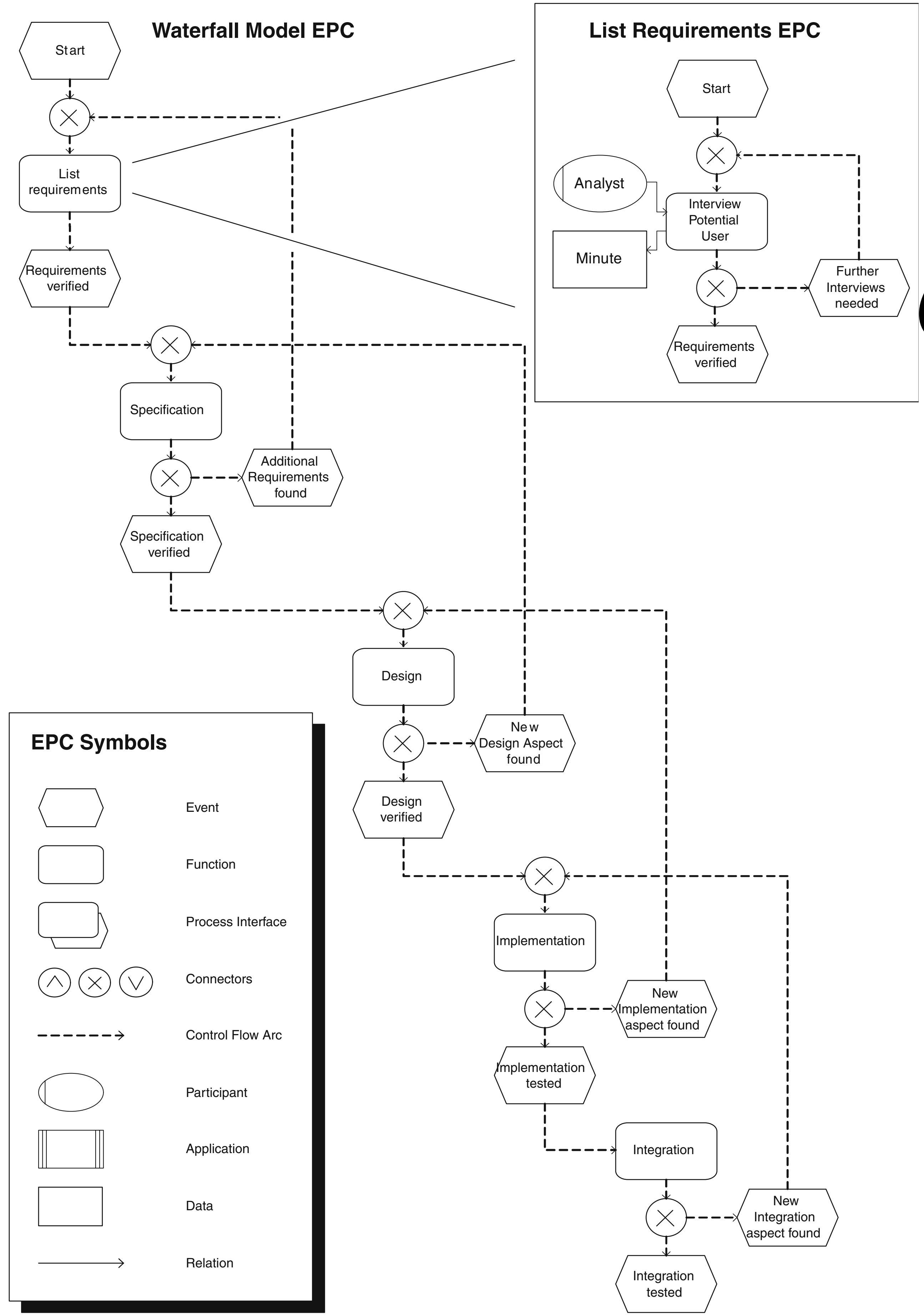
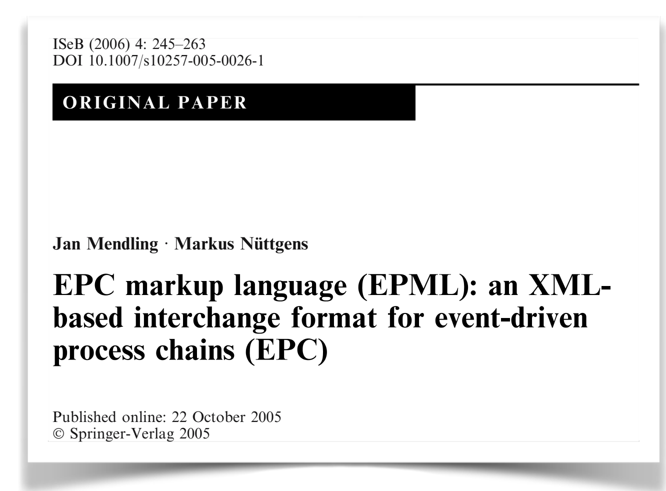


Fig. 1 Event-driven process chains representing the waterfall model for software engineering



ISeB (2006) 4: 245–263
DOI 10.1007/s10257-005-0026-1

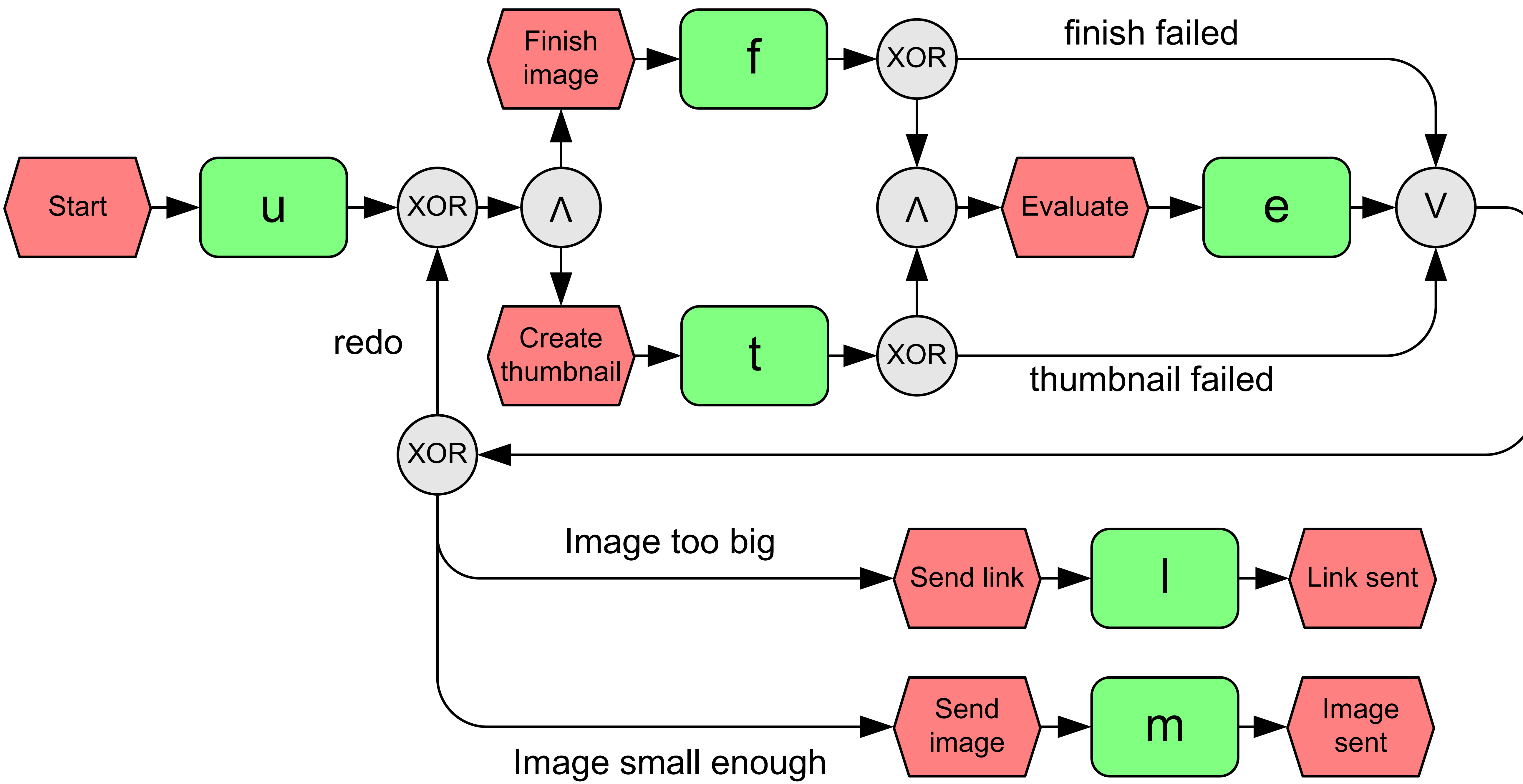
ORIGINAL PAPER

Jan Mendling · Markus Nüttgens

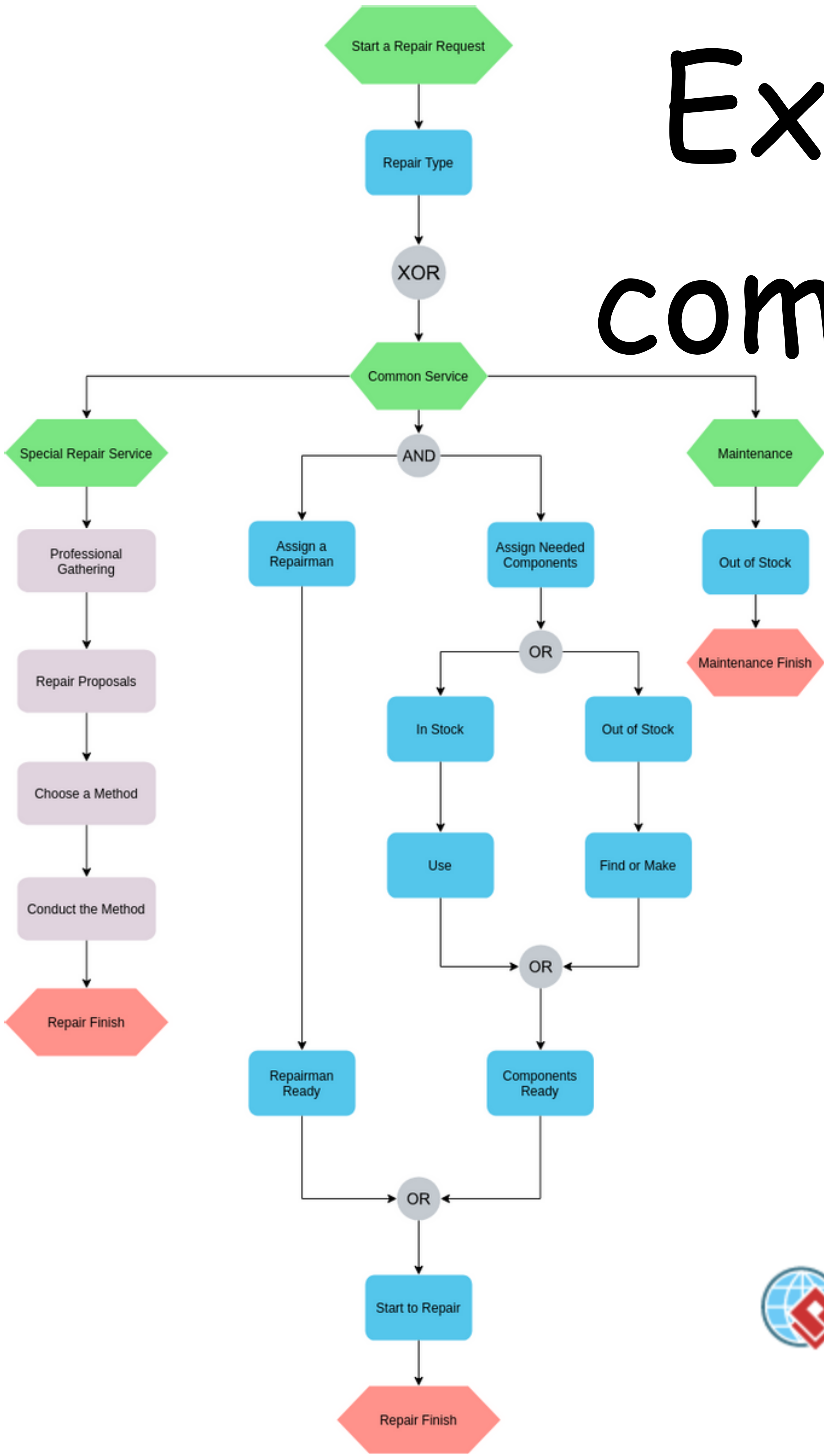
EPC markup language (EPML): an XML-based interchange format for event-driven process chains (EPC)

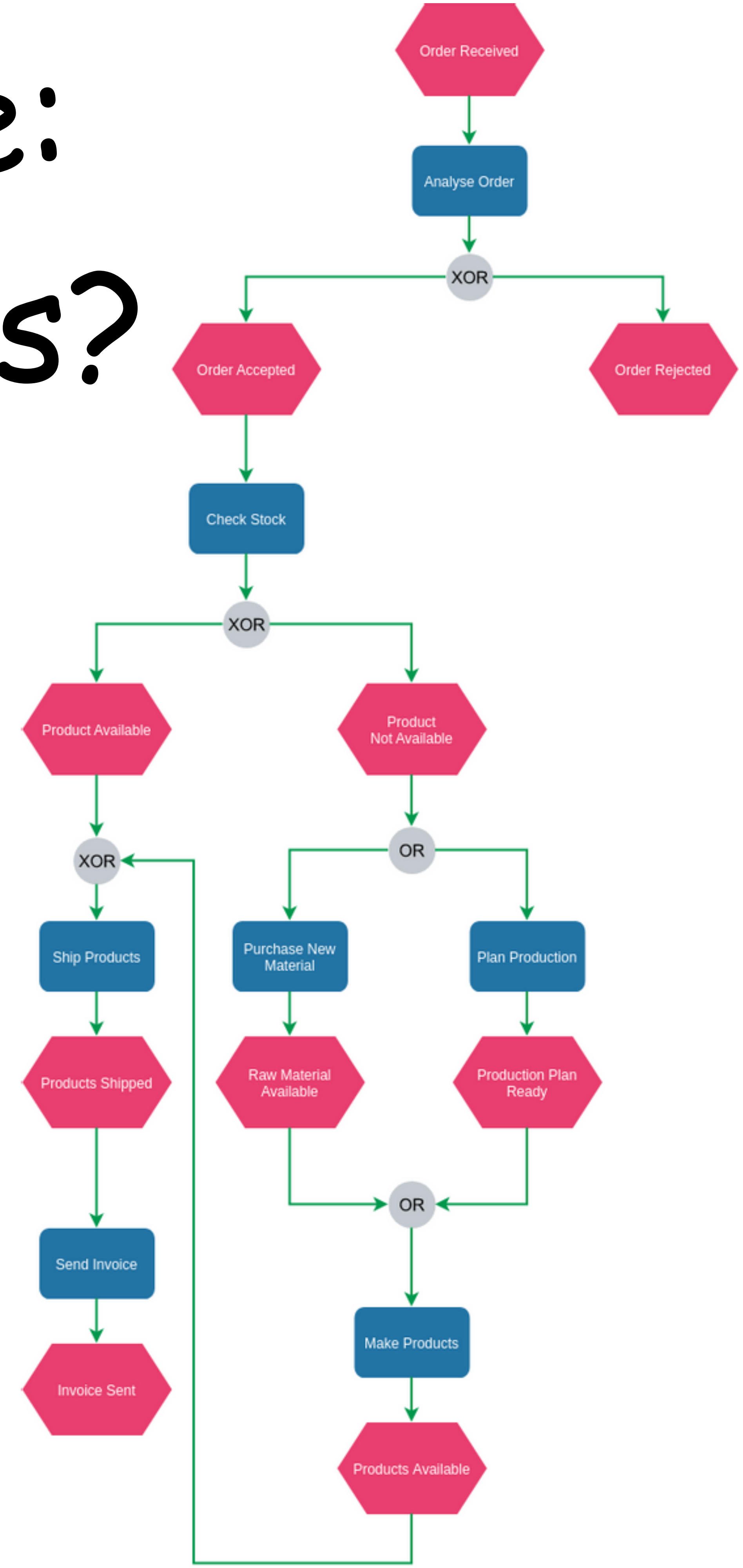
Published online: 22 October 2005

Example: any comment?



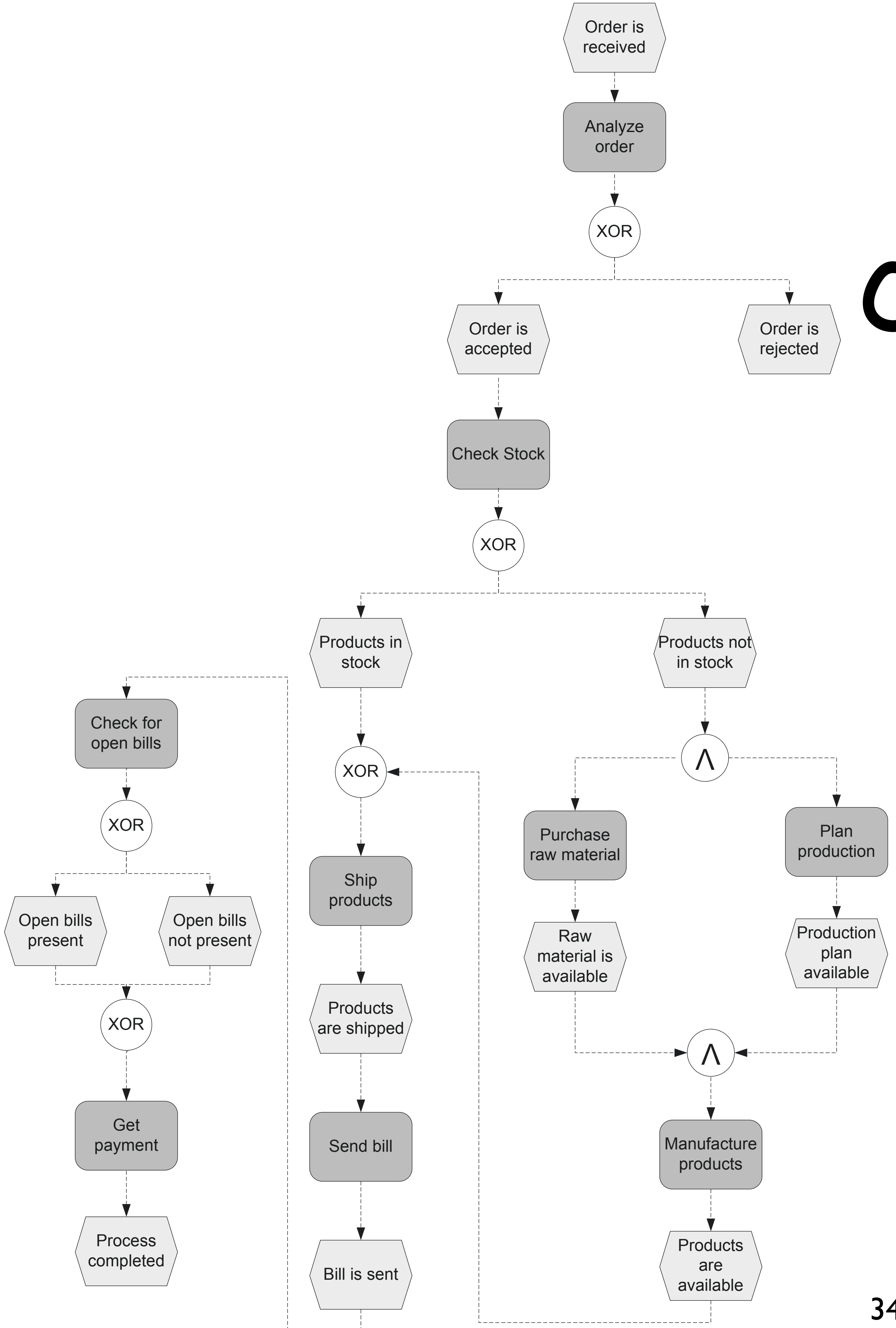
Example: comments?

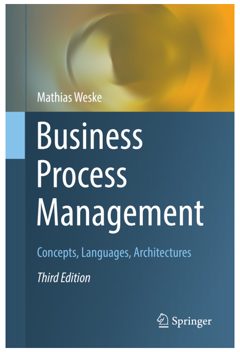




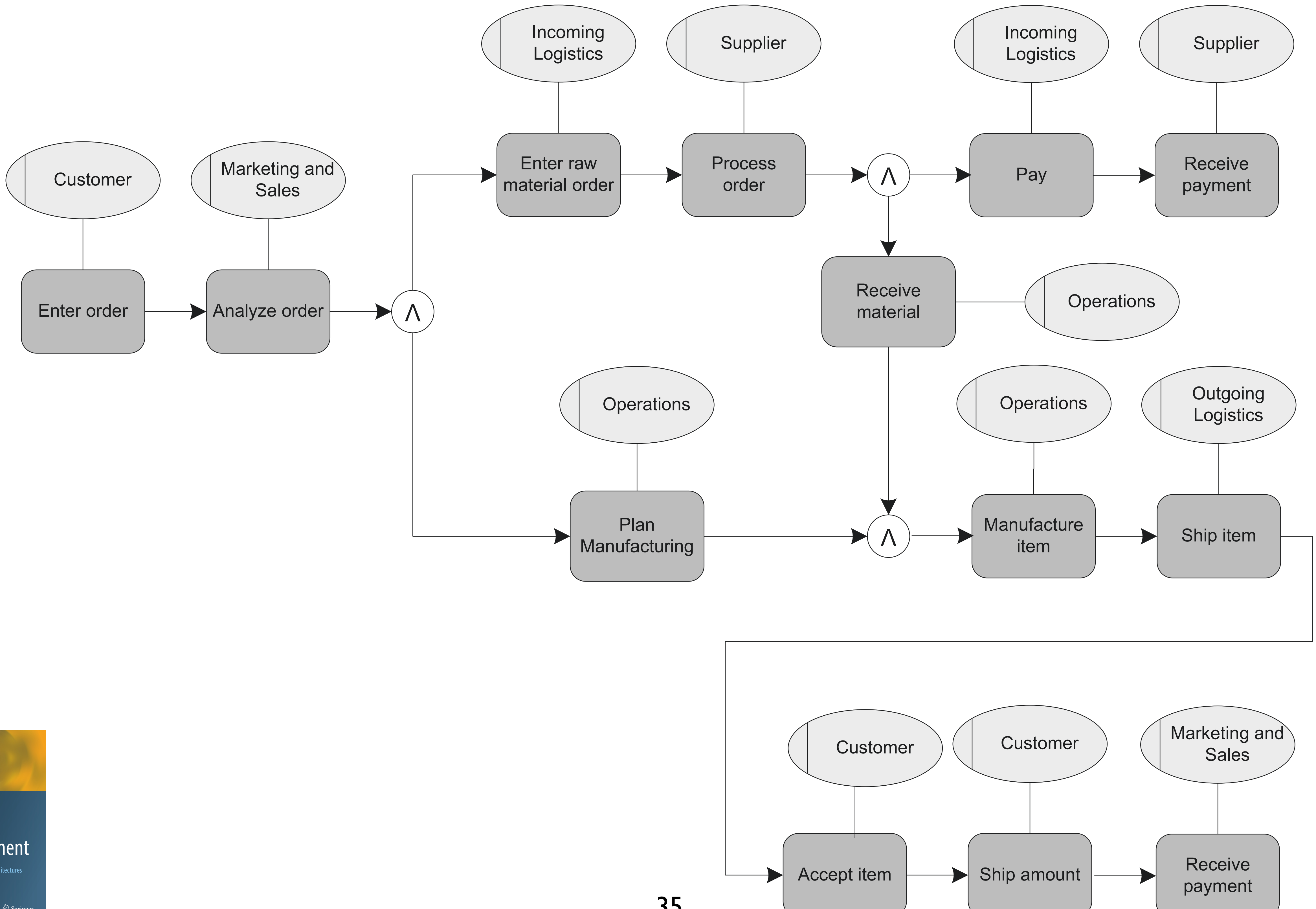
Visual Paradigm Online

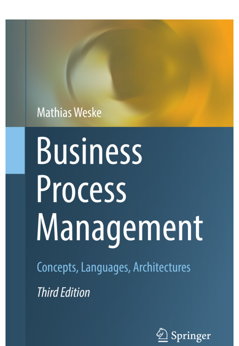
Example: comments?





Example: any comment?





EPC Semantics

EPC intuitive semantics

A process starts when some initial event(s) occurs

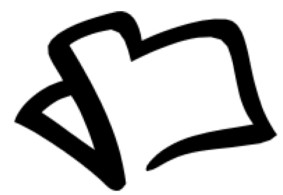
The activities are executed according to the constraints in the diagram

When the process is finished,
only final events have not been dealt with

If this is always the case, then the EPC is “correct”

Folder-passing semantics

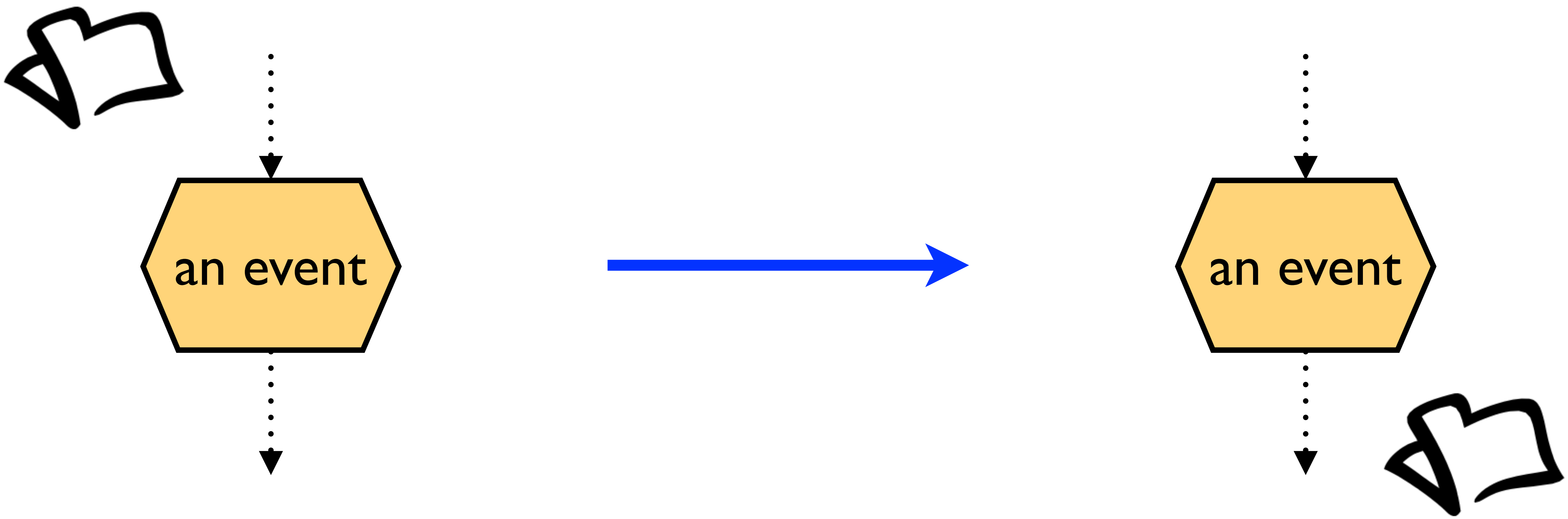
The current state of the process is determined by placing folders over the diagram



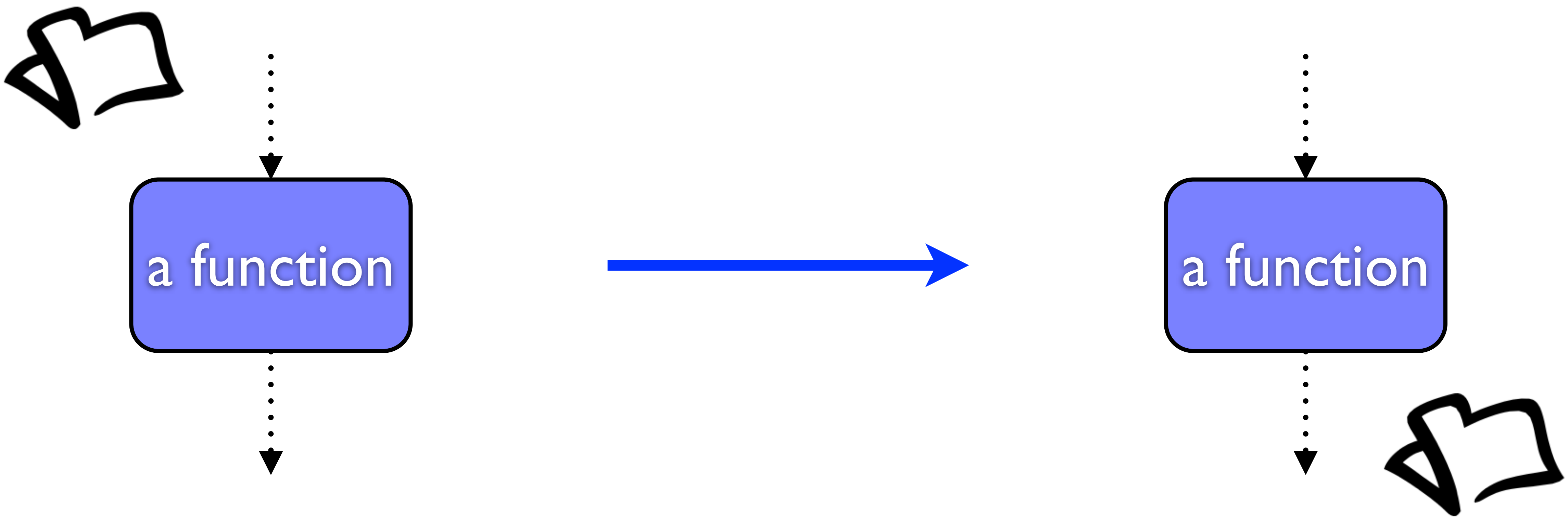
A transition relation explains how to move from one state to the next state

The transition relation is possibly nondeterministic

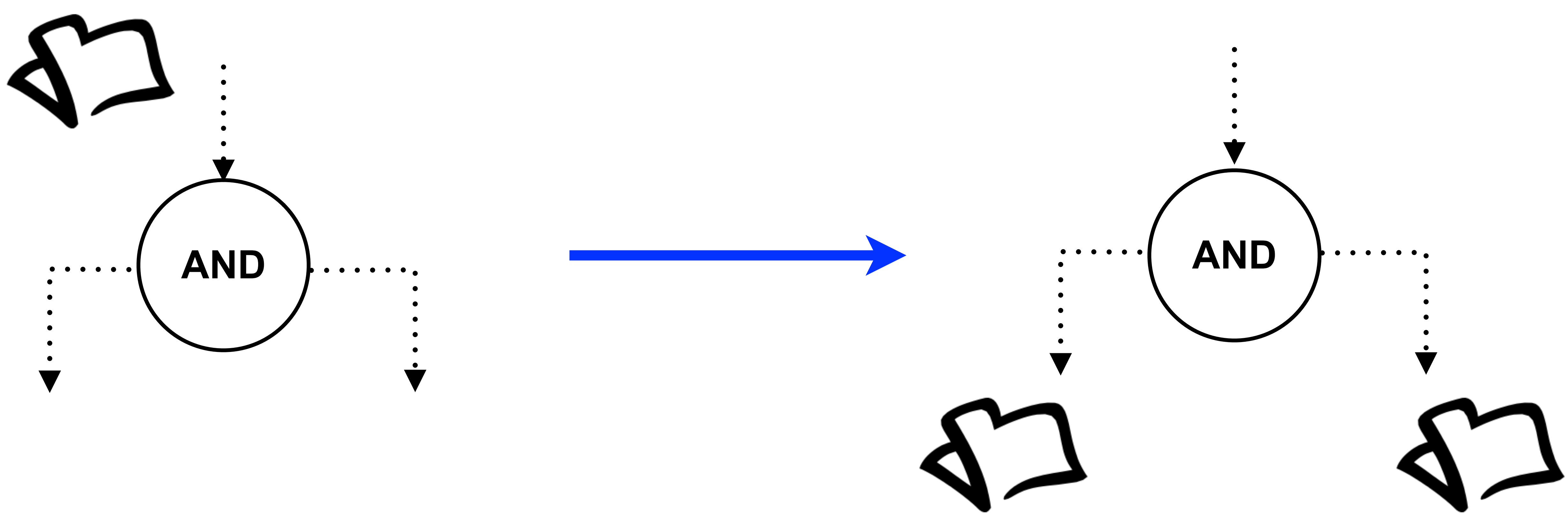
Folder-passing semantics: events



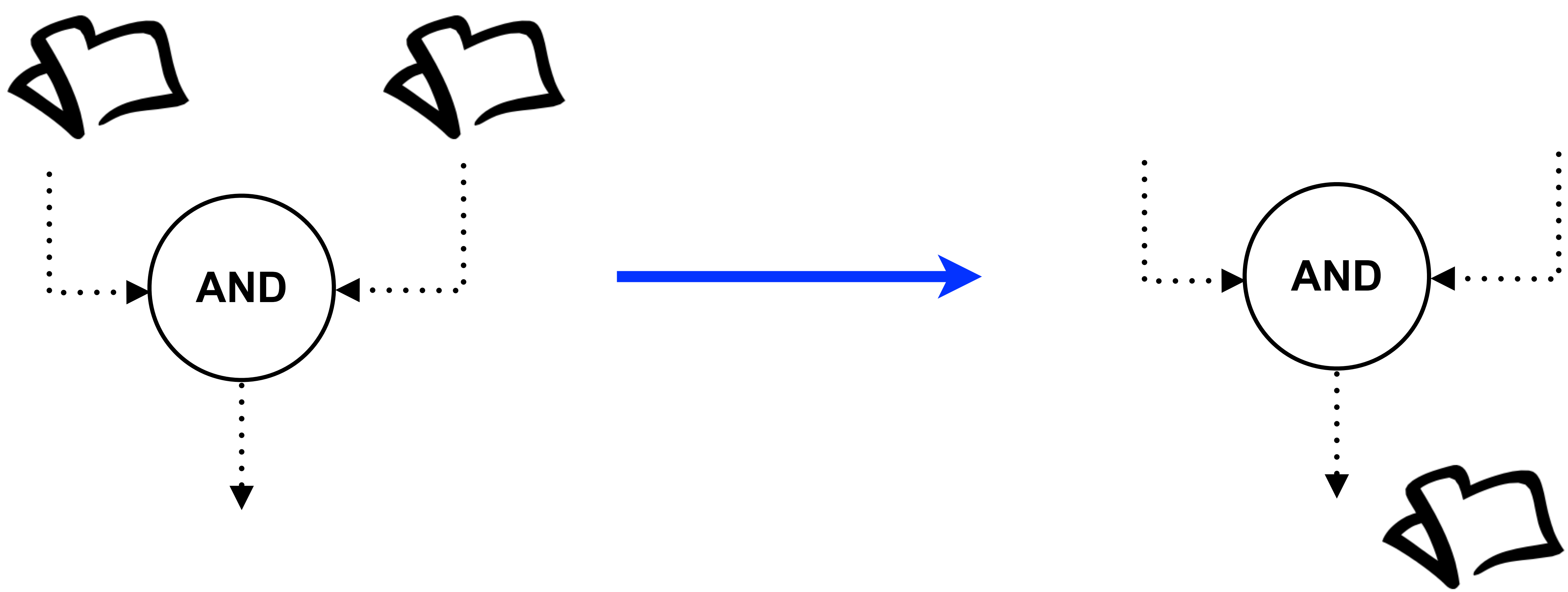
Folder-passing semantics: functions



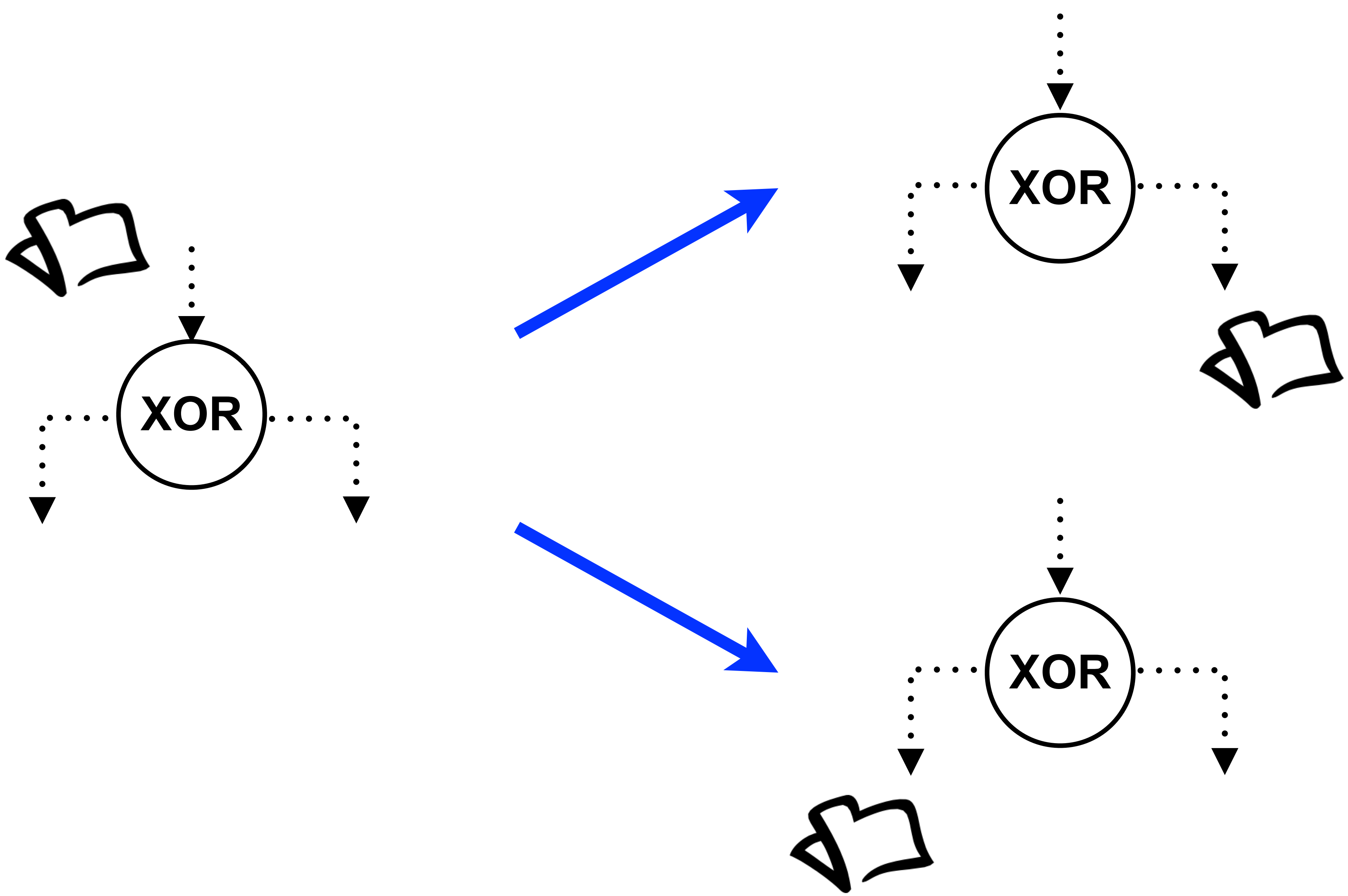
Folder-passing semantics: AND-split



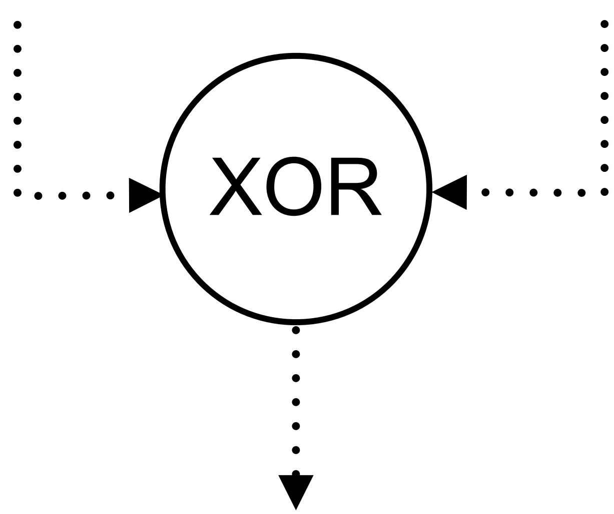
Folder-passing semantics: AND-join



Folder-passing semantics: XOR-split



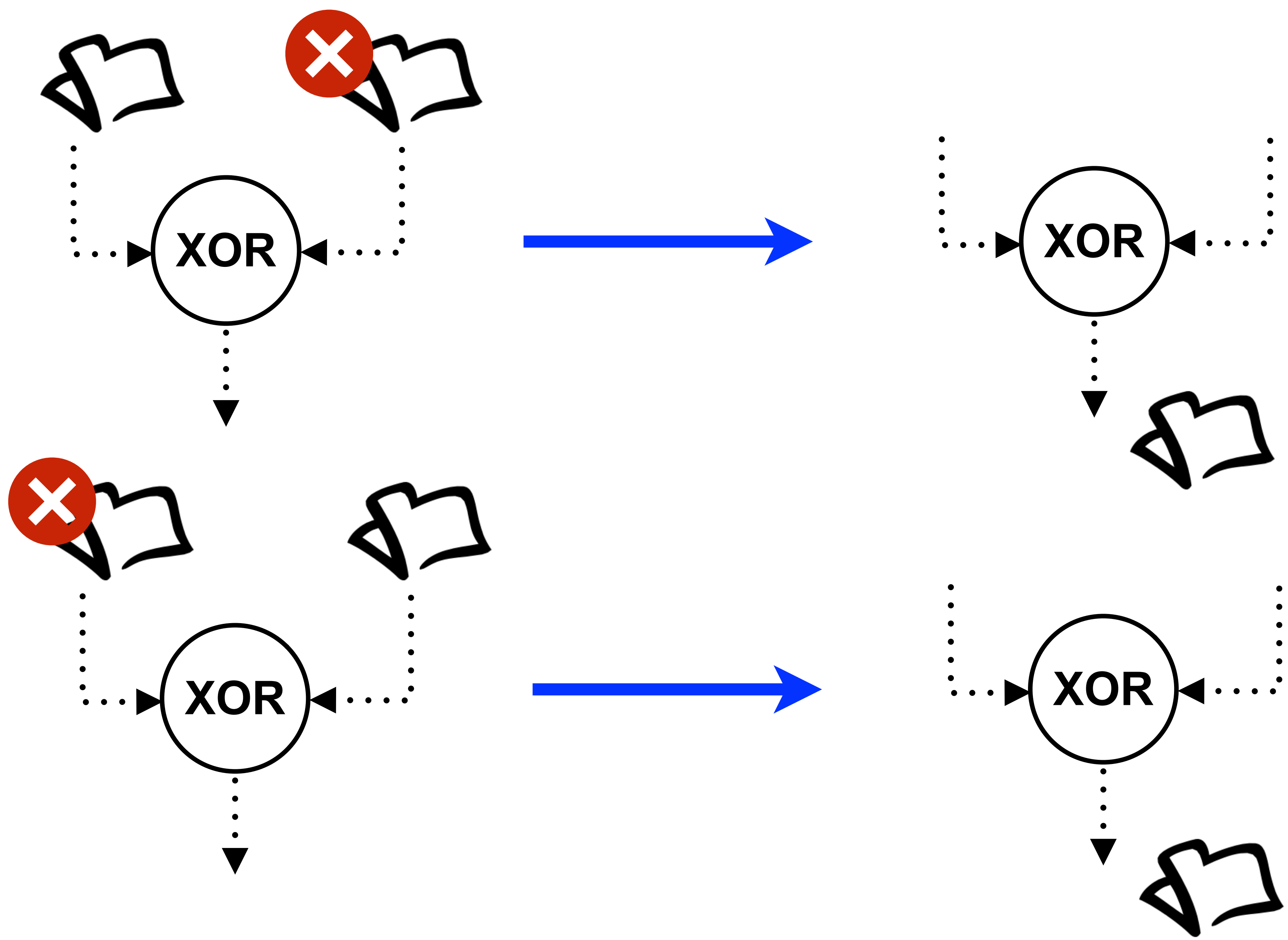
XOR join: intended meaning



if both inputs arrive,
it should block the flow

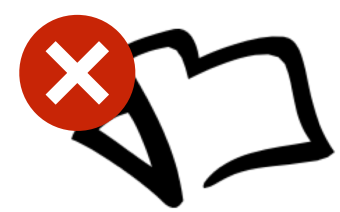
if one input arrives,
it cannot proceed unless
it is informed that
the other input will never arrive

Folder-passing semantics: XOR-join



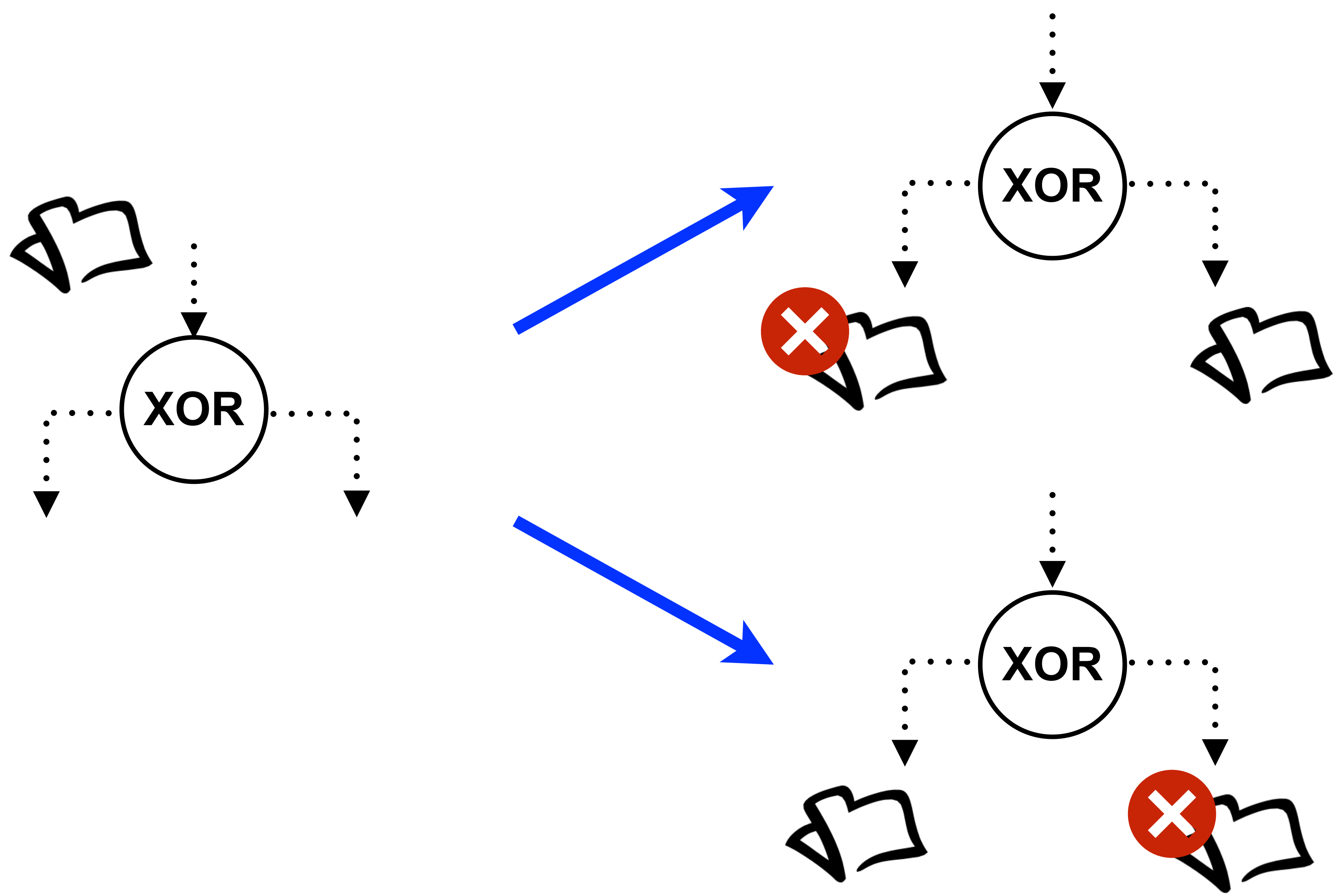
Folder-passing semantics?

How can we infer the absence of folders?

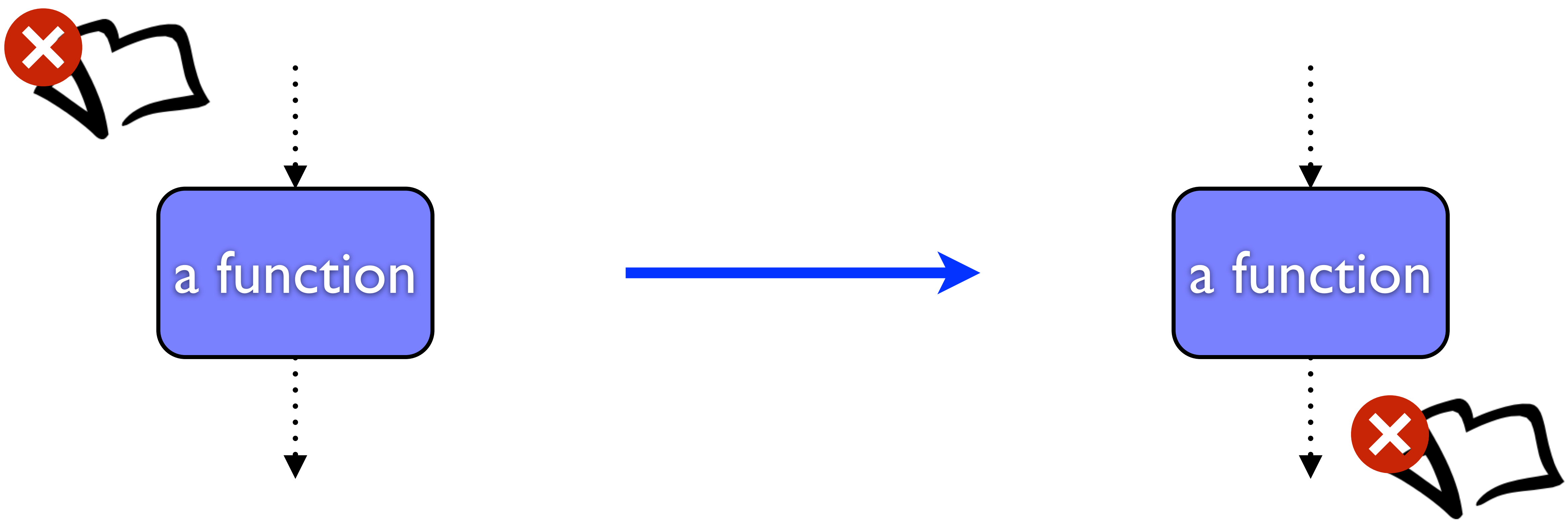


When and how should such information be propagated?

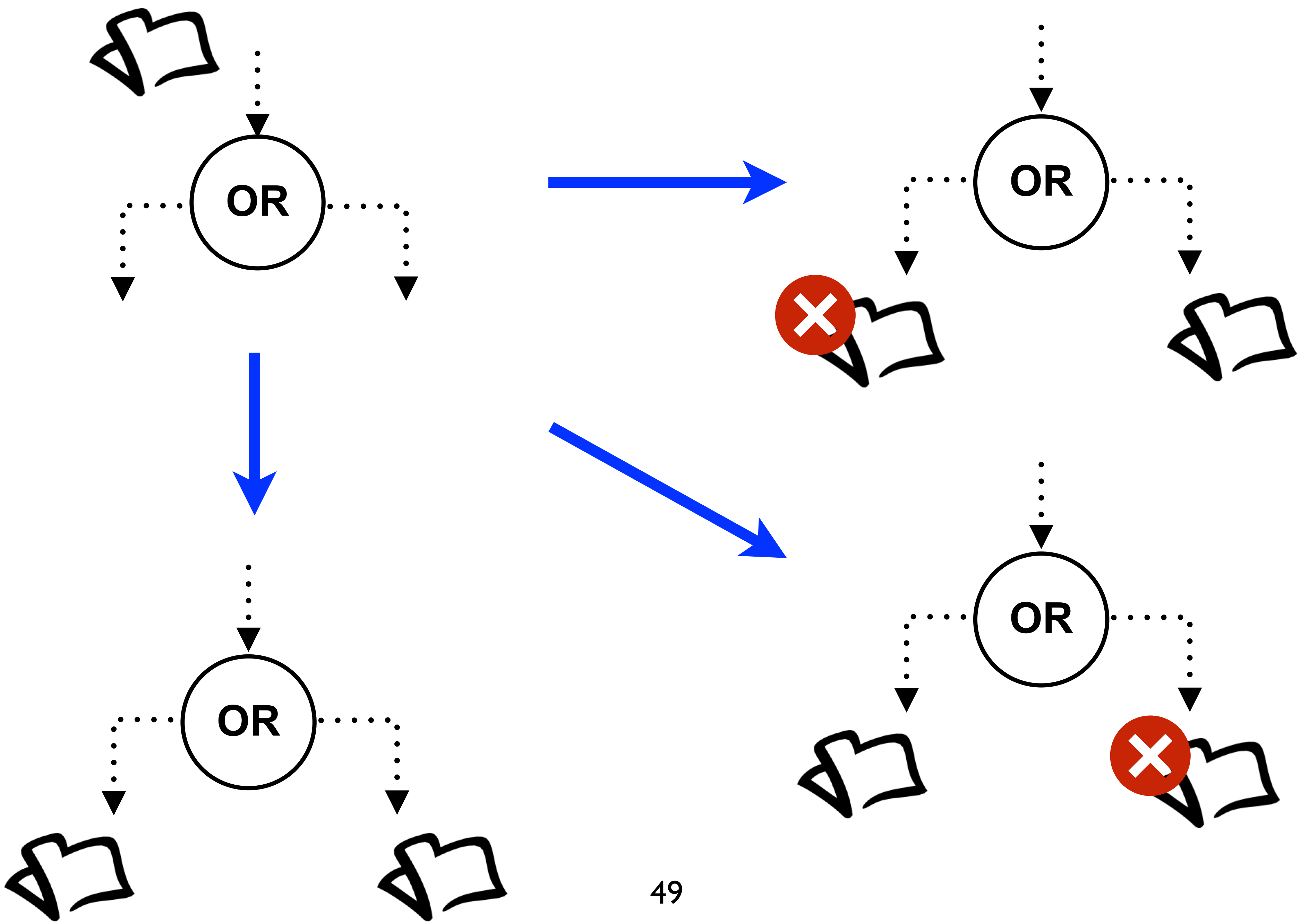
Absence of folders: creation



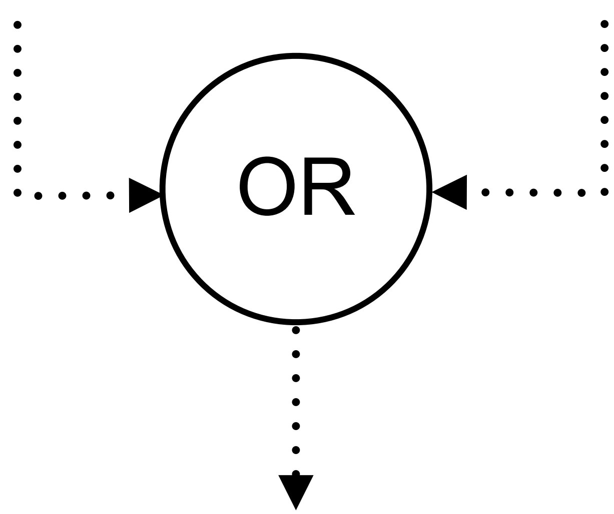
Absence of folders: propagation (example)



Folder-passing semantics: OR-split



OR join: intended meaning

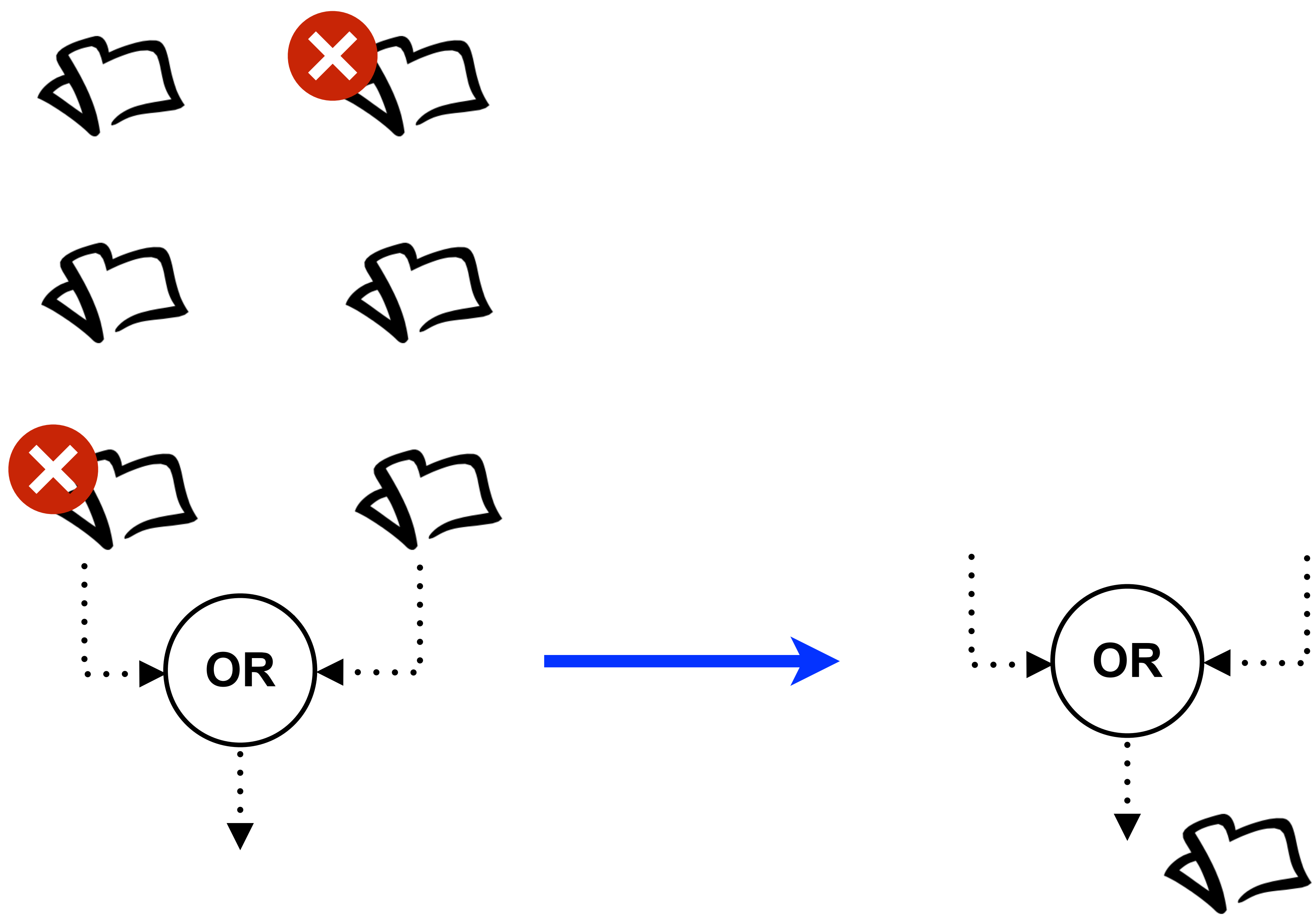


if only one input arrives,
it should release the flow

if both inputs arrive,
it should release only one output

if one input arrives,
it must wait until the other arrives or
it is guaranteed that the other will never arrive

Folder-passing semantics: OR-join?



Decorated EPC

To remove ambiguous behaviour for join connectors, designers can further annotate EPC diagrams

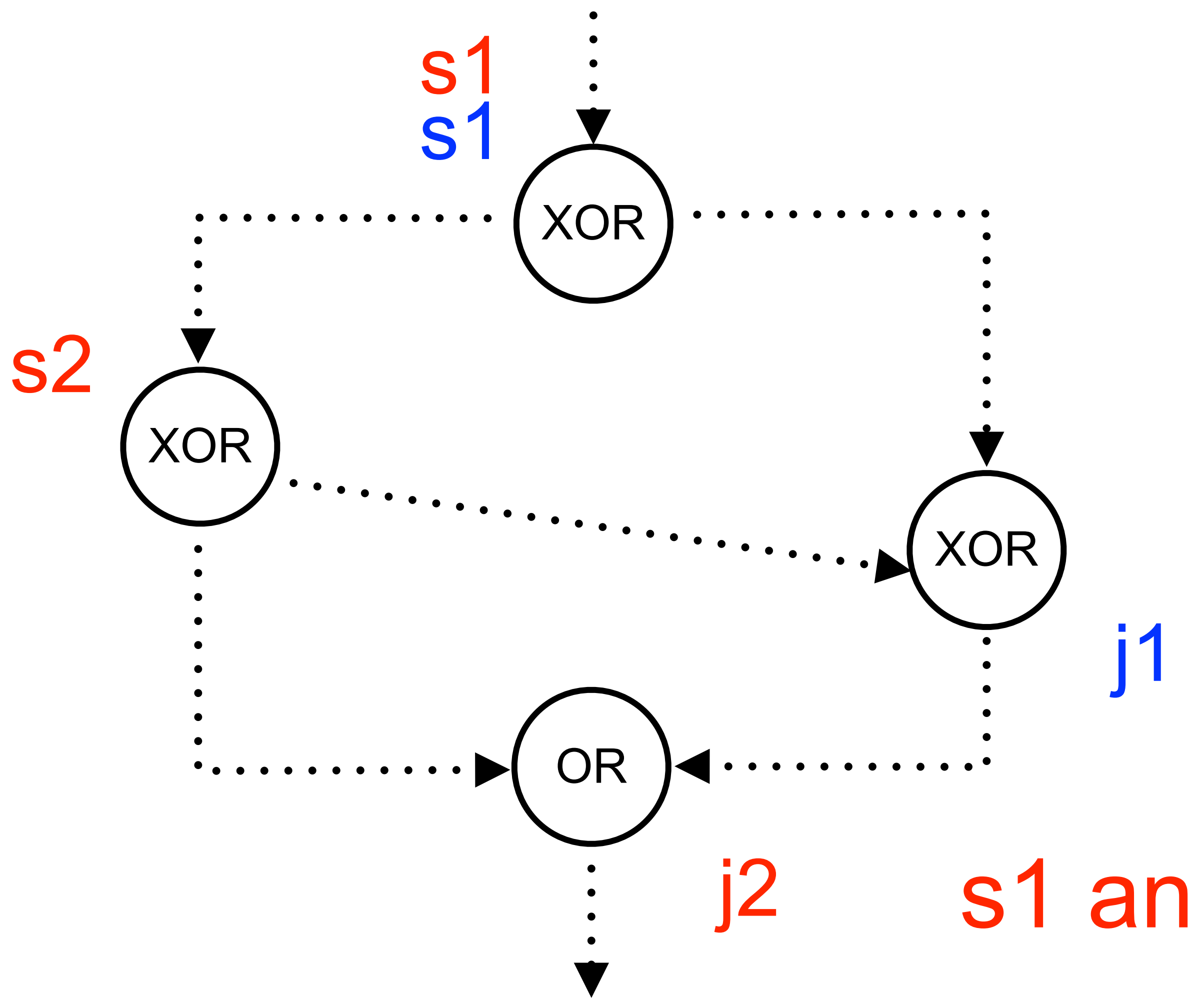
In particular we require to know:

corresponding split
which node separated the flows we are joining
(in the case of XOR/OR join)

applicable policy
how to trigger outgoing flow
(avoid OR join ambiguous behaviour)

Candidate split

A **candidate split** for a join node is any split node whose outputs are connected to the inputs of the join

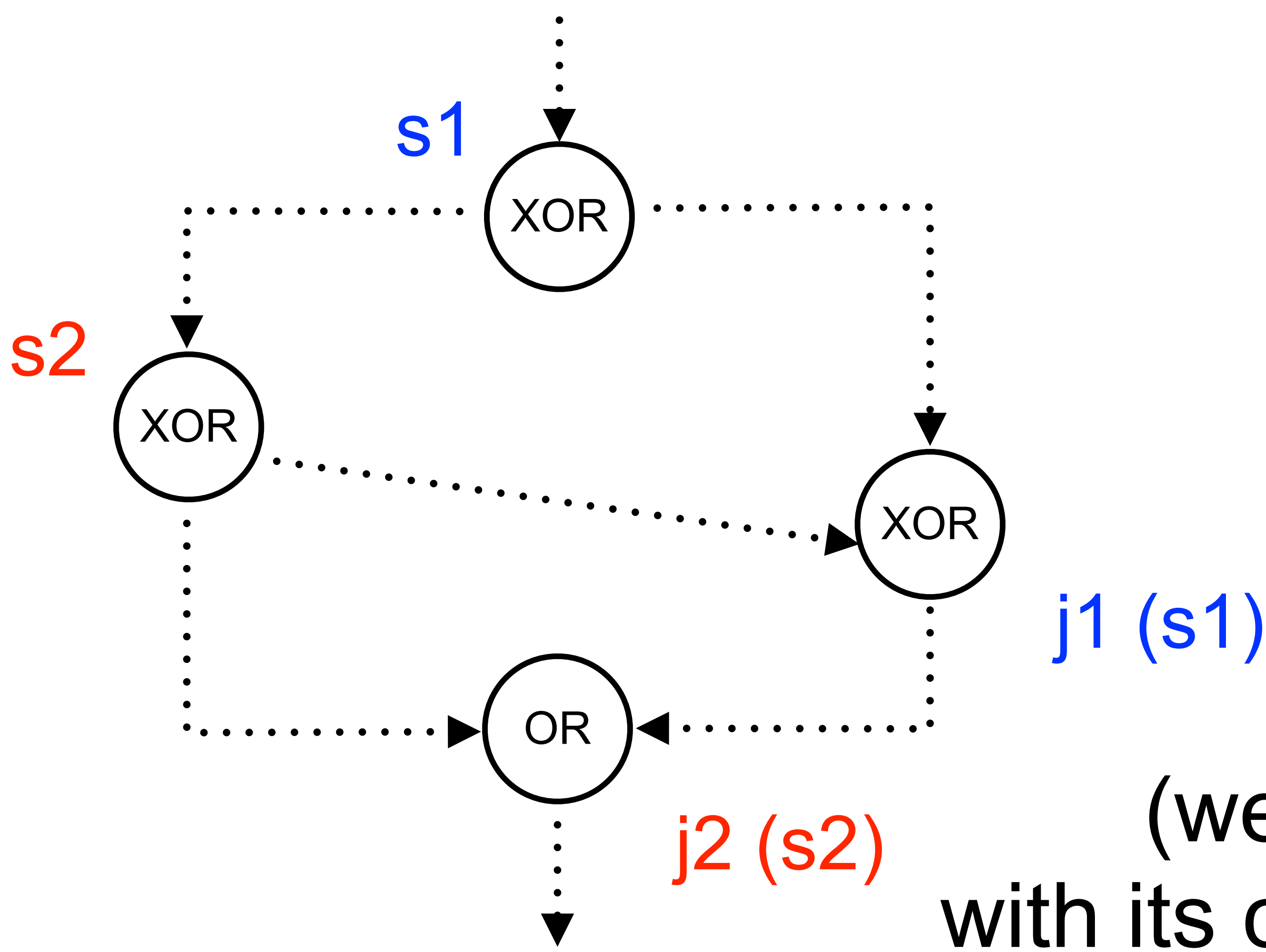


s1 is a candidate split for j1

s1 and s2 are candidate splits for j2

Corresponding split

A **corresponding split** for a join node
is a chosen candidate split



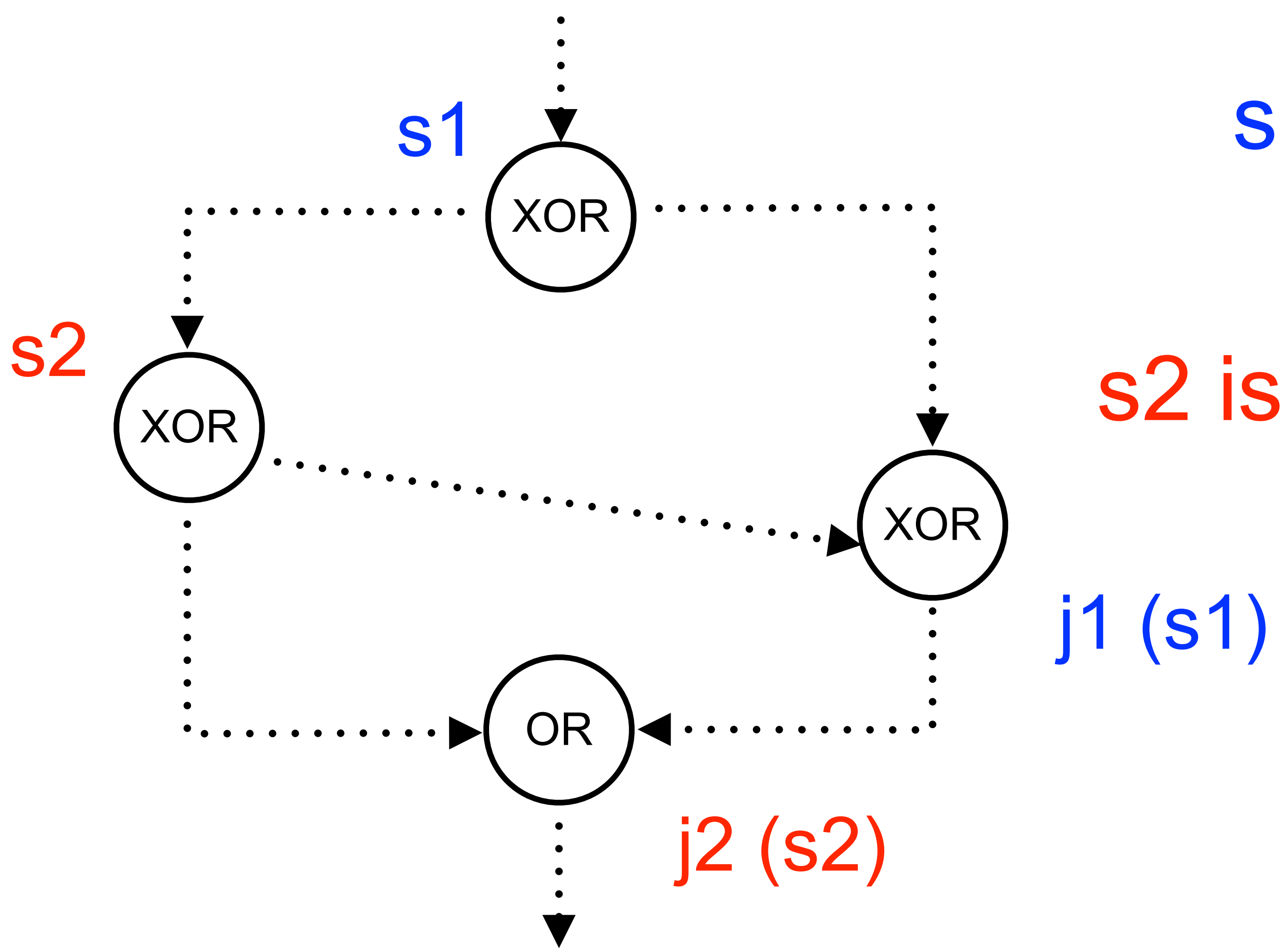
we choose s1 as a
corresponding split for j1

we choose s2 as a
corresponding split for j2

(we tag each join
with its corresponding split)

Matching split

A corresponding split for a join node is called **matching** if it has the same type as the join node



s1 is a matching split for j1

s2 is not a matching split for j2

OR join: policies

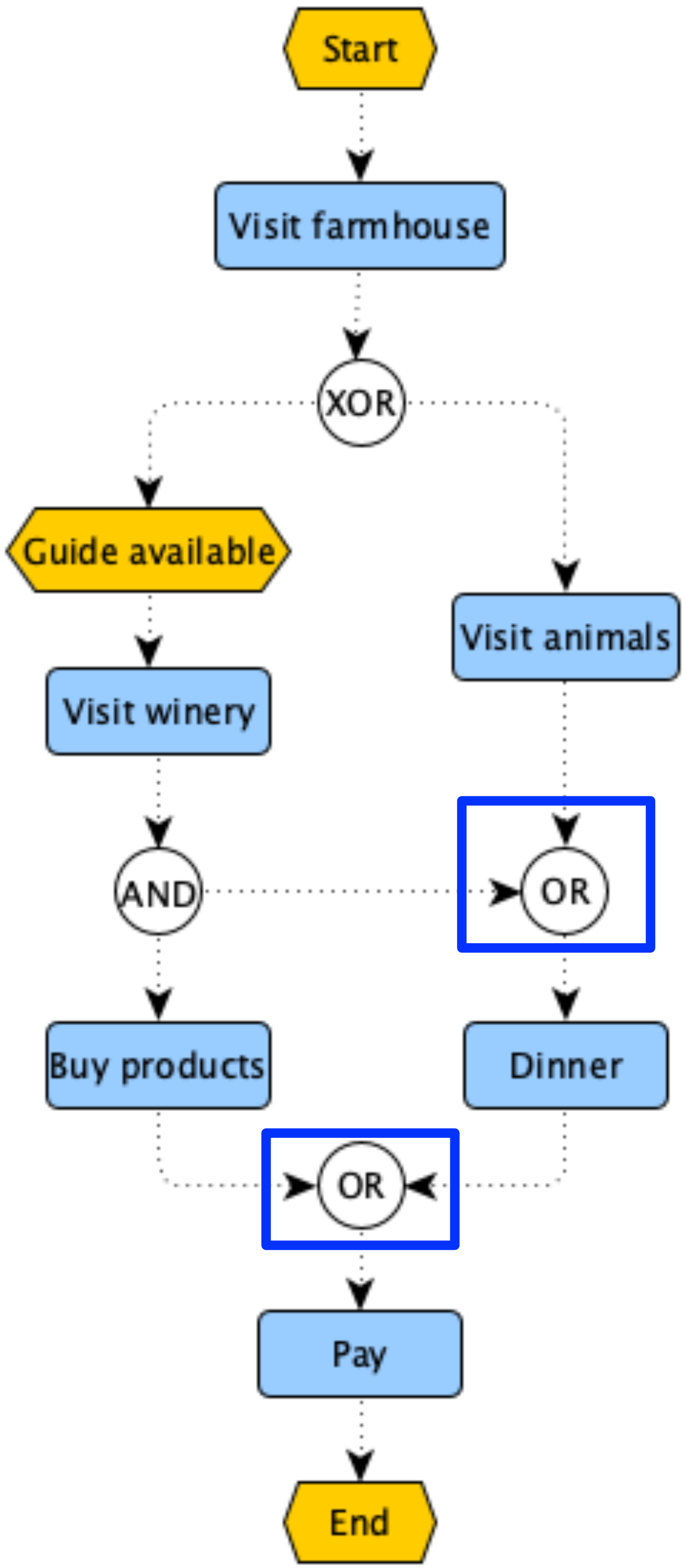
If an OR join has a **matching split**, its semantics is **wait-for-all**: wait for the completion of all *activated* paths

Otherwise, also other policies can be chosen:

every-time: trigger the outgoing path on each input

first-come: wait for the first input and ignore the second

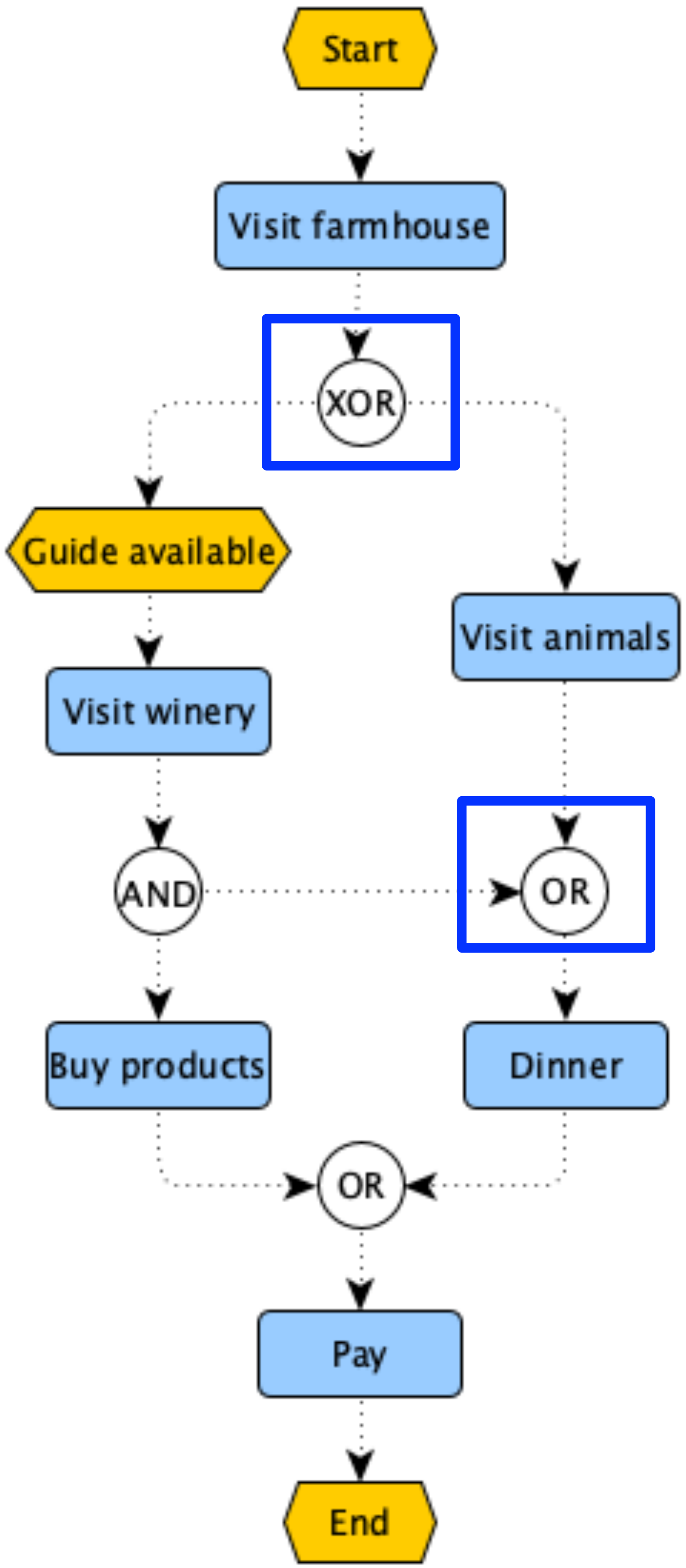
Assumption: every OR join is tagged with a policy (some suggested to have different trapezoid symbols)



Example

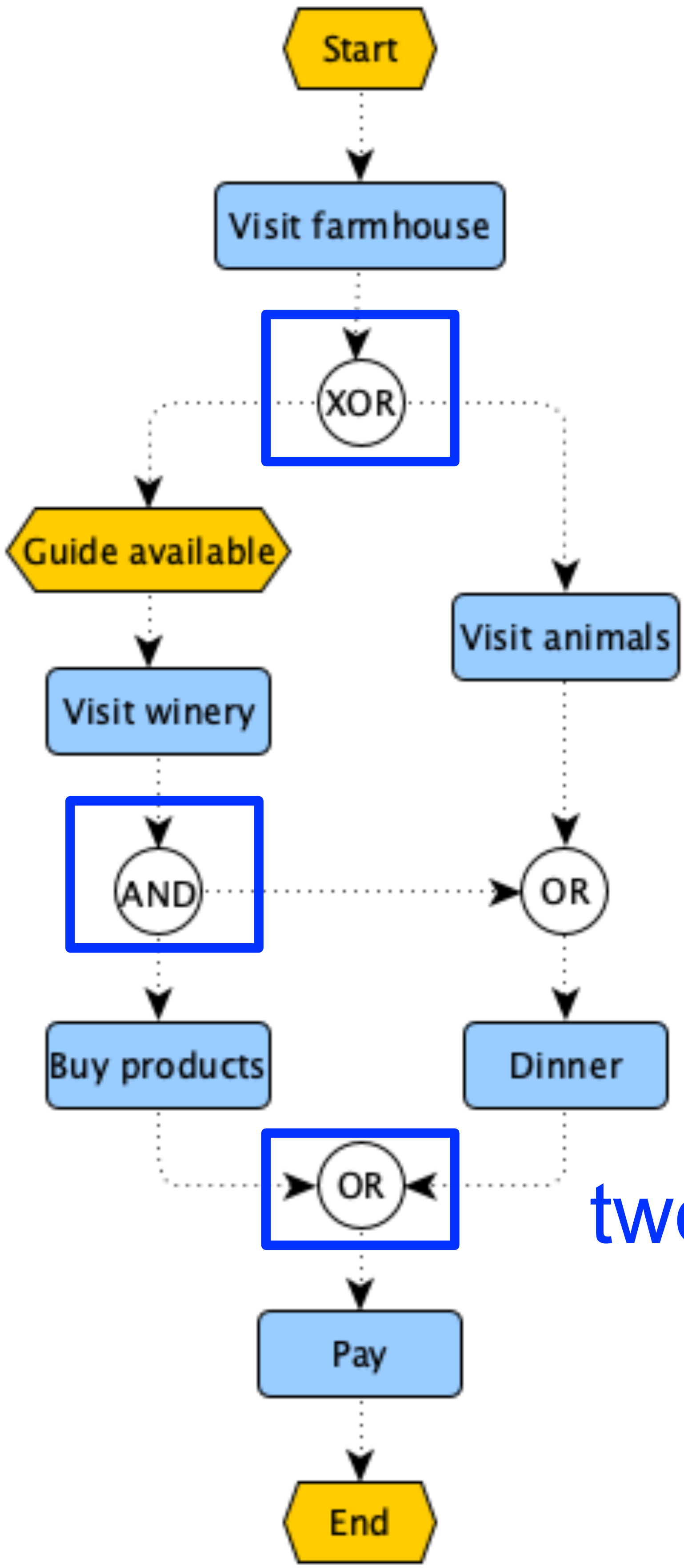
two OR joins
but no OR split

Example

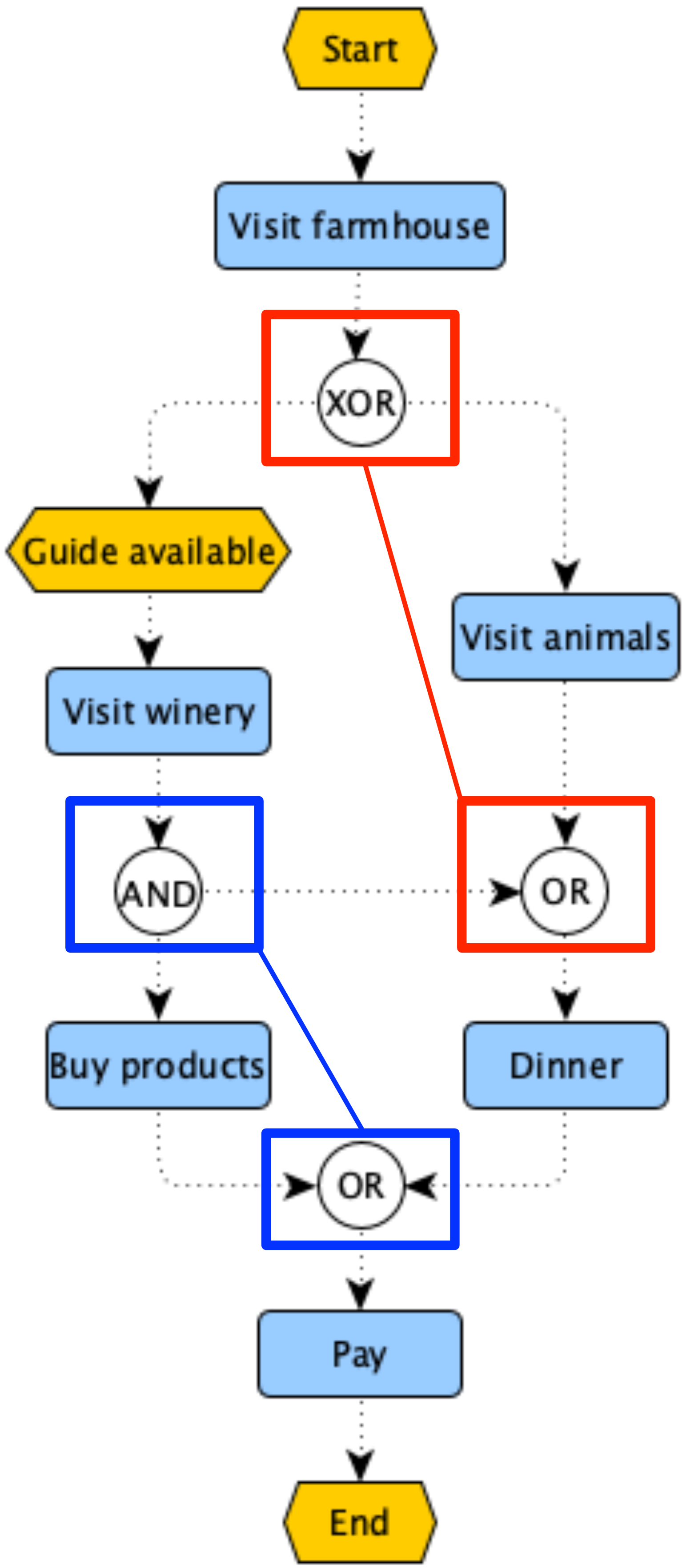


only one
candidate split

Example



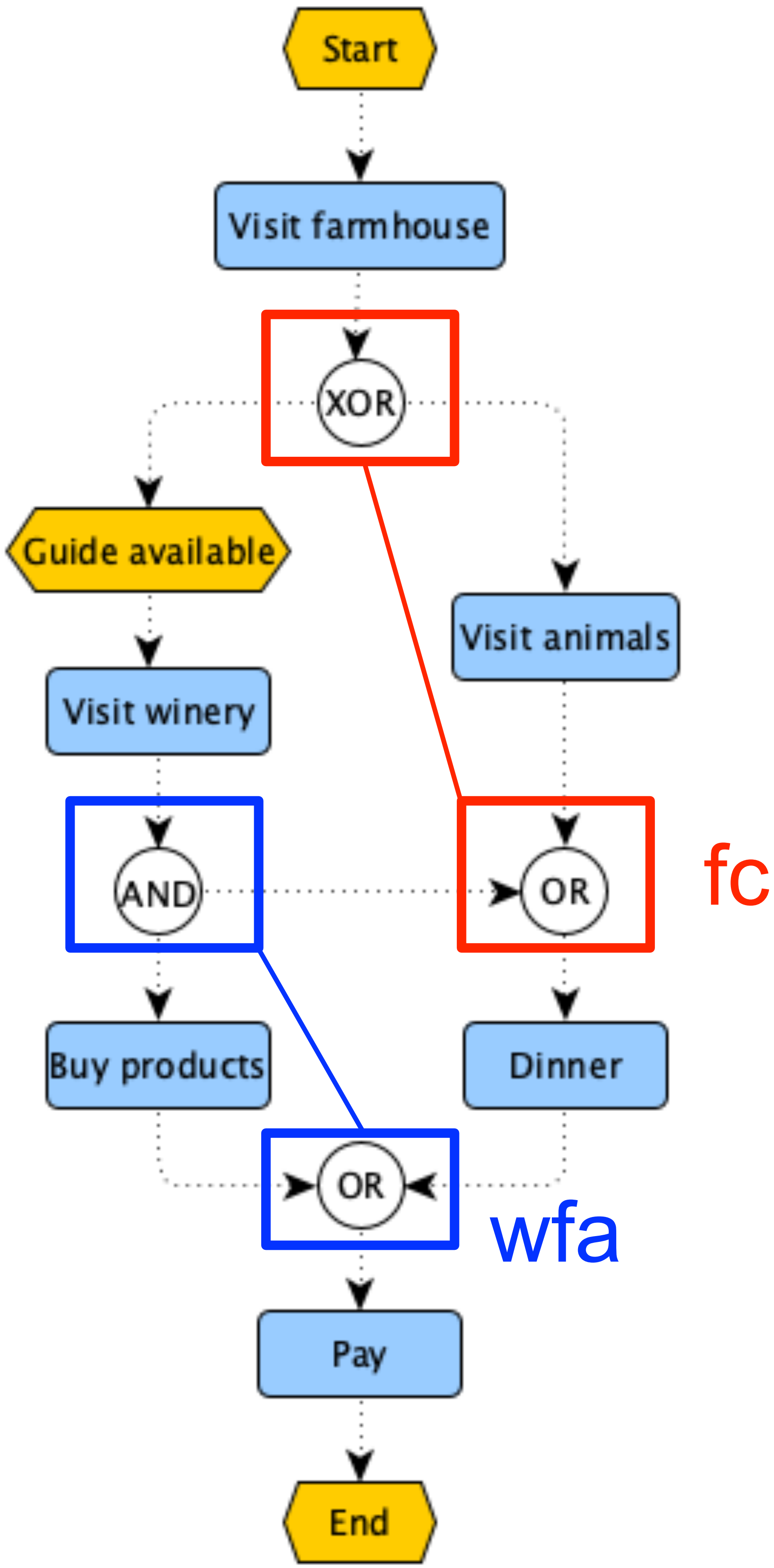
two candidate splits



Example

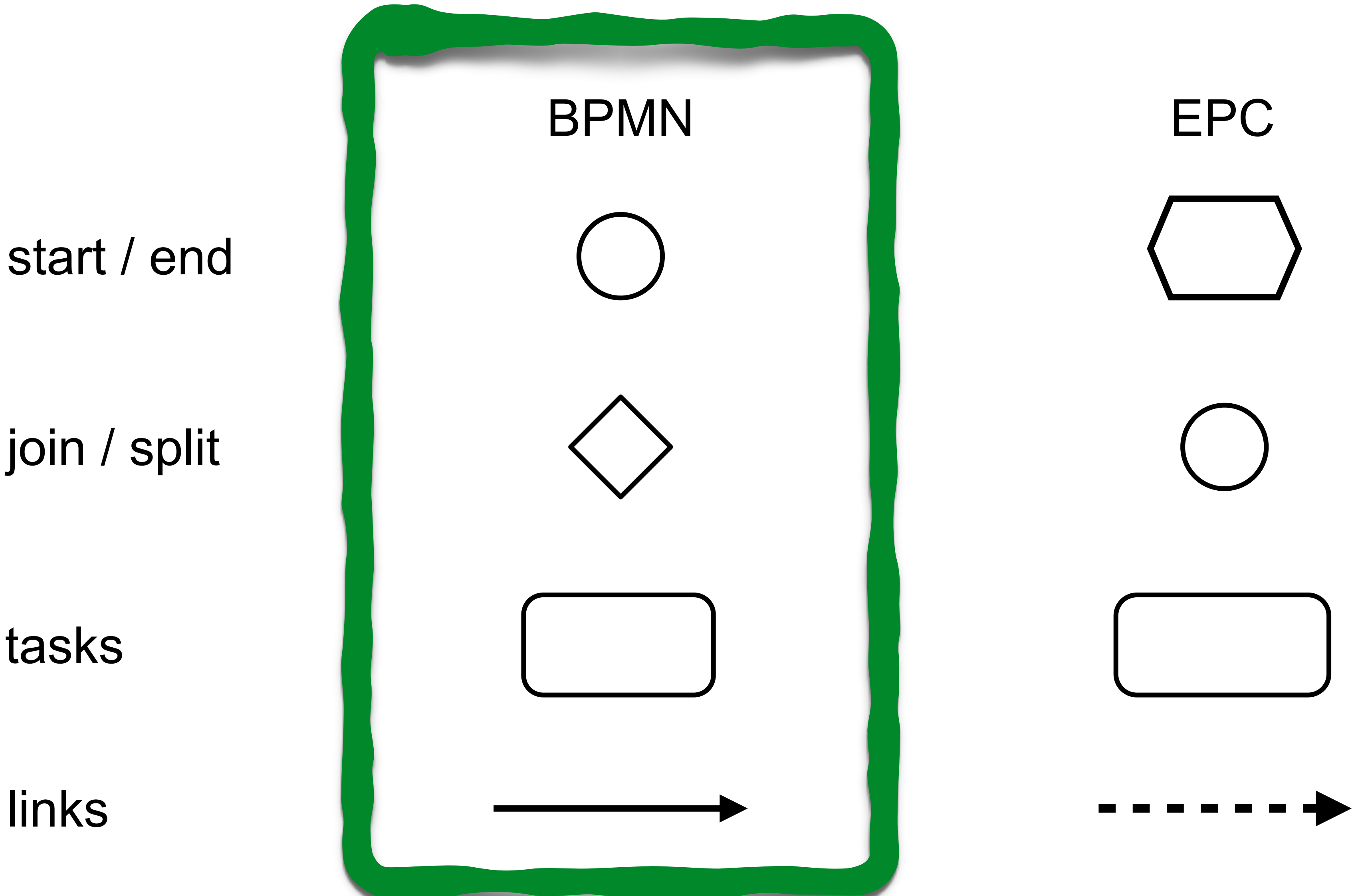
assign corresponding splits

Example



assign policies

A closer look at standards: EPC and BPMN



BPMN

Main goal:

to define a **graphical notation**
that is **readily understandable:**

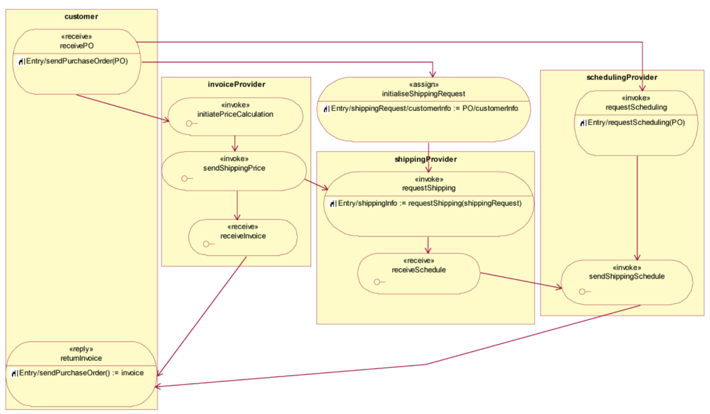
by **business analysts** (initial drafts of processes)

by **technical developers** (process implementation)

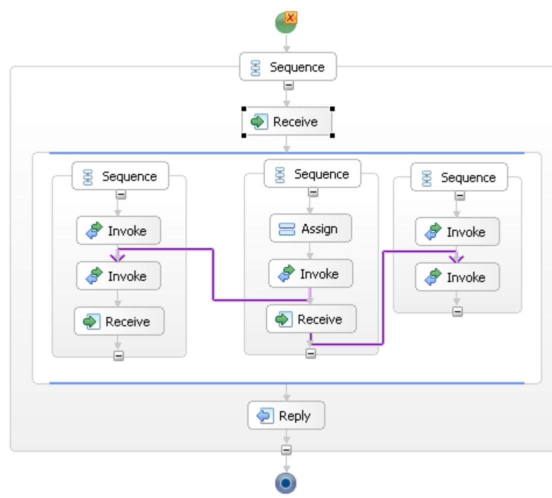
by **business people** (process management)

Before BPMN

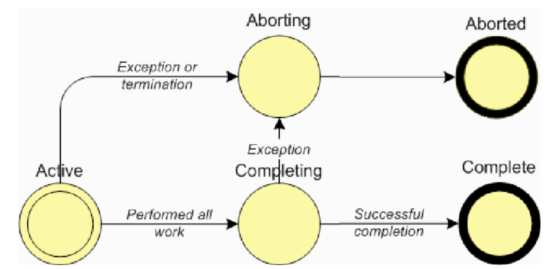
BPMD
BP Definition Metamodel



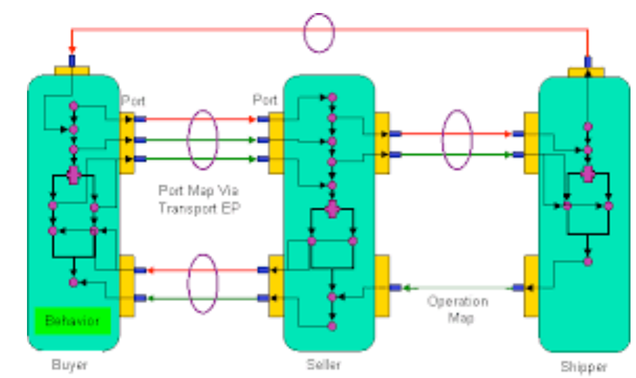
OASIS's BPEL
BP Execution Language



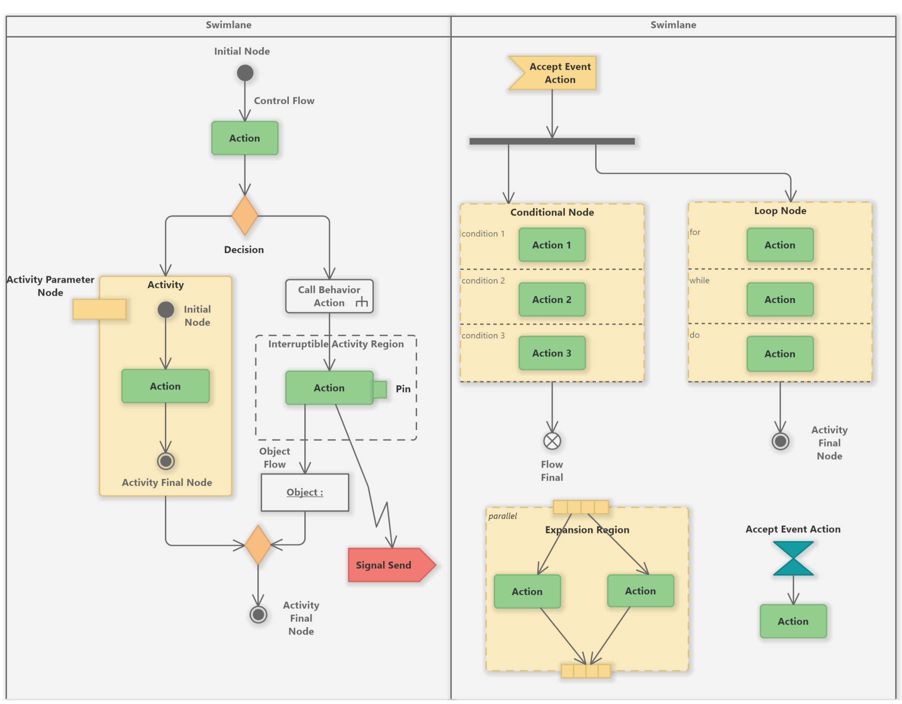
BPMI.org's BPML
BP Modelling Language



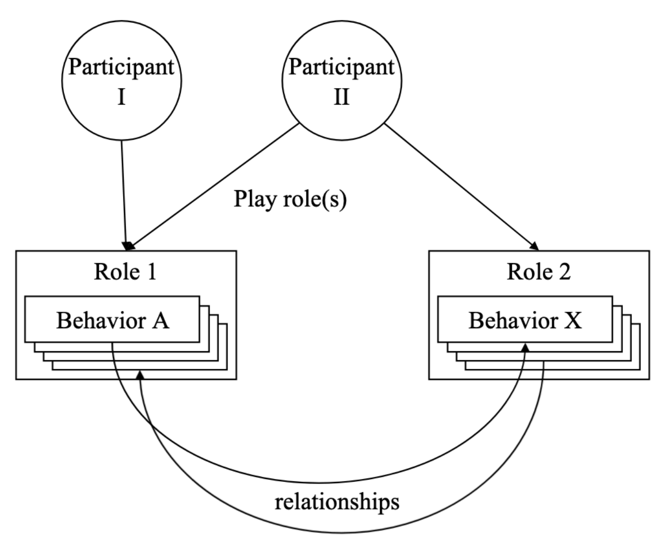
Microsoft's XLANG



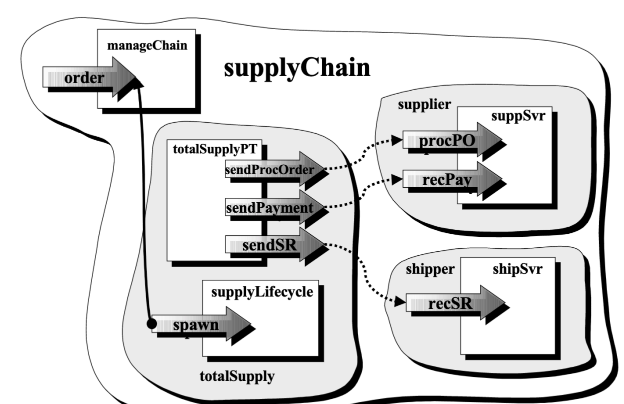
UML2 AD
Activity Diagram



W3C's WS CDL
Choreography Description Language



IBM's WSFL
WS Flow Language



Standardisation

In the context of graphical models for business processes

the development of BMPN is an important step in:

reducing the fragmentation that existed
with myriad of process modelling tools and notations

exploiting experiences with many divergent proposals to
consolidate the best ideas

supporting the adoption of **inter-operable**
business process management systems

A joint effort!

ACTION · Altaworks · ap engines · atg · ATTUNITY This way to B2B · Aventail · AVIS Europe · British American Tobacco · bea · BLACK PEARL

BLAZE SOFTWARE · BOV Aktiengesellschaft · bowstreet · BSOFTB · Trade.com · BUSINESS THREADS · CAP GEMINI ERNST & YOUNG · CASEwise enterprise modeling · C24 · COMPASS

concentus Technology Corporation · consilient · Covasoft · covigo · CROSSWORLDS SOFTWARE · CSC · cyclone commerce · DataChannel · Deloitte Consulting · DHL WORLDWIDE EXPRESS

entricom · Patterns · ec-tone · Enterworks · epiance · epicentric · EVIDIAN A Groupe Bull Company · eXcelon corp. · EXTERPRISE · EXTRICITY

FileNET · fuegotech Go ahead. Change. · FUJITSU · genient · Collaborate Inc. · HANDYSOFT · hp invent · HUBSPAN · i2 · IBM

ILOG · IONA · intraspect · Infosys · Intalio · intel · IPNet · jamcracker · KPMG Consulting · KNOWLEDGE TECHNOLOGIES INTERNATIONAL

LEVEL 8 · Lucent Technologies · Lombardi SOFTWARE · MEGA International · MERANT · mercator · METASERVER · NextSet · NOBILIS Software, Inc. · NORTEL NETWORKS

NOVO · ONTOLOGY.ORG · OPENDESIGN · osm.net · PASSPORT CORPORATION · PEGASYSTEMS · POPKIN SOFTWARE · PROACTIVITY · proforma · propelis

PureEdge · Reef · ripcord SYSTEMS · RYCON TECHNOLOGIES · s1 · SAP · SAVVION · SEEBEYOND · Sherwood INTERNATIONAL · SilverStream

spiritsoft · Staffware · STERLING COMMERCE · Sun microsystems · SUNGARD BUSINESS INTEGRATION · suplicity · SYBASE · Systar · taviz · techspan

TIBCO The Power of Now · TIMOGEN SYSTEMS · UNITECH SYSTEMS, Inc. · versata · VerticalNet · VIGNETTE · VITRIA · webGAIN · XML GLOBAL · XMLFund

Short history

2000 - **Business Process Management Initiative** (BPMI.org) (independent organization, studying **open specifications** for the management of **e-Business processes**)

2005 - BPMI and the Object Management Group™ (OMG™) merge their activities on BPM forming the **Business Modeling & Integration Domain Task Force** (**BMI -DTF**)

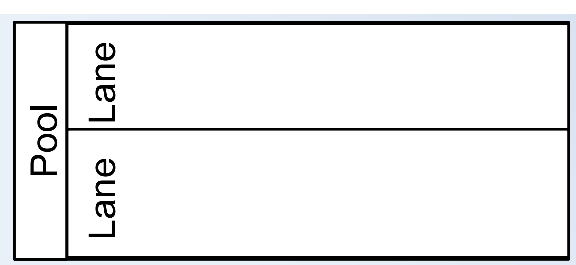
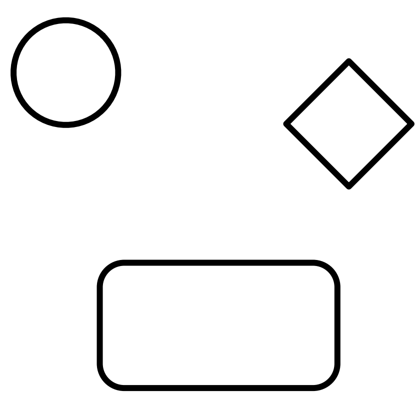
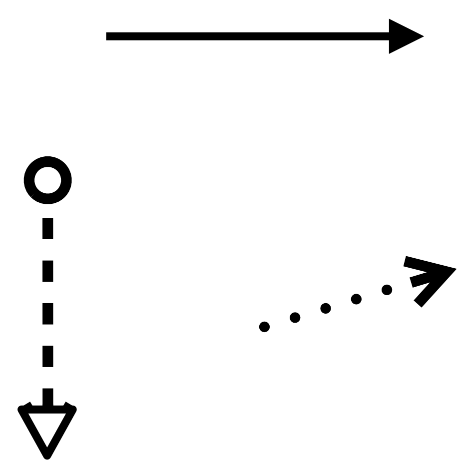
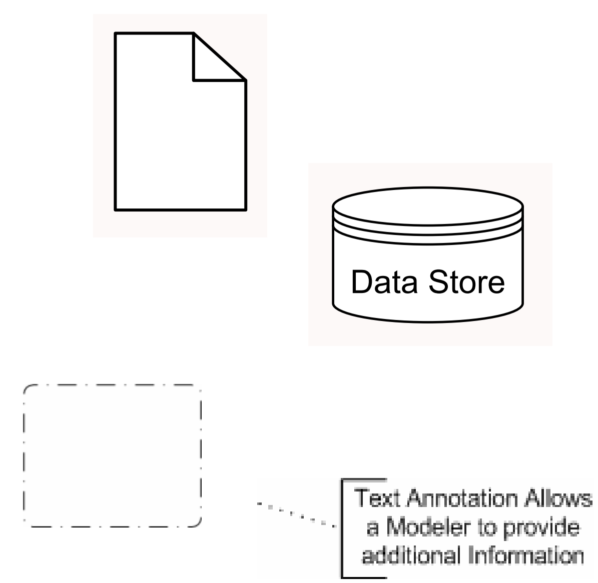
2006 - **BPMN 1.0 approved**
2007 - BPMN 1.1 approved
2009 - BPMN 1.2 approved

2009 - BPMN 2.0 Beta 1 proposed
2010 - BPMN 2.0 Beta 2 proposed
2011 **- BPMN 2.0 Final delivered**

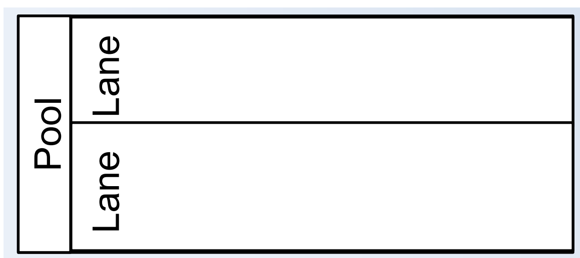
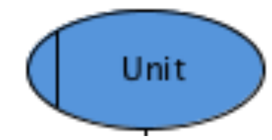
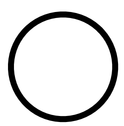
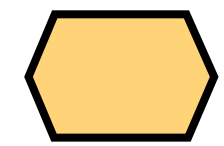
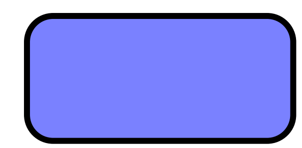
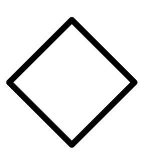
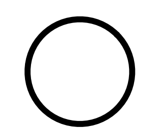
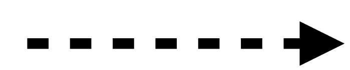
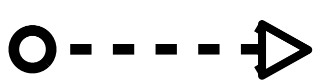
Business process diagrams

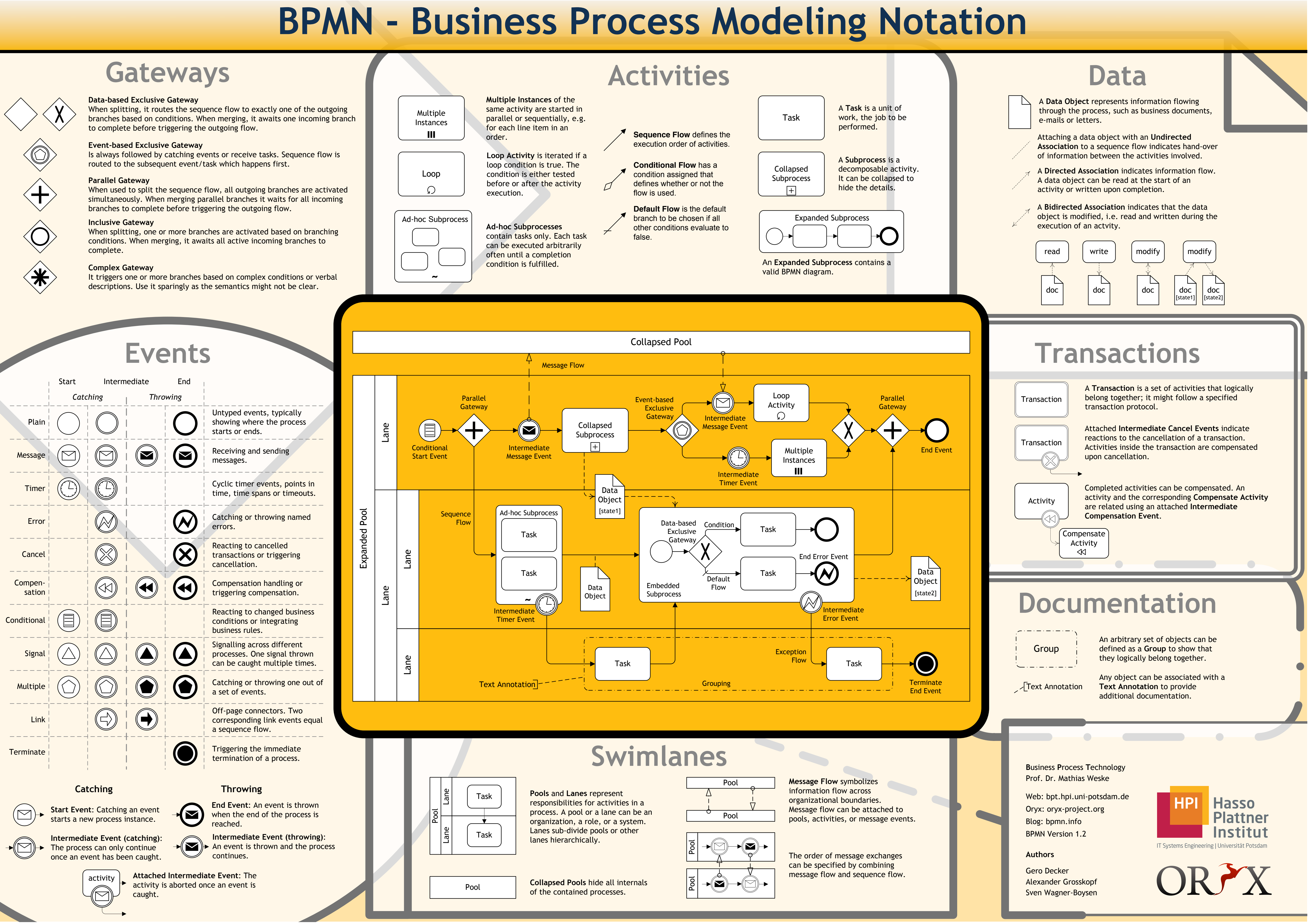
BPMN defines a standard for
Business Process Diagrams (**BPD**)
based on **flowcharting technique**

Four categories of elements

swimlanes	flow objects	connecting objects	artefacts
			

BPMN vs EPC (roughly)

	swimlanes	organization unit	
	event	event	
	activity	function	
	gateway	connector	
	sequence flow	control flow	
	message flow		

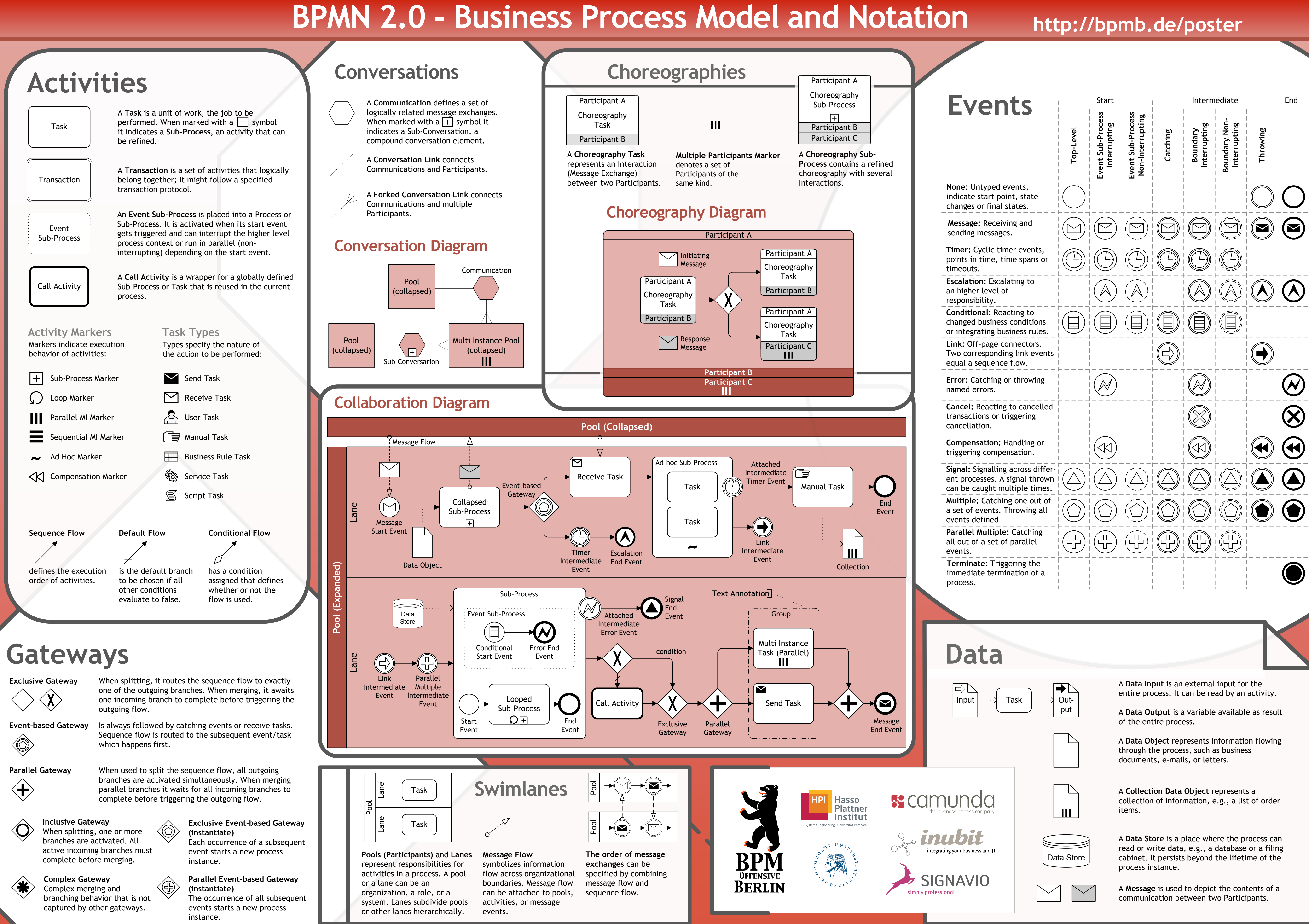


BPMN 1.0 poster

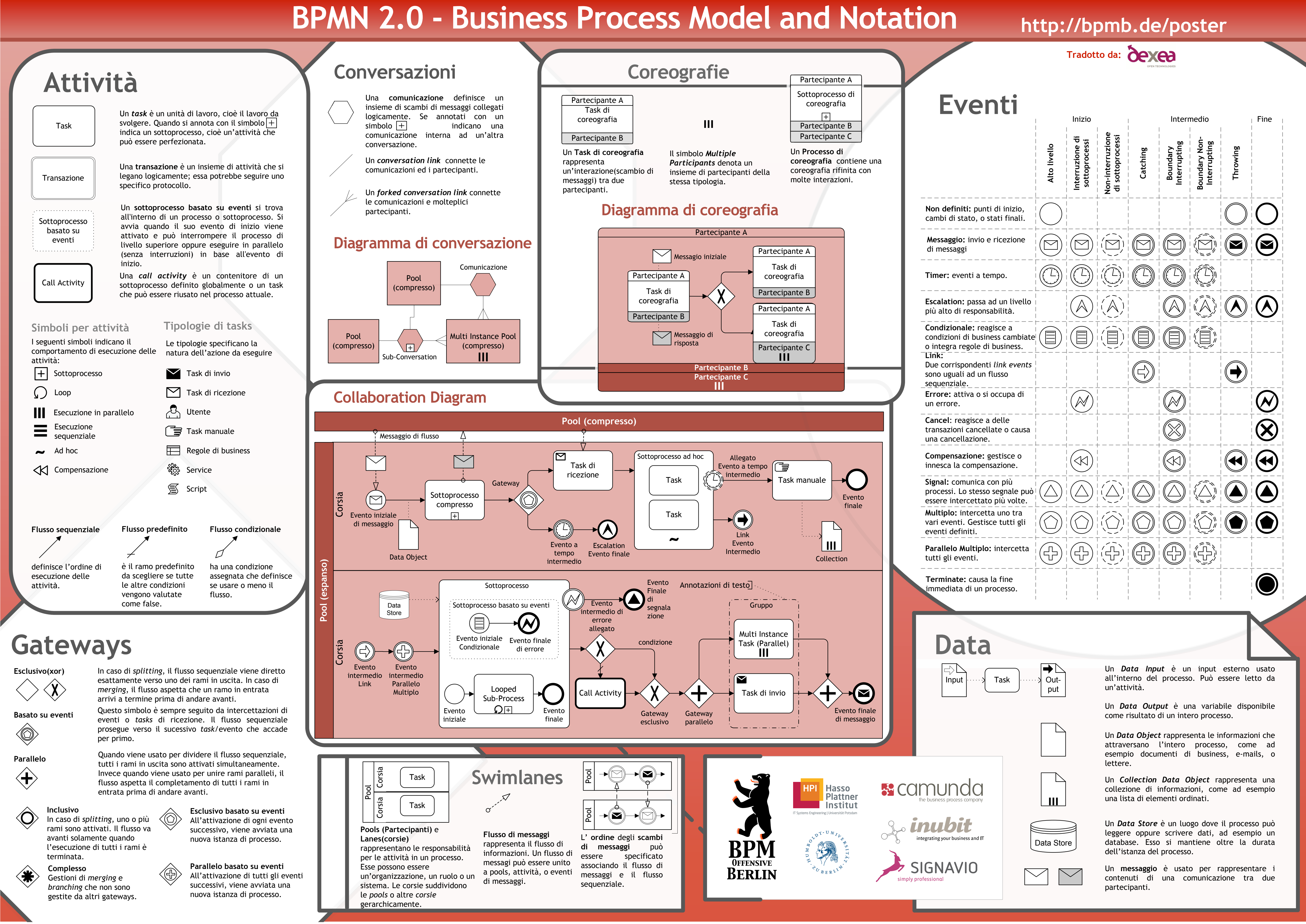
BPMN 2.0 vs 1.0

Updated (new markers):
Events
Activities
Gateways
Artefacts

Added:
Choreographies
Full metamodel
XML Serialization
Diagram Interchange
BPMN Execution Semantics (verbal)



BPMN 2.0 poster



BPMN 2.0 poster (in Italian)

BPMN 2.0 (2009/11) FAQ

What is BPMN?

BPMN is a graphical notation that depicts the steps (end to end flow) in a business process.

Specifically designed to coordinate the sequence of processes and **the messages that flow** between participants in a related set of activities.

Why is BPMN important?

The world of business processes has changed dramatically over the past few years. Processes can be coordinated from behind, within and over organizations boundaries. A business process now spans multiple participants and coordination can be complex.

Until BPMN, there has not been a standard modelling technique developed that addresses these issues. BPMN provides users with a **royalty free notation**.

This will benefit users in a similar manner in which UML standardised the world of software engineering.
There will be training courses, books and a body of knowledge that users can access in order to better implement a business process.

Will there be a major rewrite?

Not for 2 or 3 years…

(good work! 13+ years and still no revision is planned)

Pros and Cons

Strengths of BPMN

Simplicity:
A small set of basic symbols

Extensibility:
many decorations available
(new ones can be added in the future)

Graphical design: **intuitive**

Generality: **orchestration + collaboration + choreography**

Tool availability:
`.bpmn` **exchange format**

Weaknesses of BPMN

over 100 graphical elements

verbose description (500 pages)

difficult to learn comprehensively:
different readings of the same diagram are possible

different BPMN vendors implement the execution of
BPMN diagrams in different ways
(and for different subsets)

BPMN basics

Swimlanes (pools, lanes)

Swimlanes

A **swimlane** is a mechanism
to **organise activities into separate visual categories**
to illustrate different capabilities or **responsibilities**

Present in many process modelling methodologies

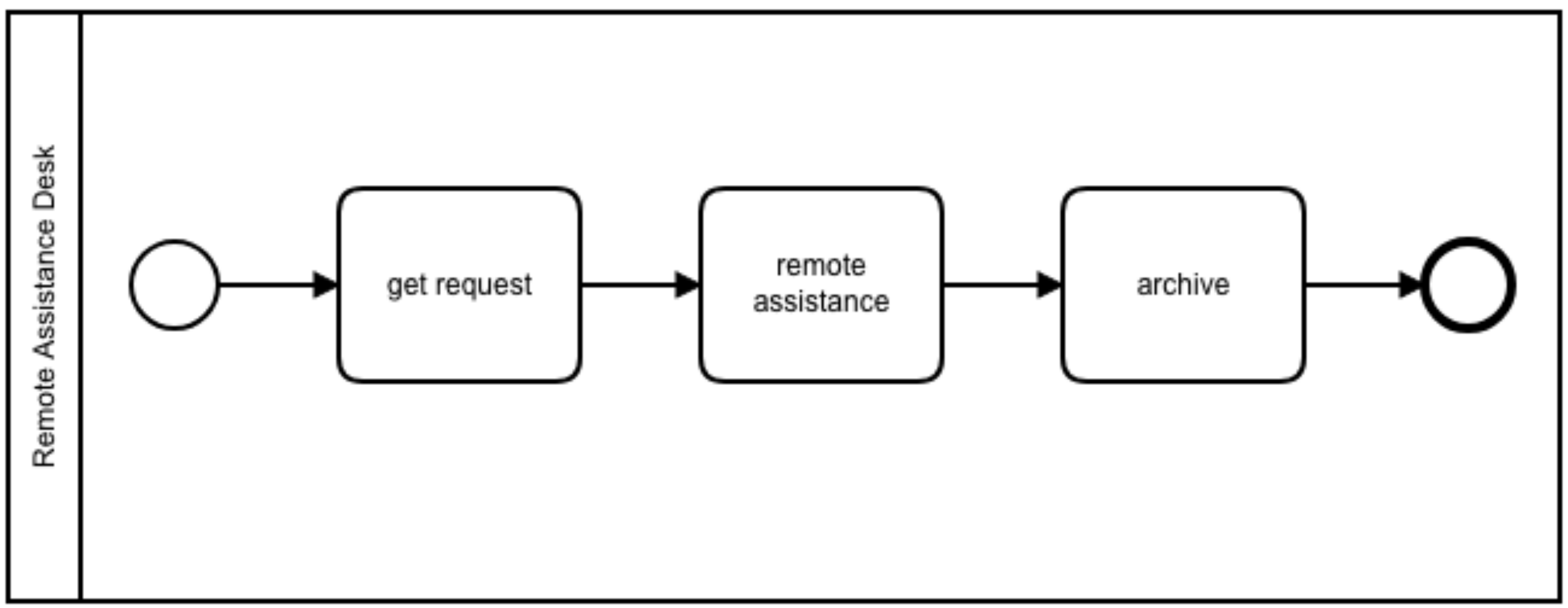
BPMN supports two main swimlane objects:

pool

lanes

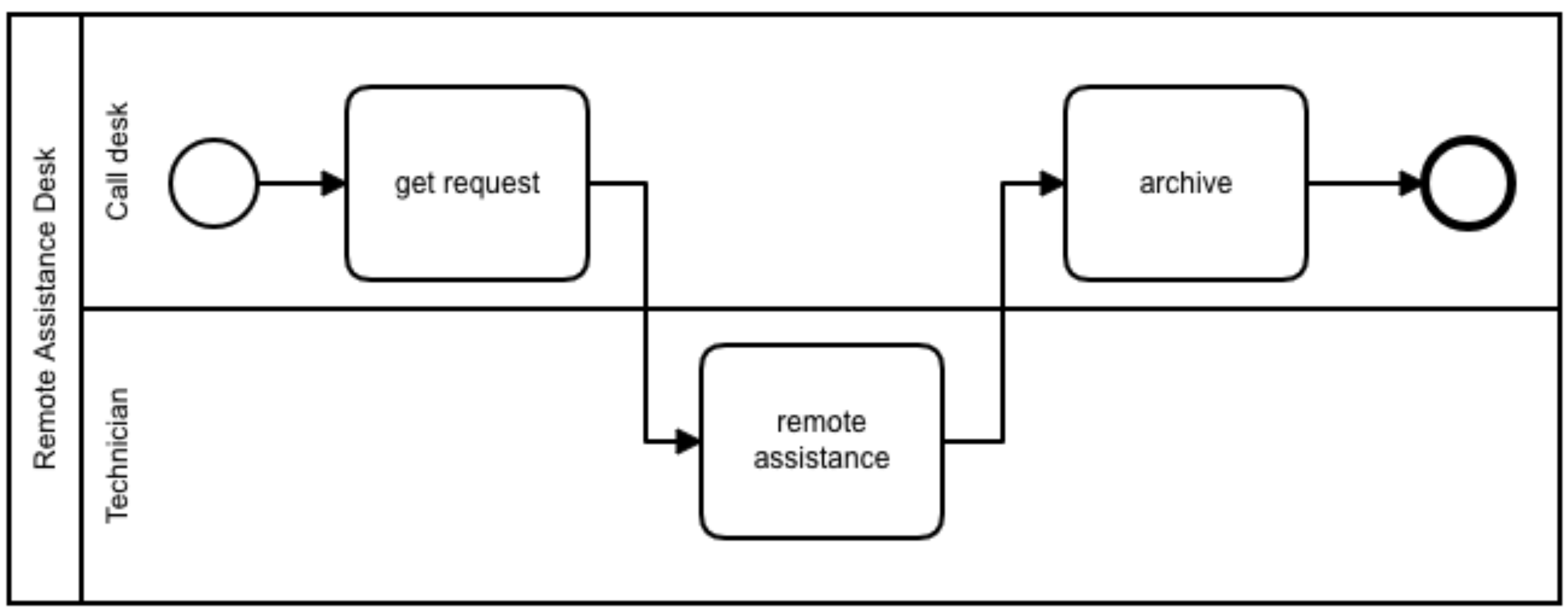
Pools

A **pool** represents a participant (or role) in a process (represented as a rectangle with a name)



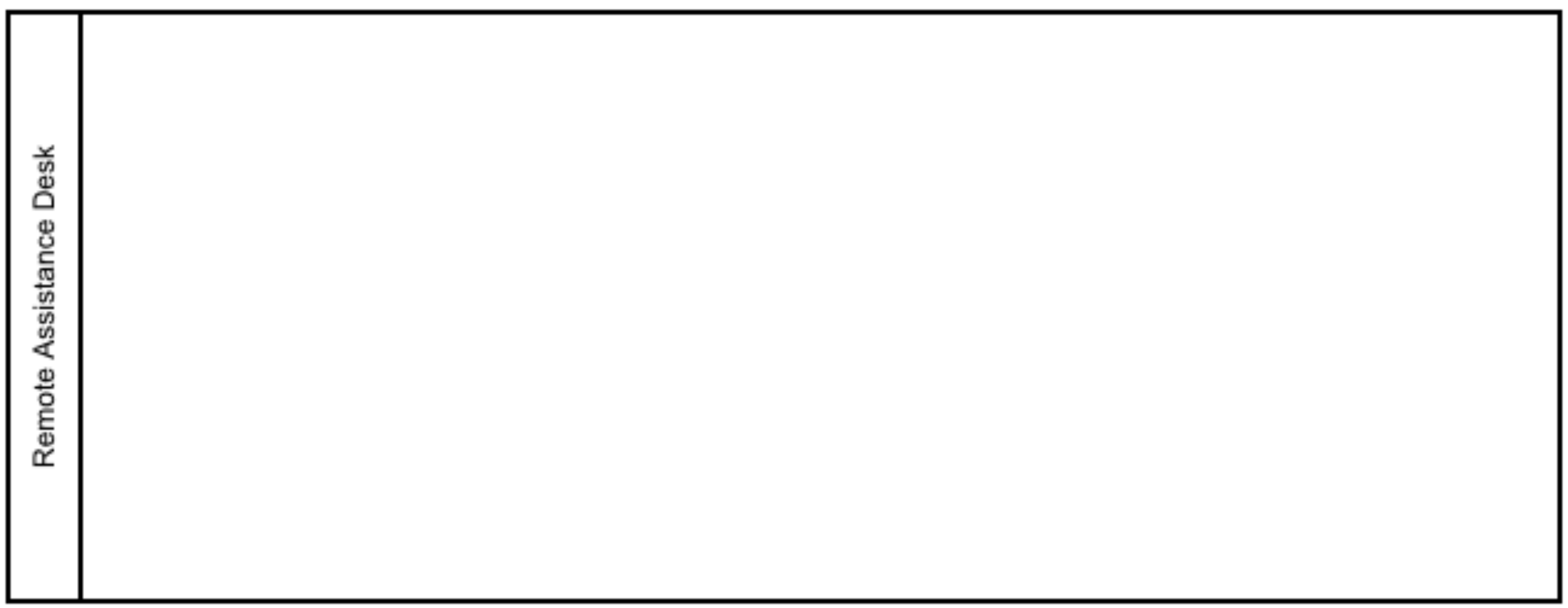
Lanes

A **lane** is a hierarchical sub-partition within a pool that is used to organise and categorise activities (inner rectangle that extends to the entire length of the pool



Collapsed pools

Internal process is not exposed
(like a black-box)



Requirements

A Pool MUST contain 0 or 1 business process.

A Pool can contain 0 or more lanes.

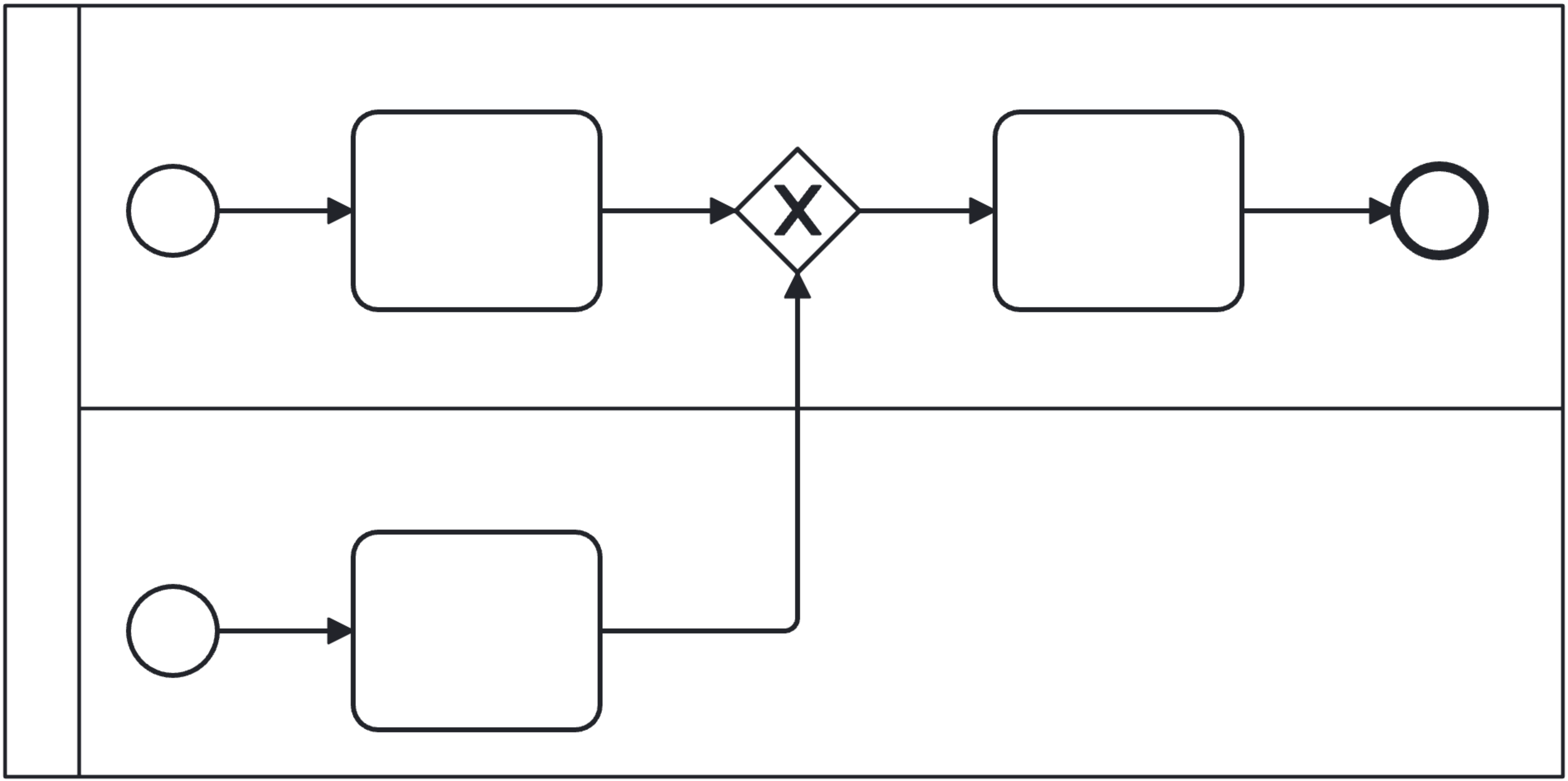
Two pools can only be connected with message flows.

Naming conventions:
a noun possibly preceded by an adjective

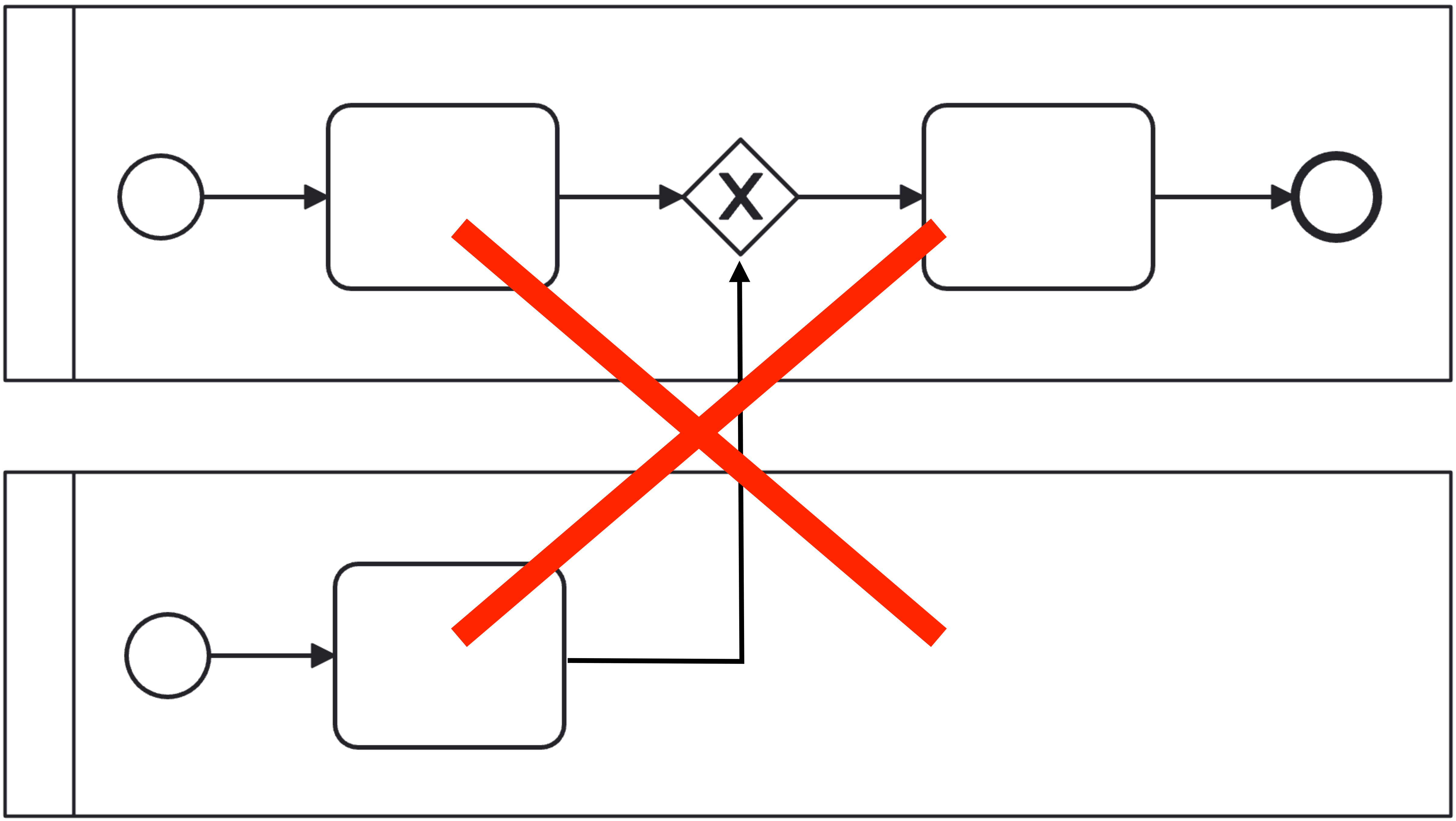
the label is often obtained by "nominalizing" the verb that describes the main action in the process (e.g., claim handling, order fulfillment)

Avoid long labels
Articles are often omitted

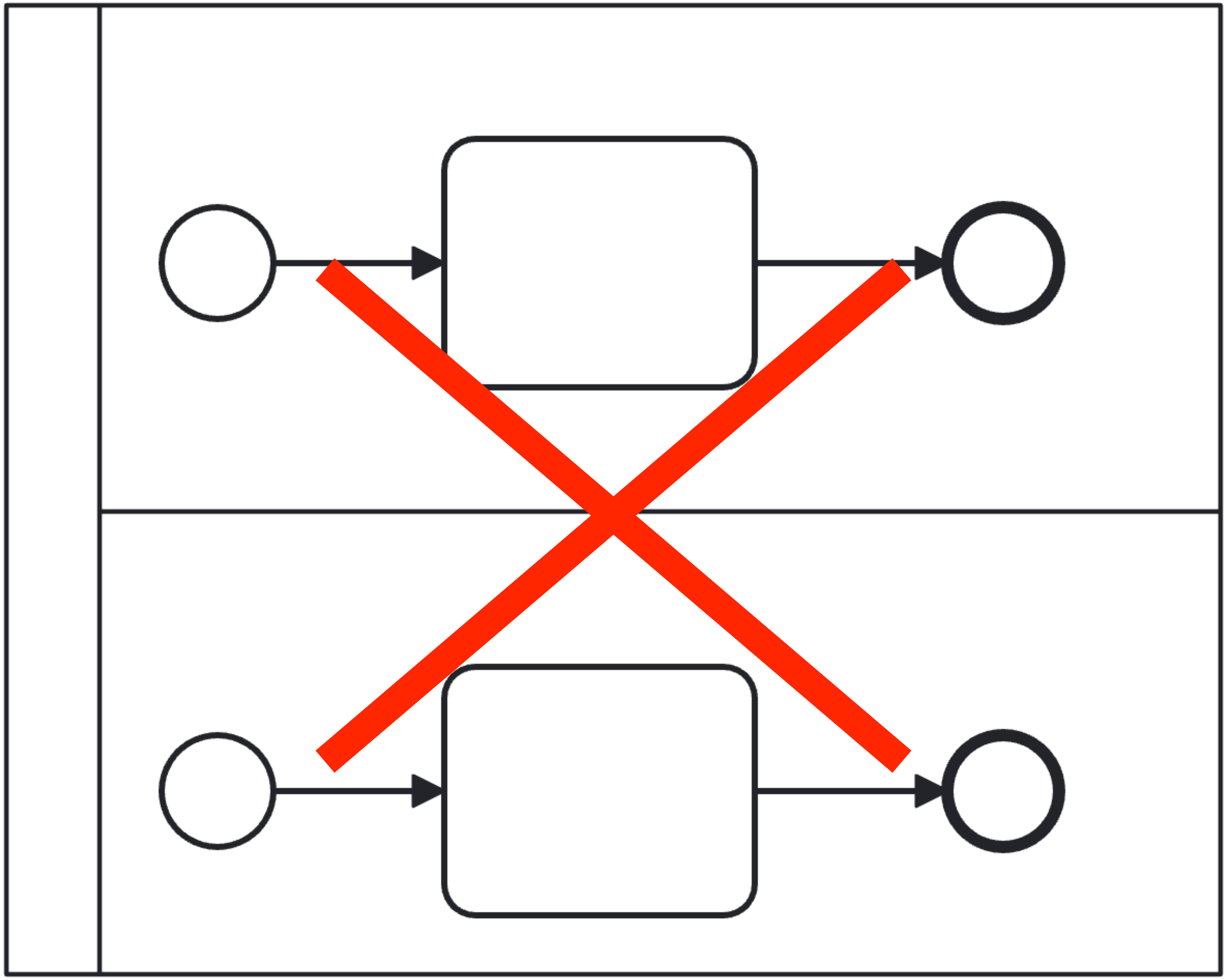
Example



Example



Example



Flow Objects
(events, activities, gateways)

Flow objects

Rationale:

fix a small set of core elements
so that modellers must learn a small number of shapes:

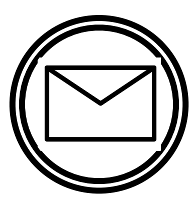
events **activities** **gateways**

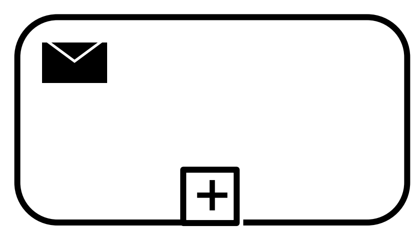
Flow objects

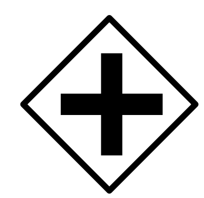
Rationale:

fix a small set of core elements
so that modellers must learn a small number of shapes:

events **activities** **gateways**







use different border styles and internal markers
to add many more information
(this way the notation is **extensible**)

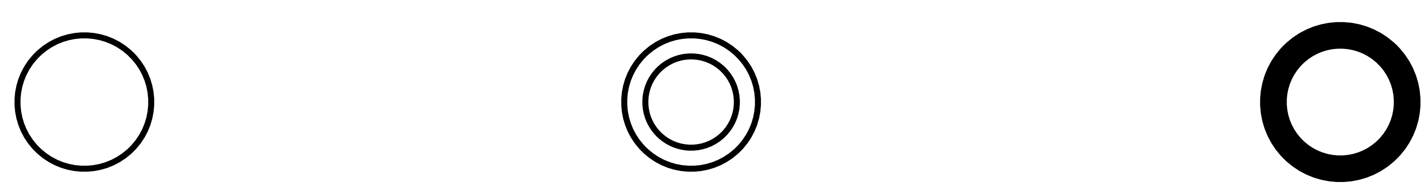
Flow objects: Events

Events

An **event** is something that “happens” during the course of a business process

An event is represented as a circle
different borders define the **type** of the event

start **intermediate** **end**



Naming conventions

Events:
the label should begin with a noun and end with a verb in past participle form to indicate something that just happened (e.g., Invoice emitted)

the noun can be preceded by an adjective (e.g., Urgent order sent)

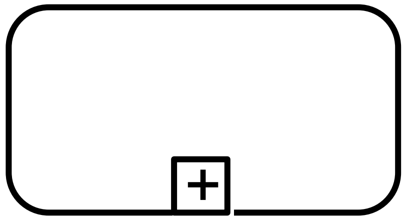
Avoid long labels
Articles are often omitted

Flow objects: Activities

Activities

An **activity** is some “unit of work” (job) to be done during the course of a business process

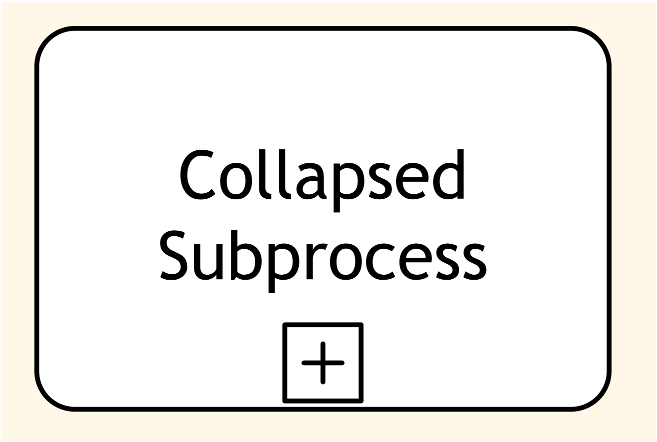
An activity is represented as a rounded box
BPMN has two main types of activities
atomic (**task**) or compound (**sub-process**)

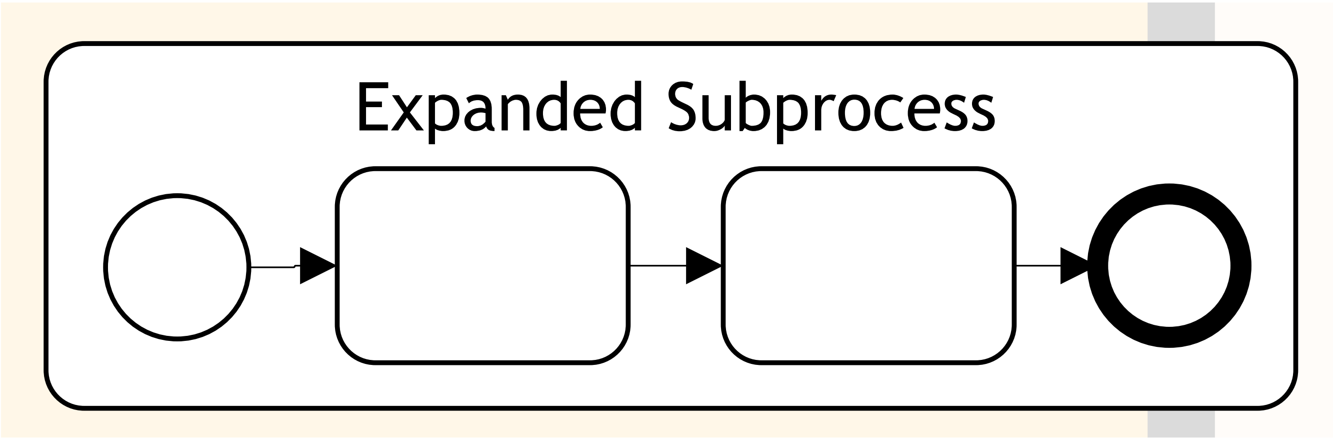


Sub-processes

Large process models are hard to parse:
we improve readability
by hiding certain parts within sub-processes

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

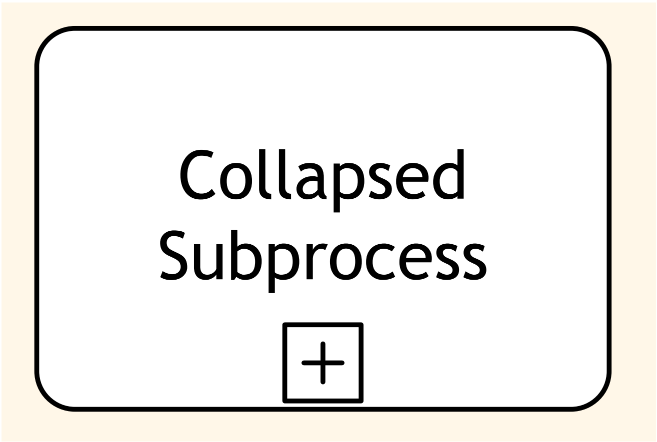


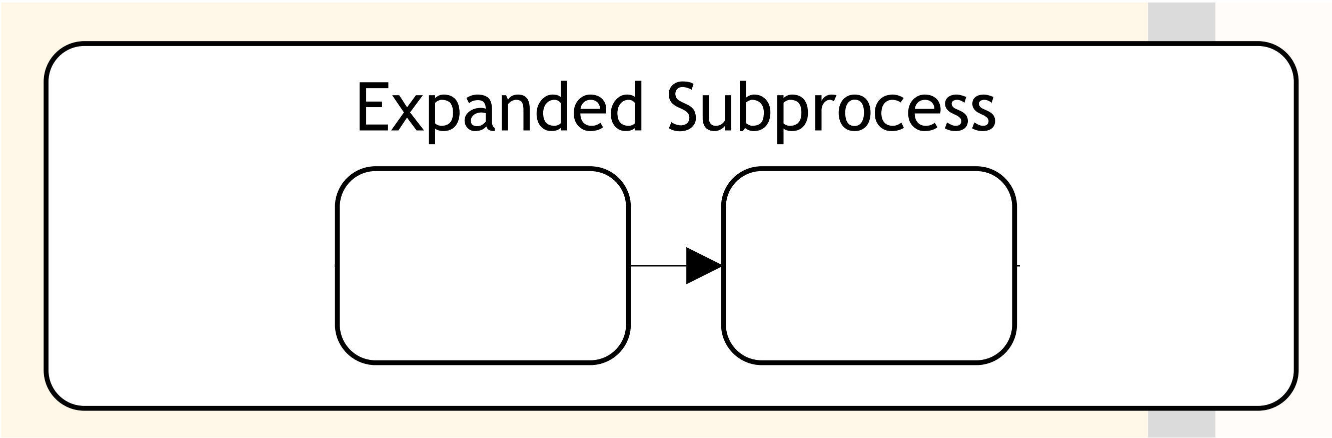


Sub-processes

Large process models are hard to parse:
we improve readability
by hiding certain parts within sub-processes

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





implicit start / end

Naming conventions

Activities:
verb in the imperative form followed by a noun
(e.g., Approve order)

the noun can be preceded by an adjective
(e.g., Issue driver license)

the verb may be followed by a complement
(e.g., Renew driver license via offline agencies)

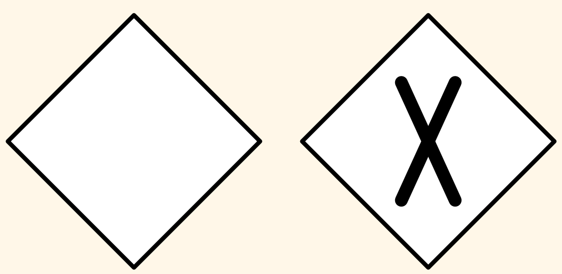
Avoid long labels
Articles are often omitted

Flow objects: Gateways

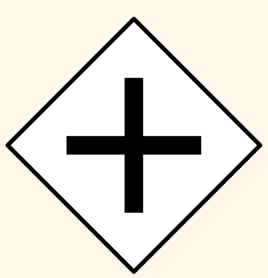
Gateways

A **gateway** is used to split/join the sequence flow (conditional, fork, wait)

A gateway is represented as a diamond shape
internal markers indicate the nature of behaviour control



Data-based Exclusive Gateway
When splitting, it routes the sequence flow to exactly one of the outgoing branches based on conditions. When merging, it awaits one incoming branch to complete before triggering the outgoing flow.



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.

Connecting objects

(sequence flow, message flow, association)

Connecting objects

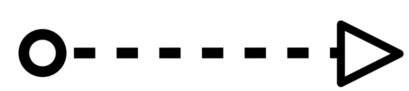
The Flow objects are connected together in a diagram to create the basic skeletal structure of a business process

Three connecting objects can be used:

Sequence flow

connected objects must **reside in the same pool** (but they can be in different lanes)

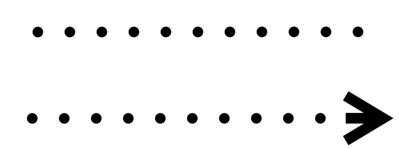
Message flow



connected objects must **reside in different pools**

to be discussed later

Association



connects flow objects with artefacts

to be discussed later

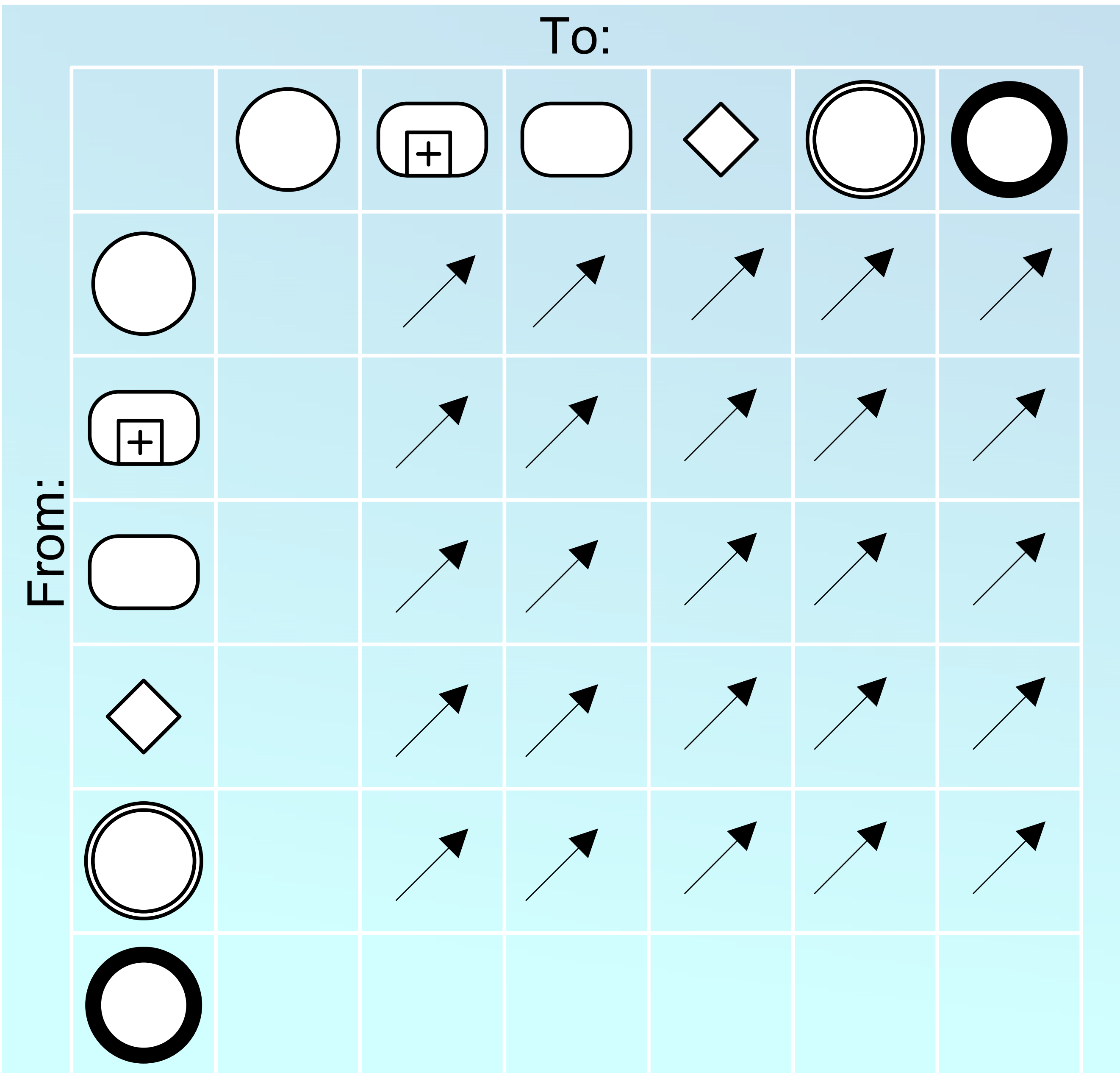
Sequence flow

A **sequence flow** is used to show the order in which activities are to be performed

the term “control flow” is generally avoided in BPMN

A sequence flow is represented by a solid line with a solid arrowhead

Requirements



each event:
at most one incoming and
at most one outgoing
sequence flow

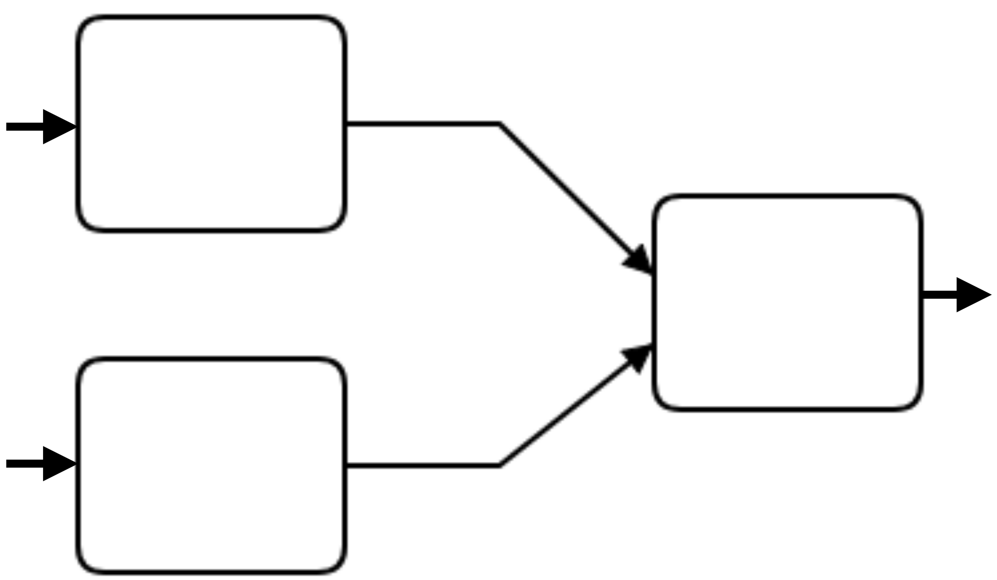
each activity:
exactly one incoming and
exactly one outgoing
sequence flow

each gateway:
one-to-many,
many-to-one,
many-to-many

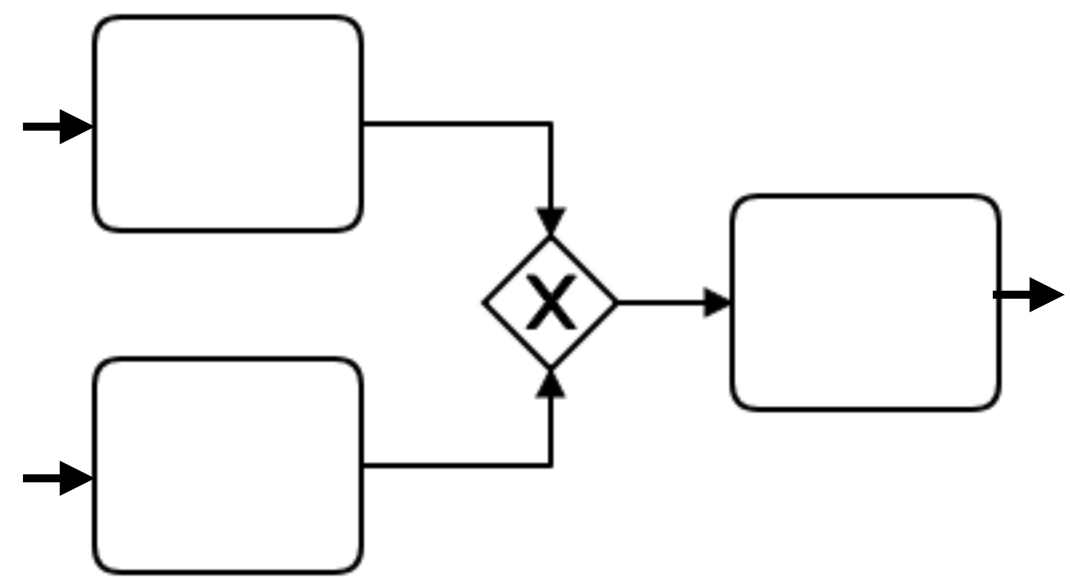
Multiple flows and implicit gateways

In principle each activity should have exactly:
one incoming arc, one outgoing arc

Be careful if this is not the case!



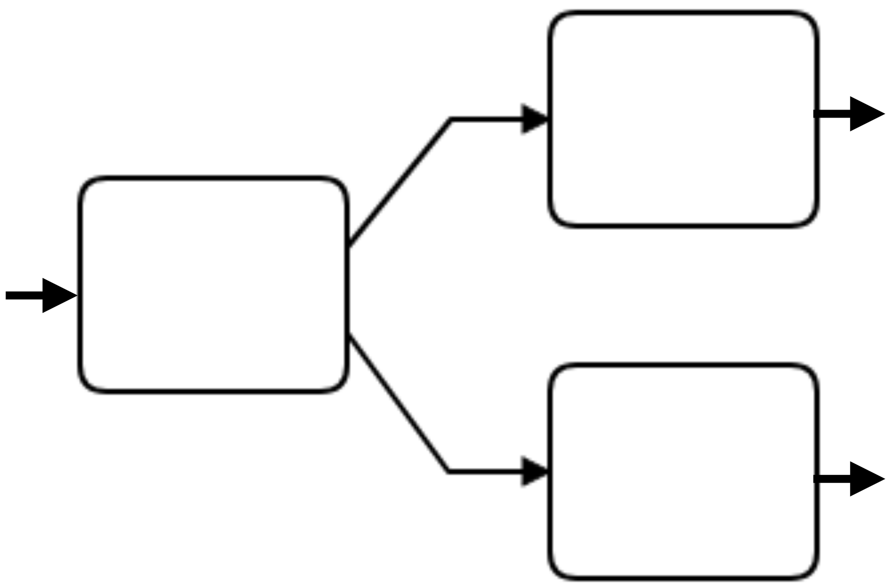
stands for



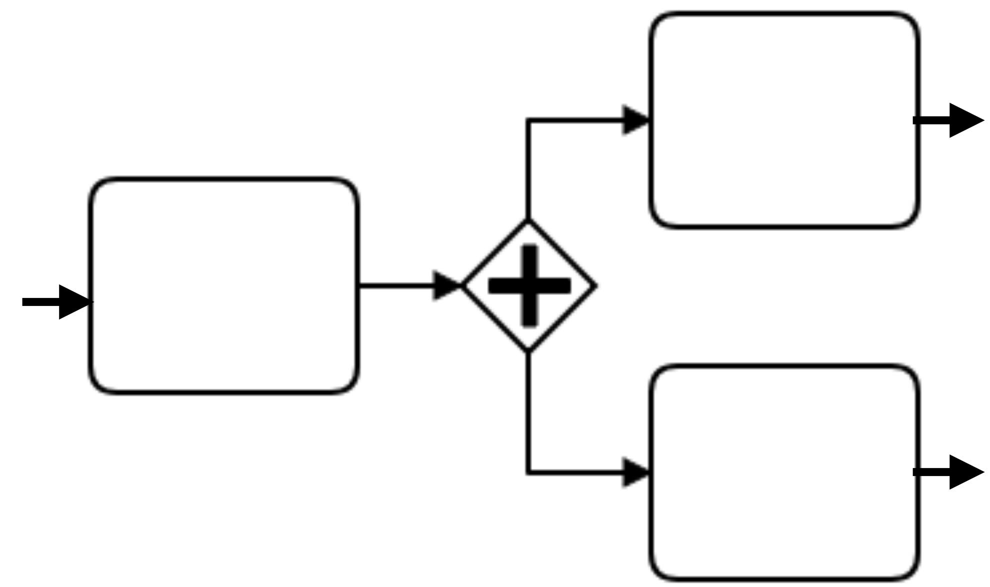
Multiple incoming flows are mutually exclusive

Multiple flows and implicit gateways

In principle each activity should have exactly:
one incoming arc, one outgoing arc
Be careful if this is not the case!

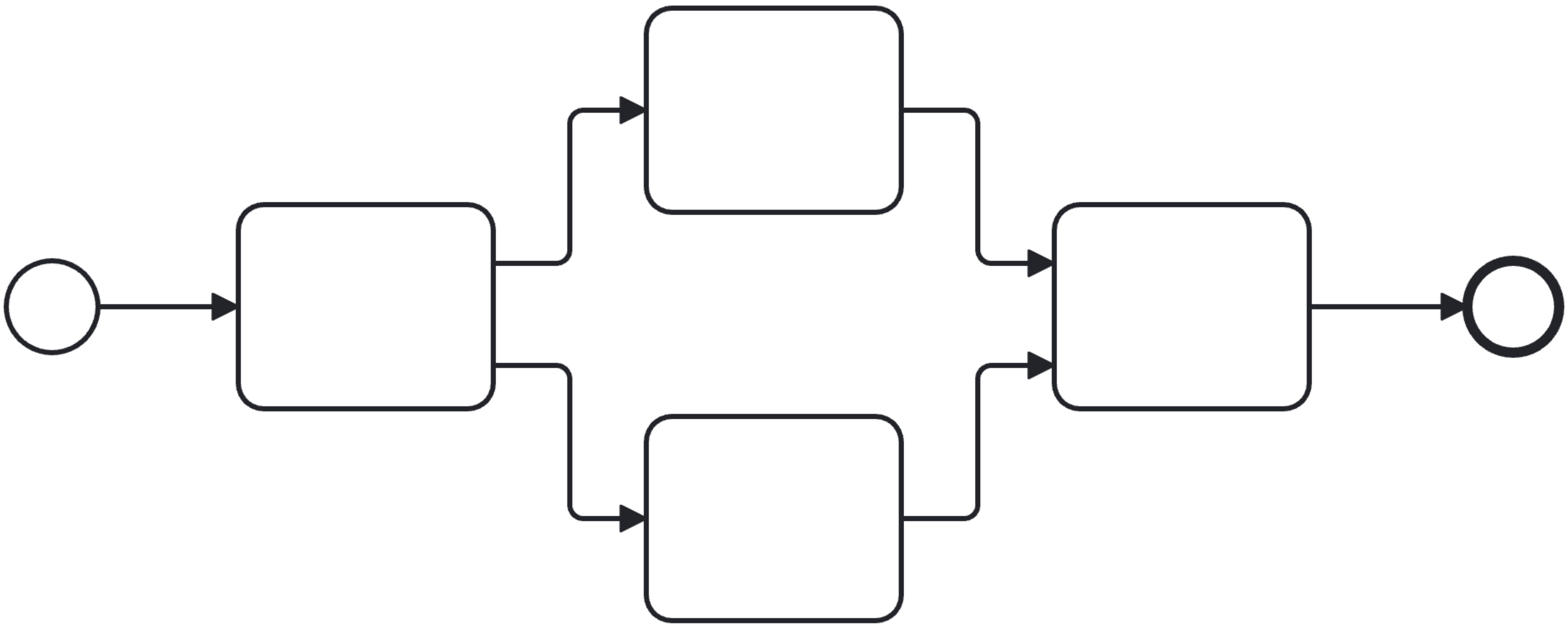


stands for



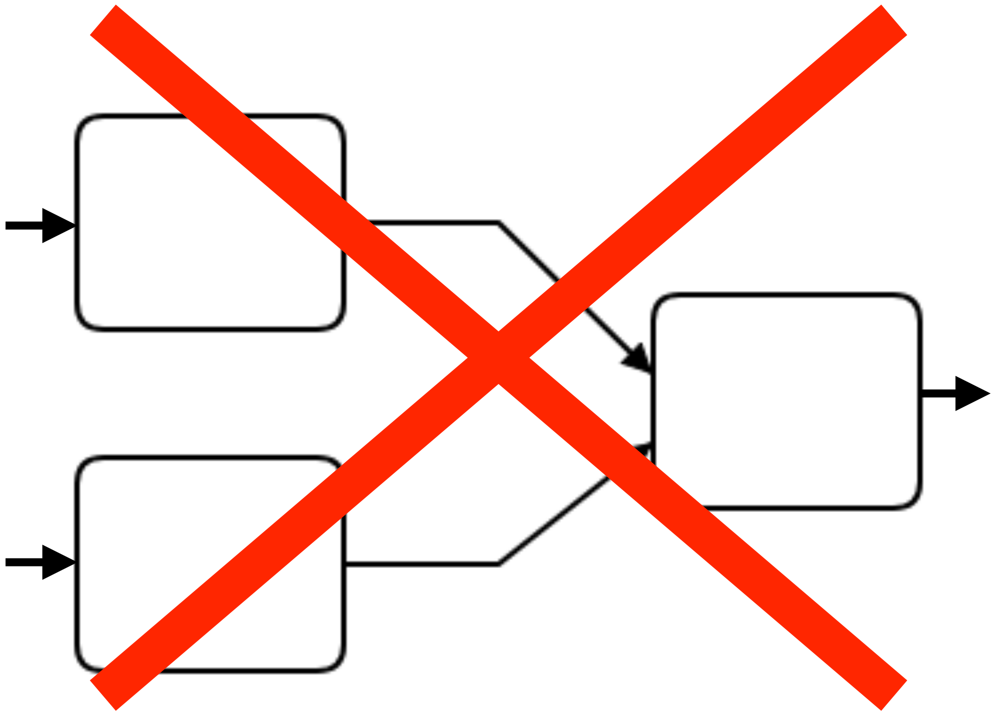
Multiple outgoing flows are activated in parallel
(unless conditions are attached to them)

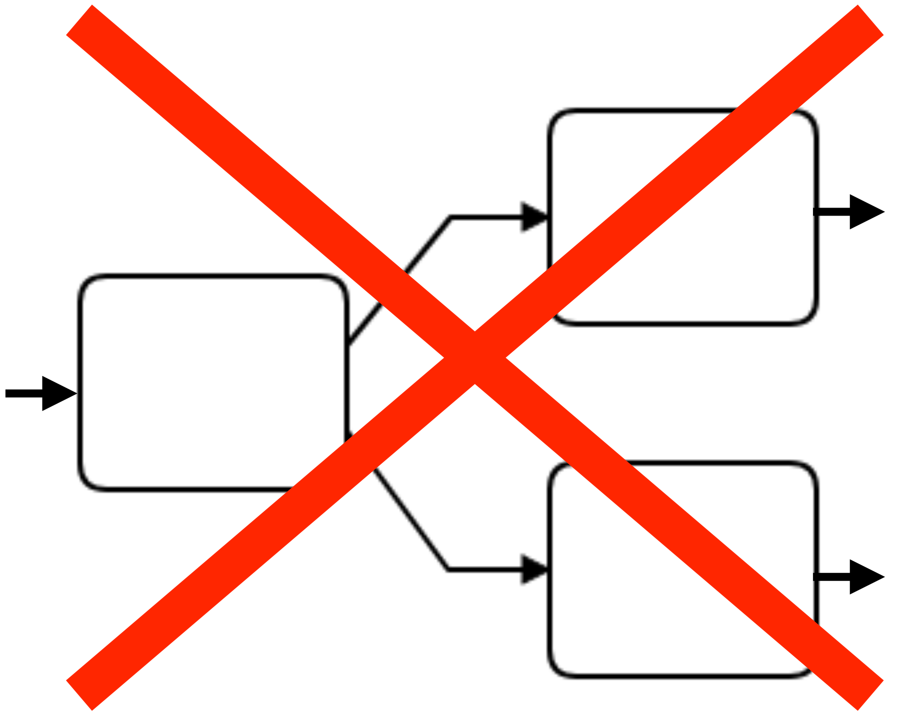
Hidden issue!



In your final projects

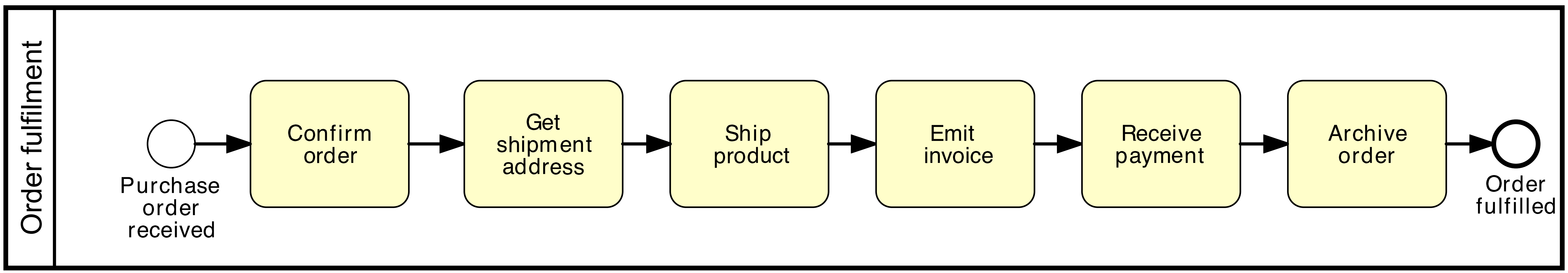
Please avoid



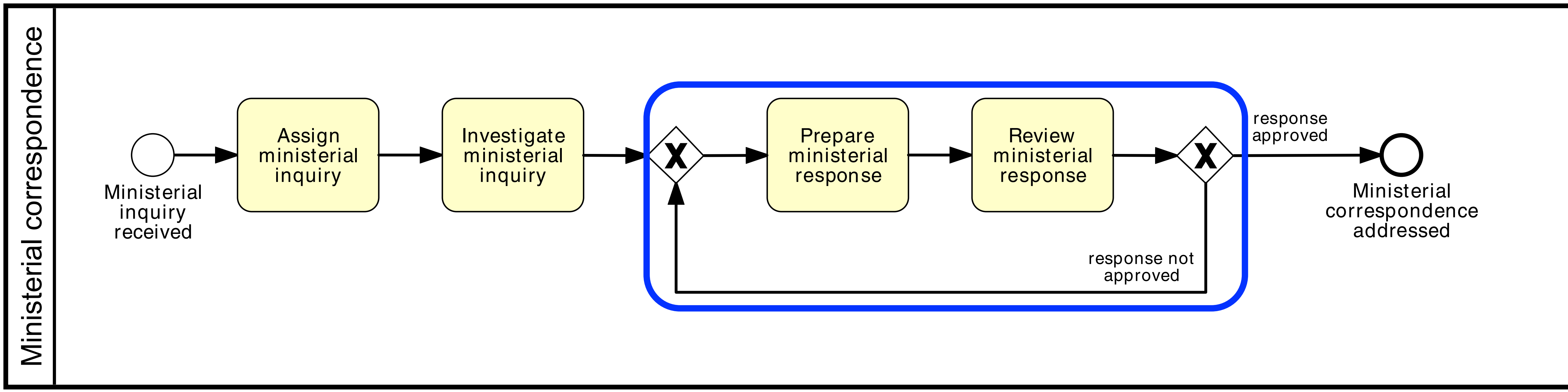


Typical patterns

Sequence: order fulfilment

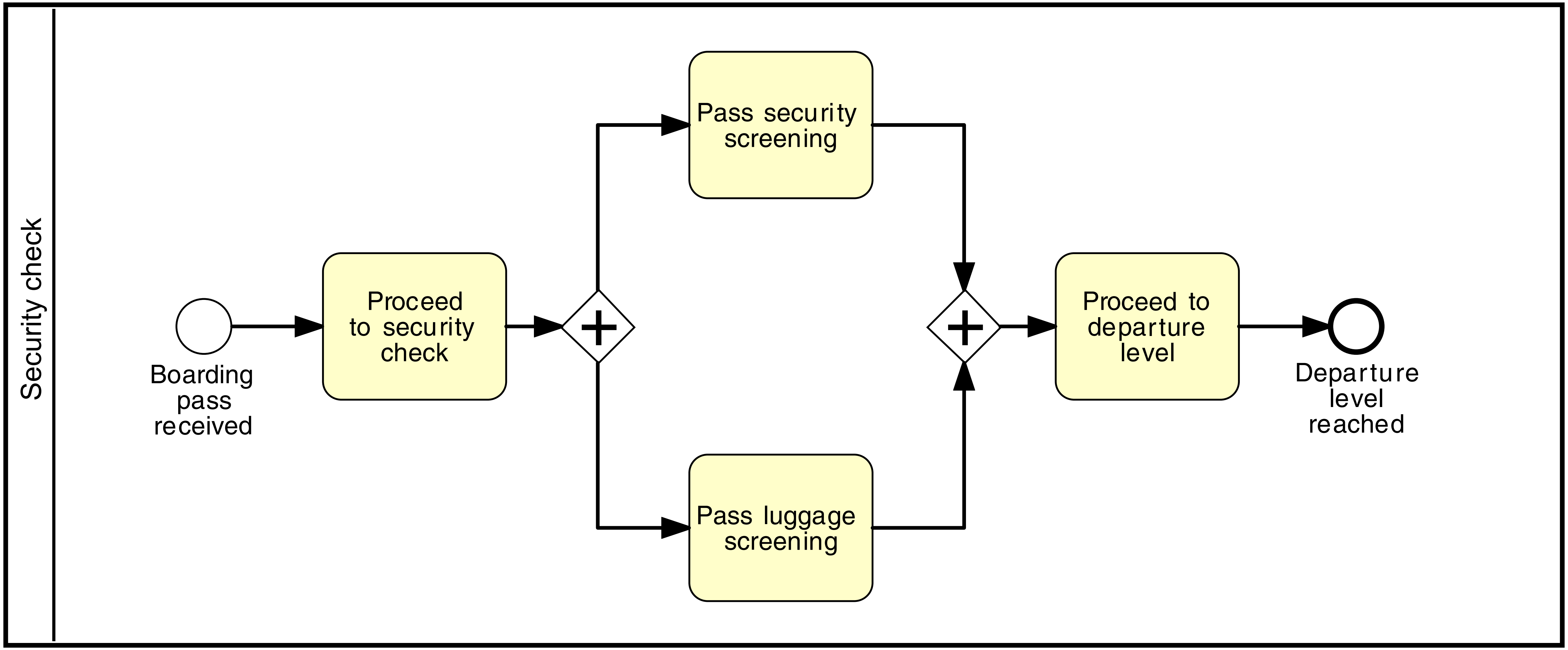


Rework and repetition: ministerial correspondence

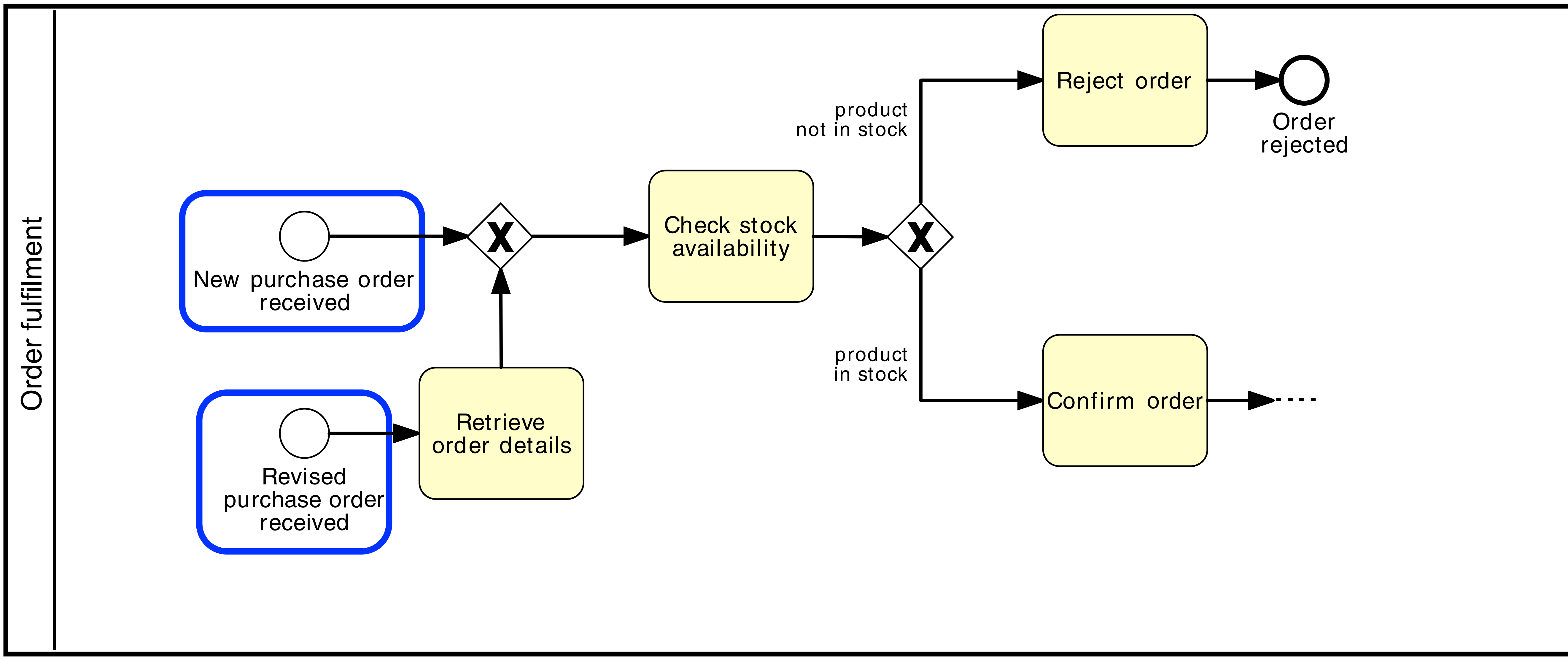


A repetition block starts with a XOR-join
and ends with a decision gateway (XOR-split)

Parallel activities: airport security check

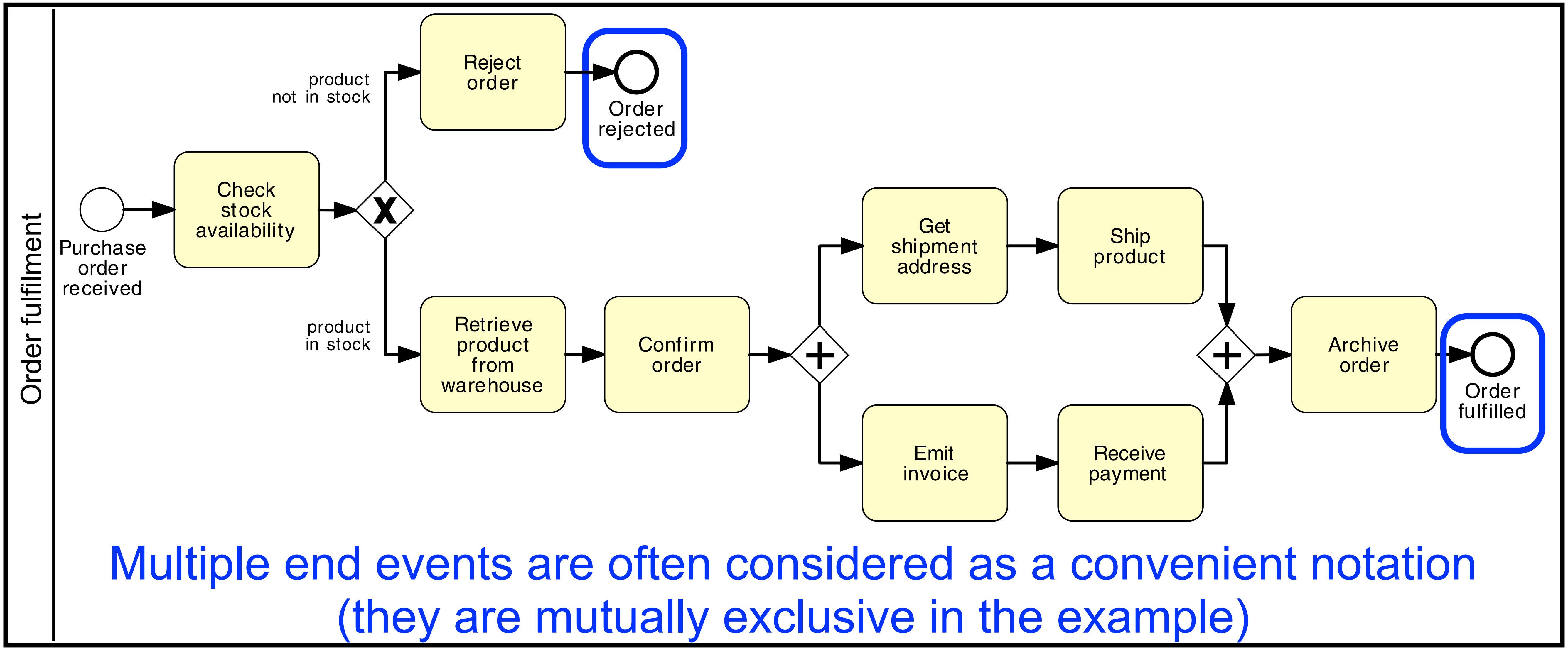


Multiple start events: order fulfilment



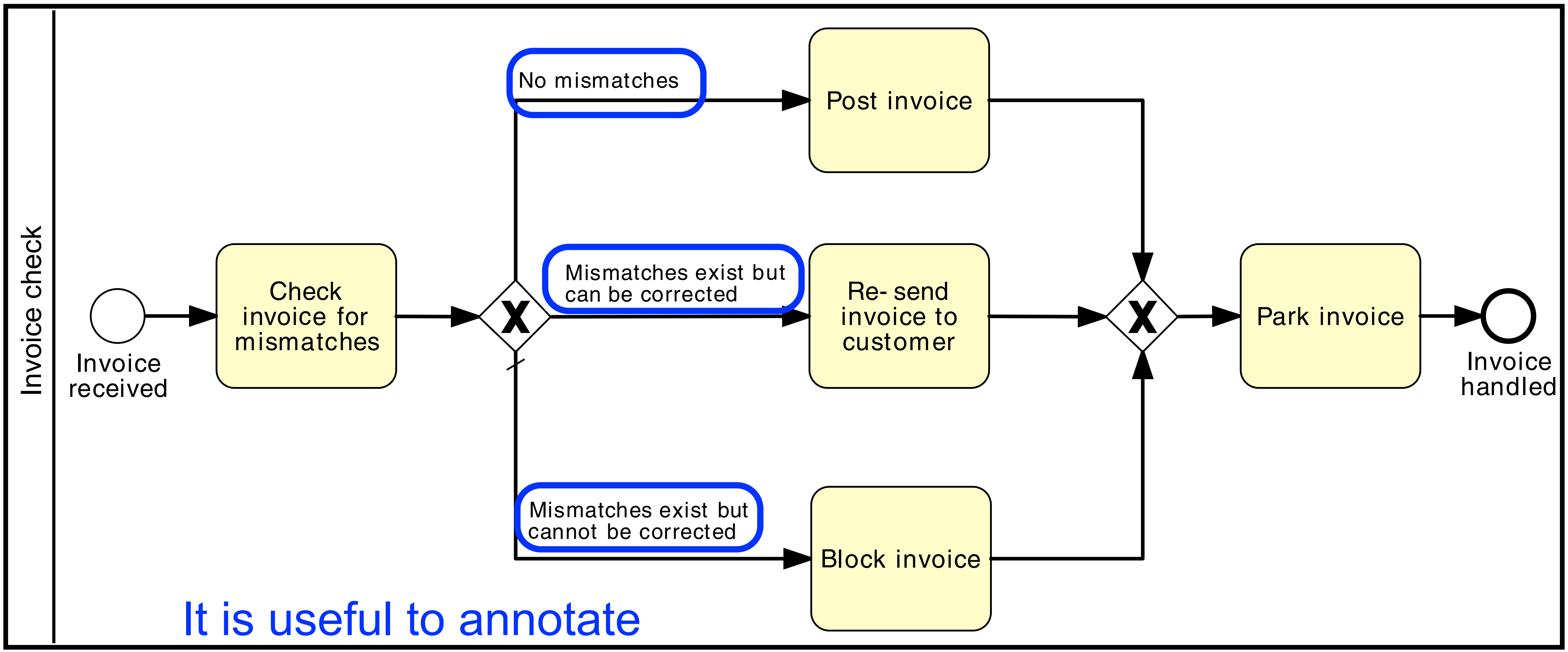
Multiple start events are often considered as a convenient notation (they capture mutually exclusive triggers to start a process instance)

Multiple end events: order fulfilment



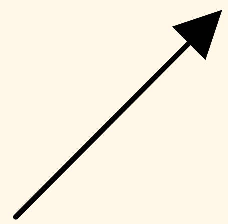
BPMN adopts **implicit termination** semantics:
a case ends only when each ``token'' reaches the end

Exclusive decisions: invoice checking process

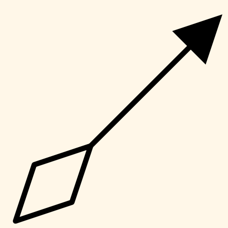


It is useful to annotate branches with the conditions under which they are taken

Annotated sequence flow

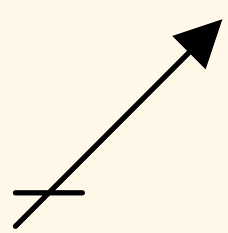


Sequence Flow defines the execution order of activities.



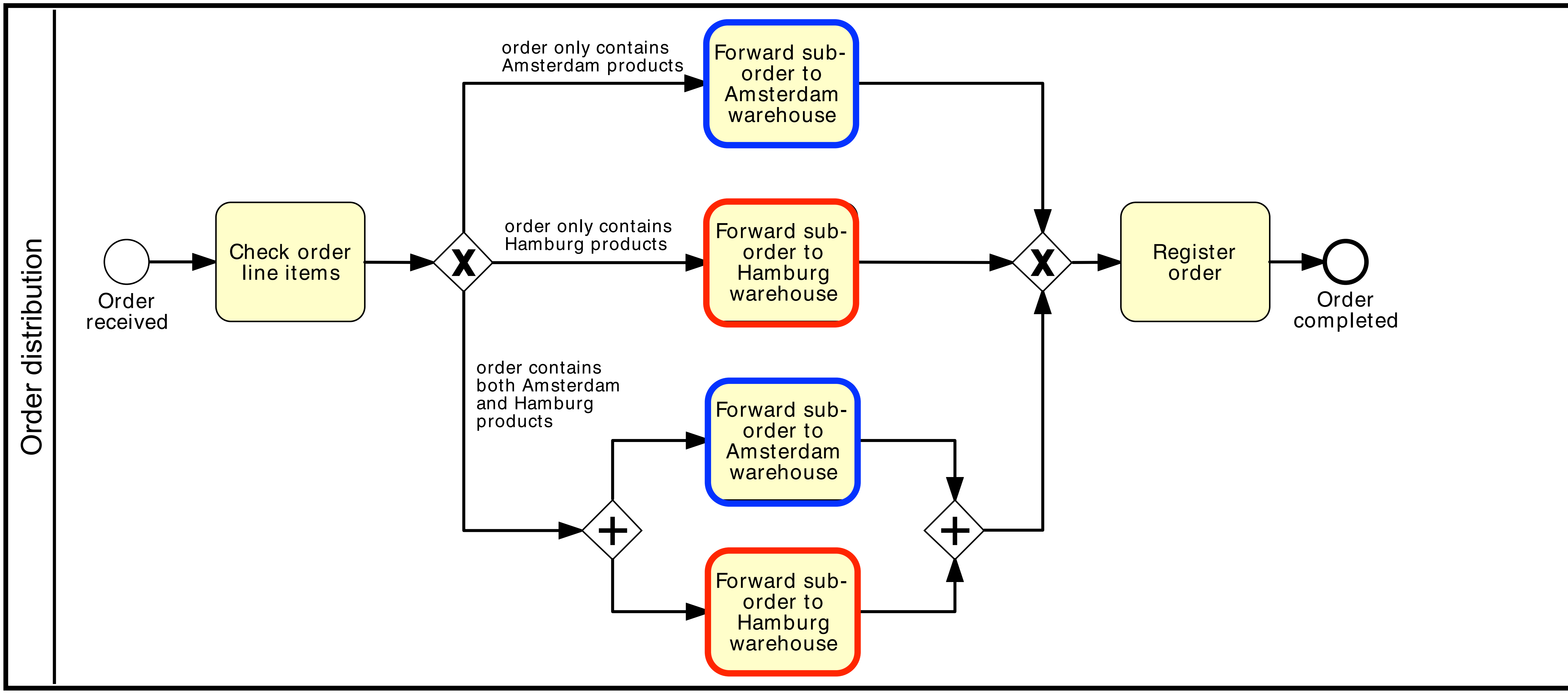
Conditional Flow has a condition assigned that defines whether or not the flow is used.

read as ``otherwise''



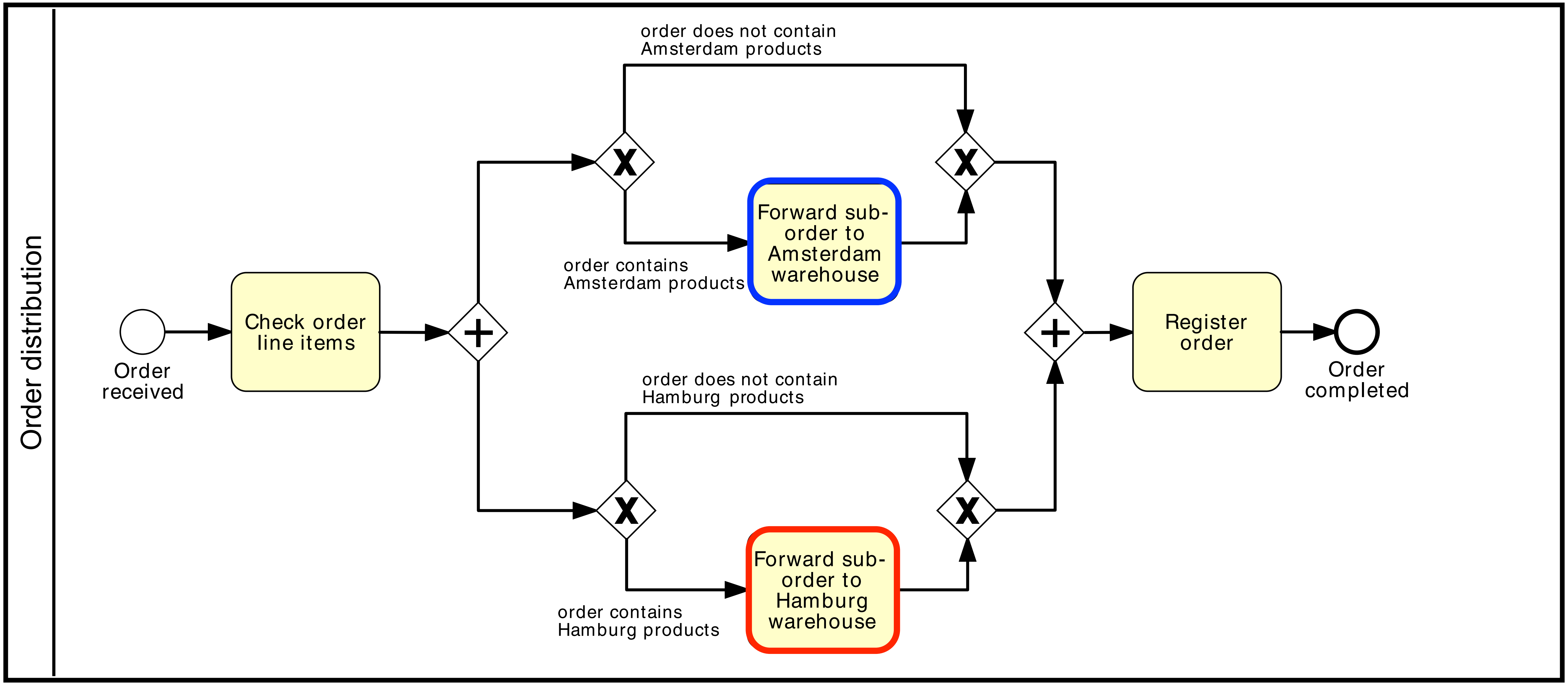
Default Flow is the default branch to be chosen if all other conditions evaluate to false.

Inclusive decisions: order distribution



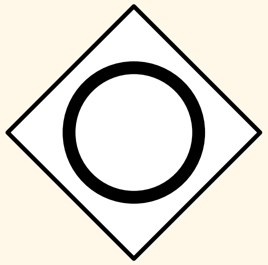
Only XOR / AND gateways, but the diagram is convoluted!
What if we had three or more warehouses? (does not scale)

Inclusive decisions: order distribution



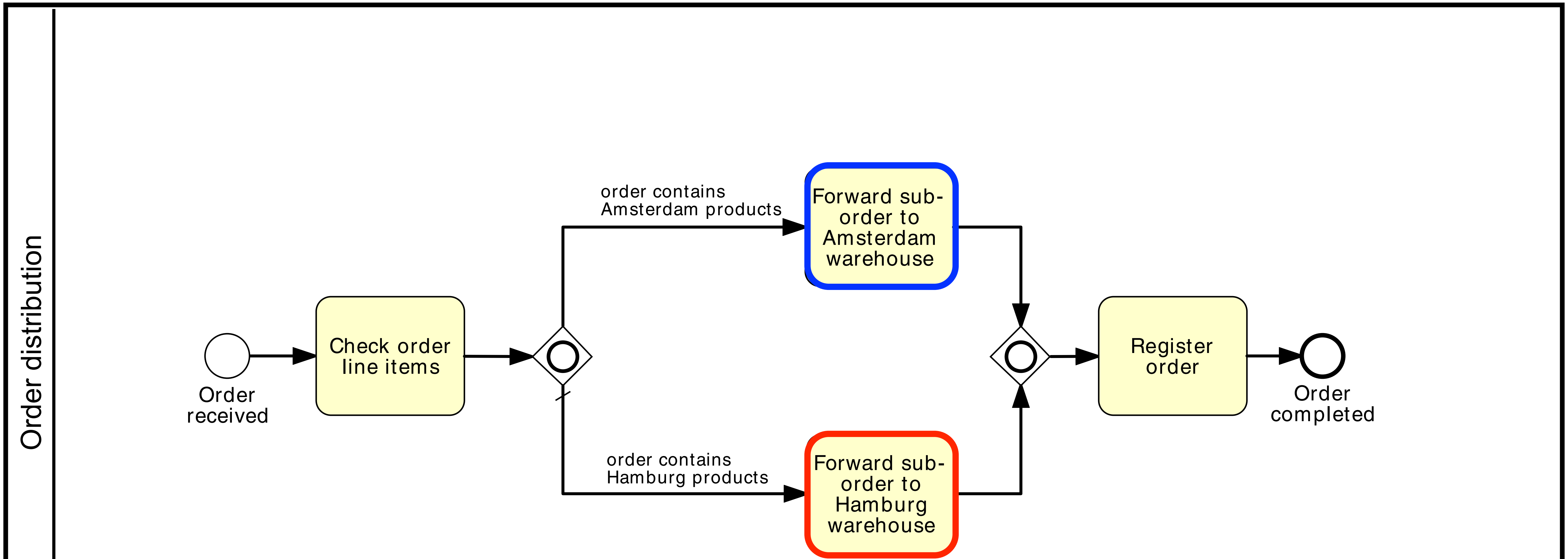
Only XOR / AND gateways, the diagram can ``scale'',
but is it correct? (also the case no-warehouse is now possible)

Inclusive decisions (one, many)



Inclusive Gateway
When splitting, one or more branches are activated based on branching conditions. When merging, it awaits all active incoming branches to complete.

Inclusive decisions: order distribution

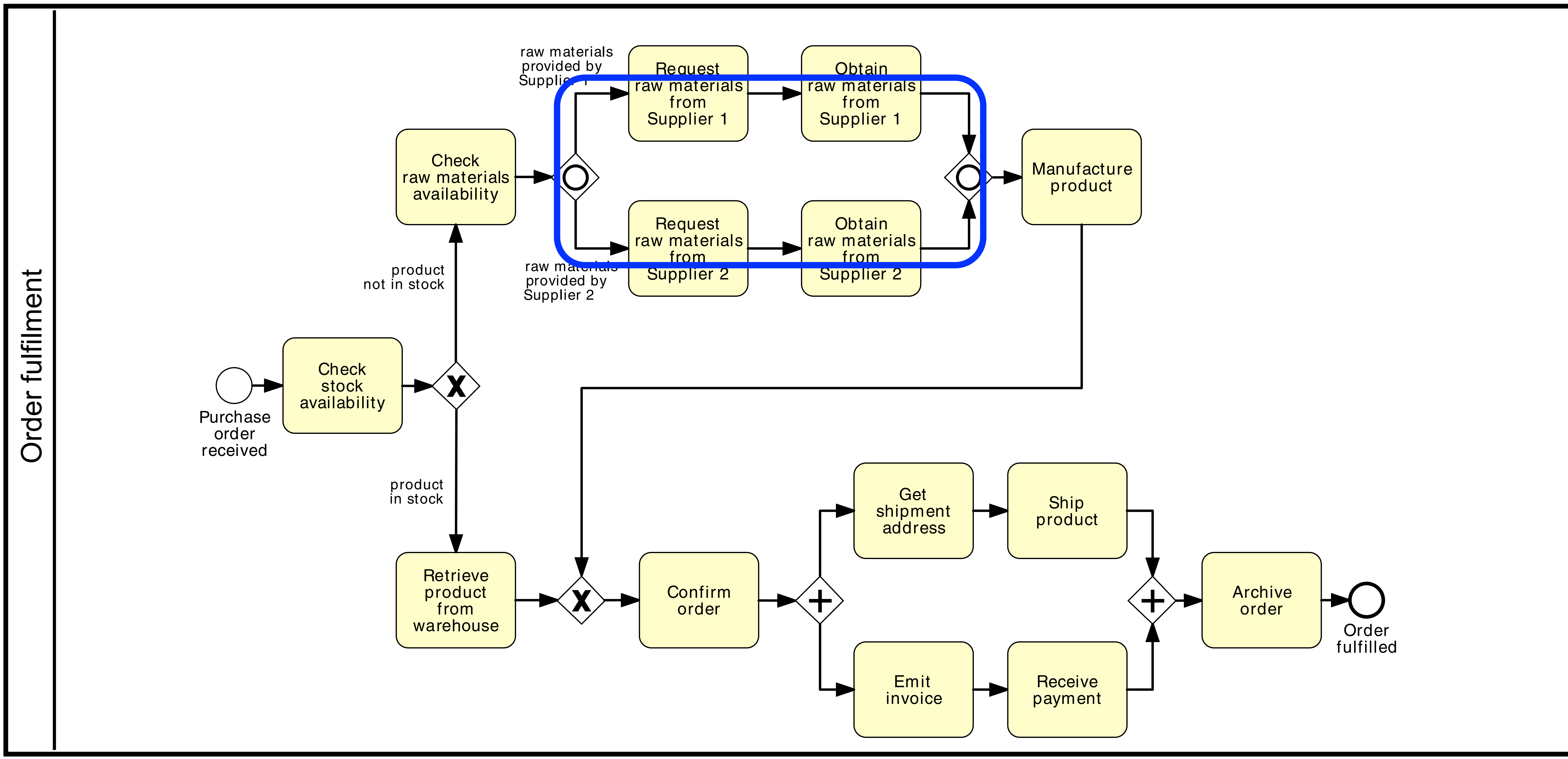


using OR gateways, the diagram can ``scale'',

but all the **issues with unmatched OR-joins in EPC** are still valid!

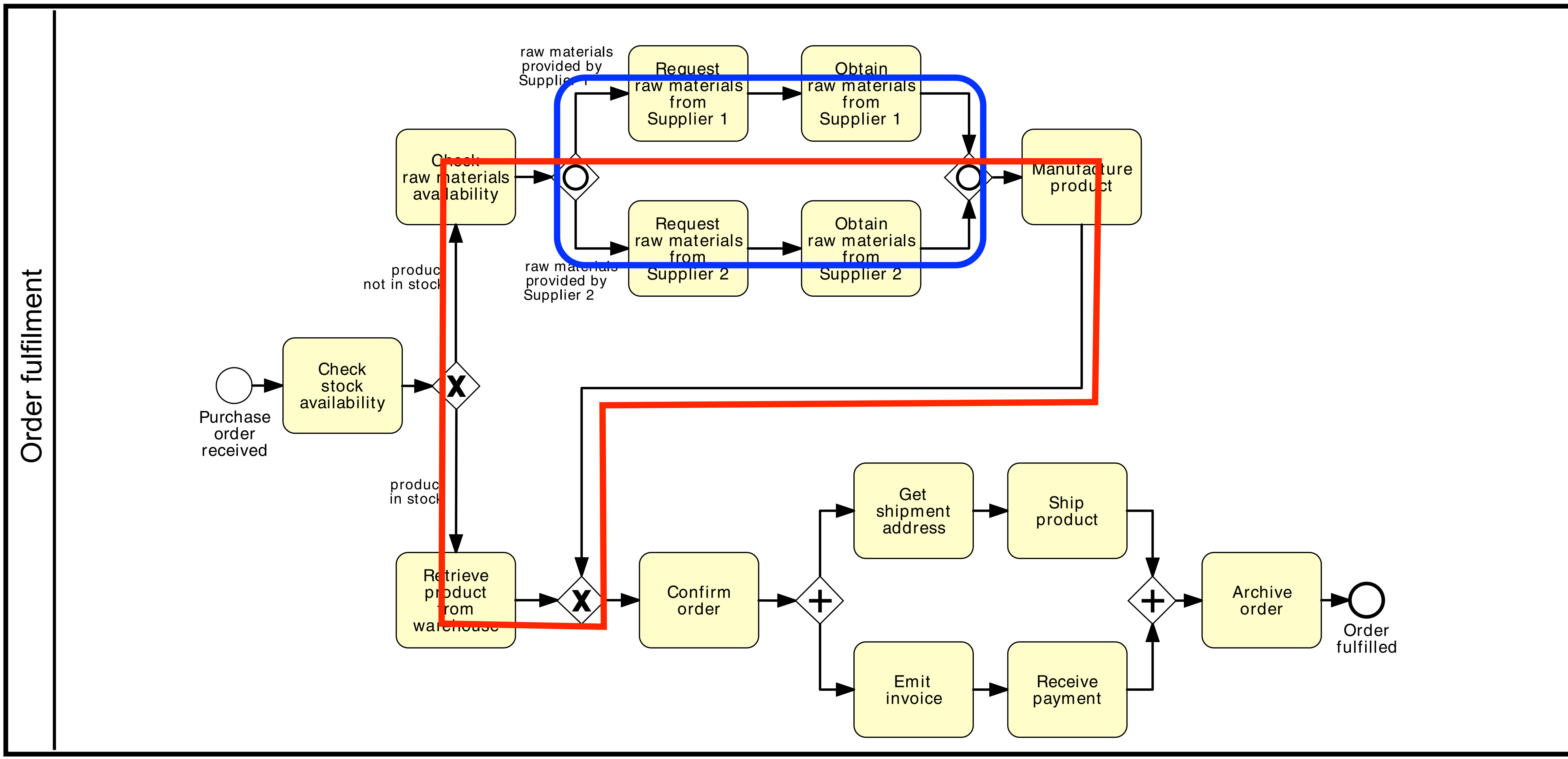
Use OR-gateways only when strictly necessary

XOR + AND + OR: order fulfilment



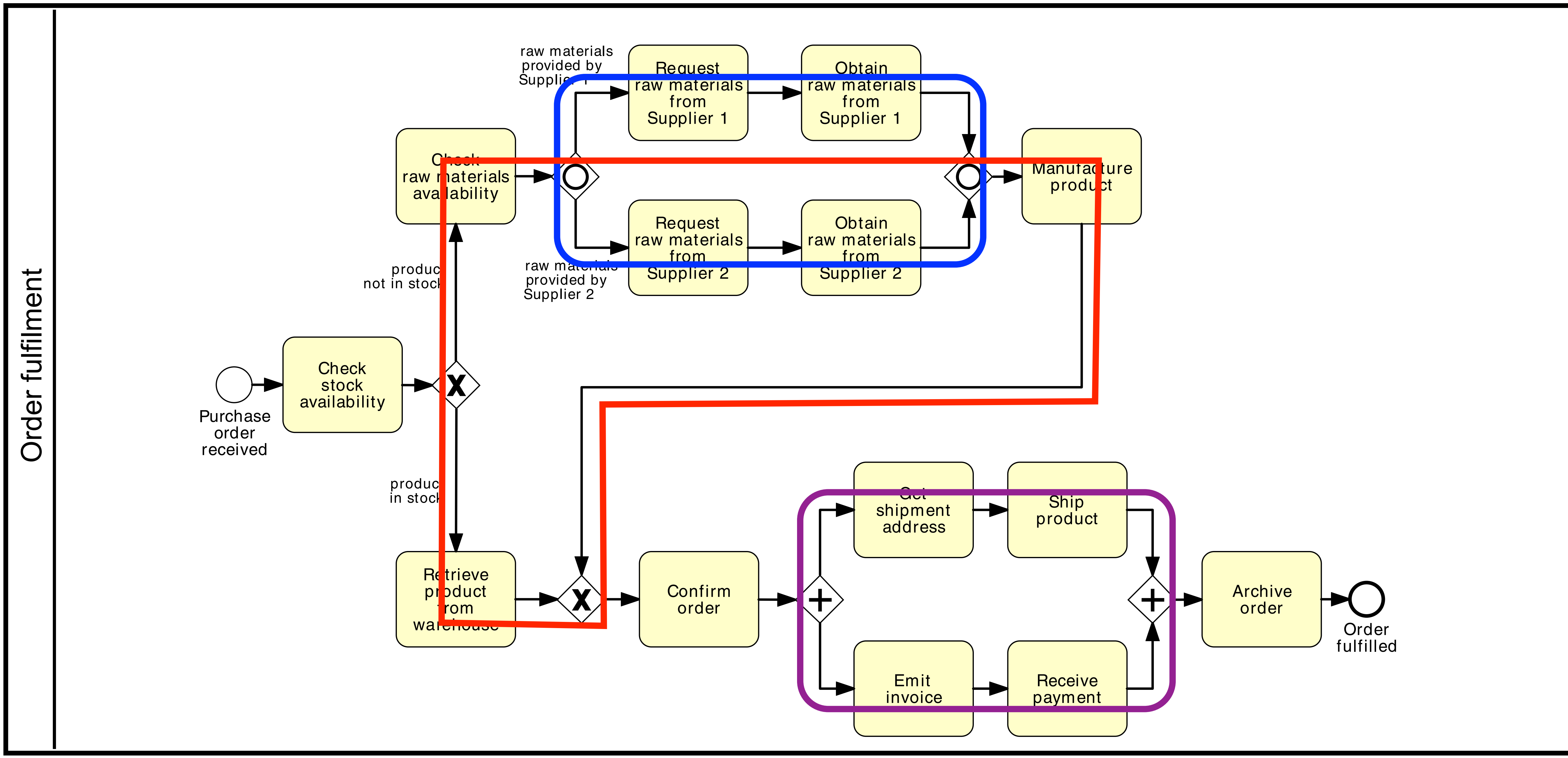
Better if gateways are balanced

XOR + AND + OR: order fulfillment



Better if gateways are balanced

XOR + AND + OR: order fulfillment



Better if gateways are balanced

Placing items in lanes

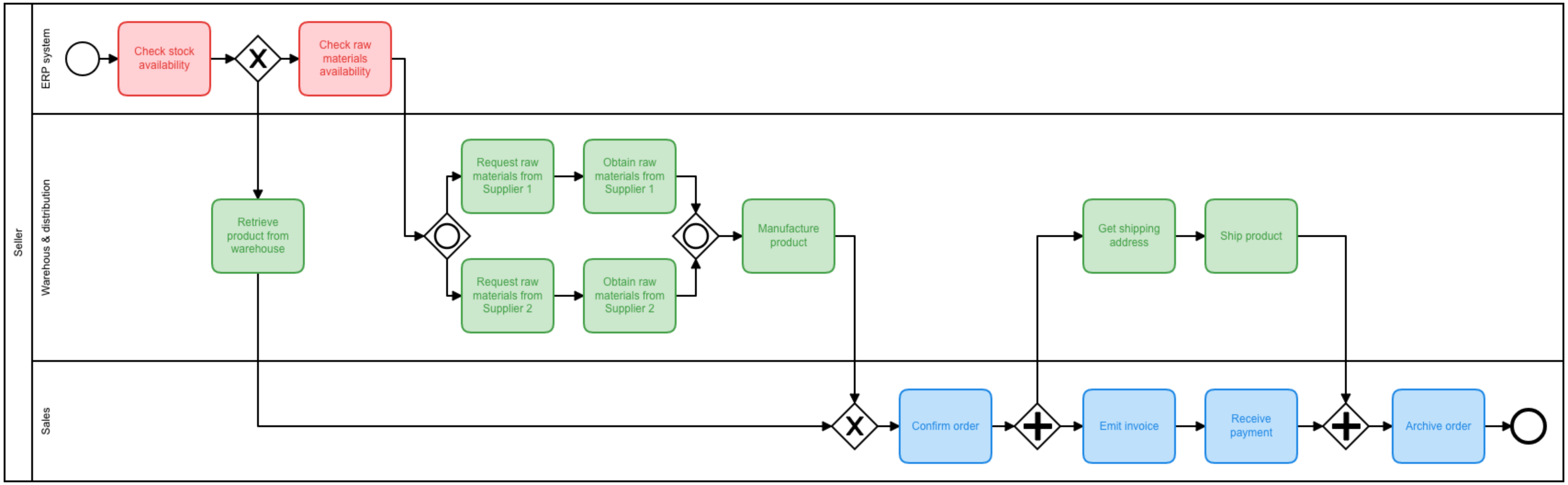
events: must be placed in the proper lane

activities: must be placed in the proper lane

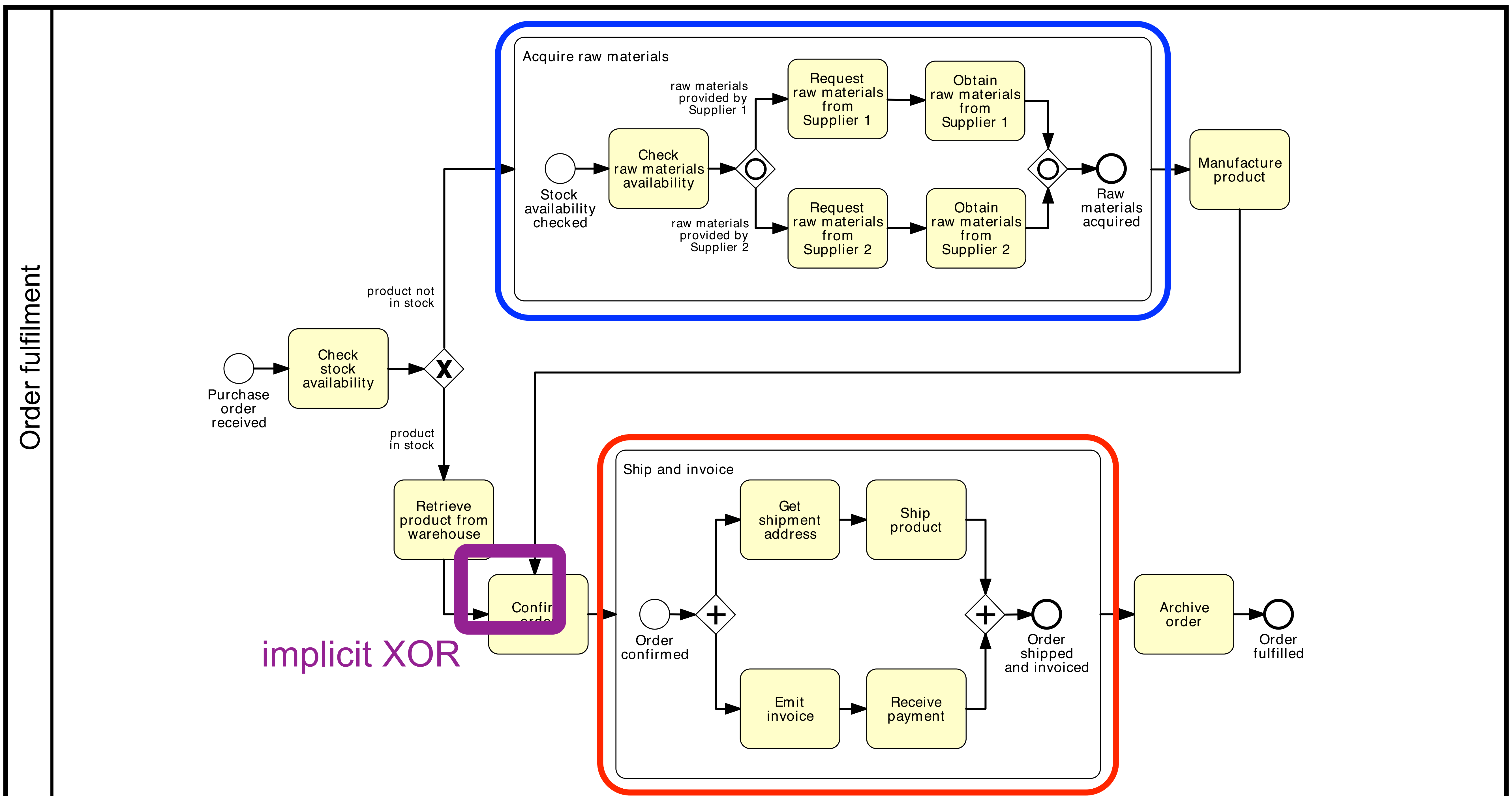
gateways:
(X)OR-splits: same lane as preceding decision activity
AND-split: placement is irrelevant
(any kind of) **join**: placement is irrelevant

data-objects: placement is irrelevant

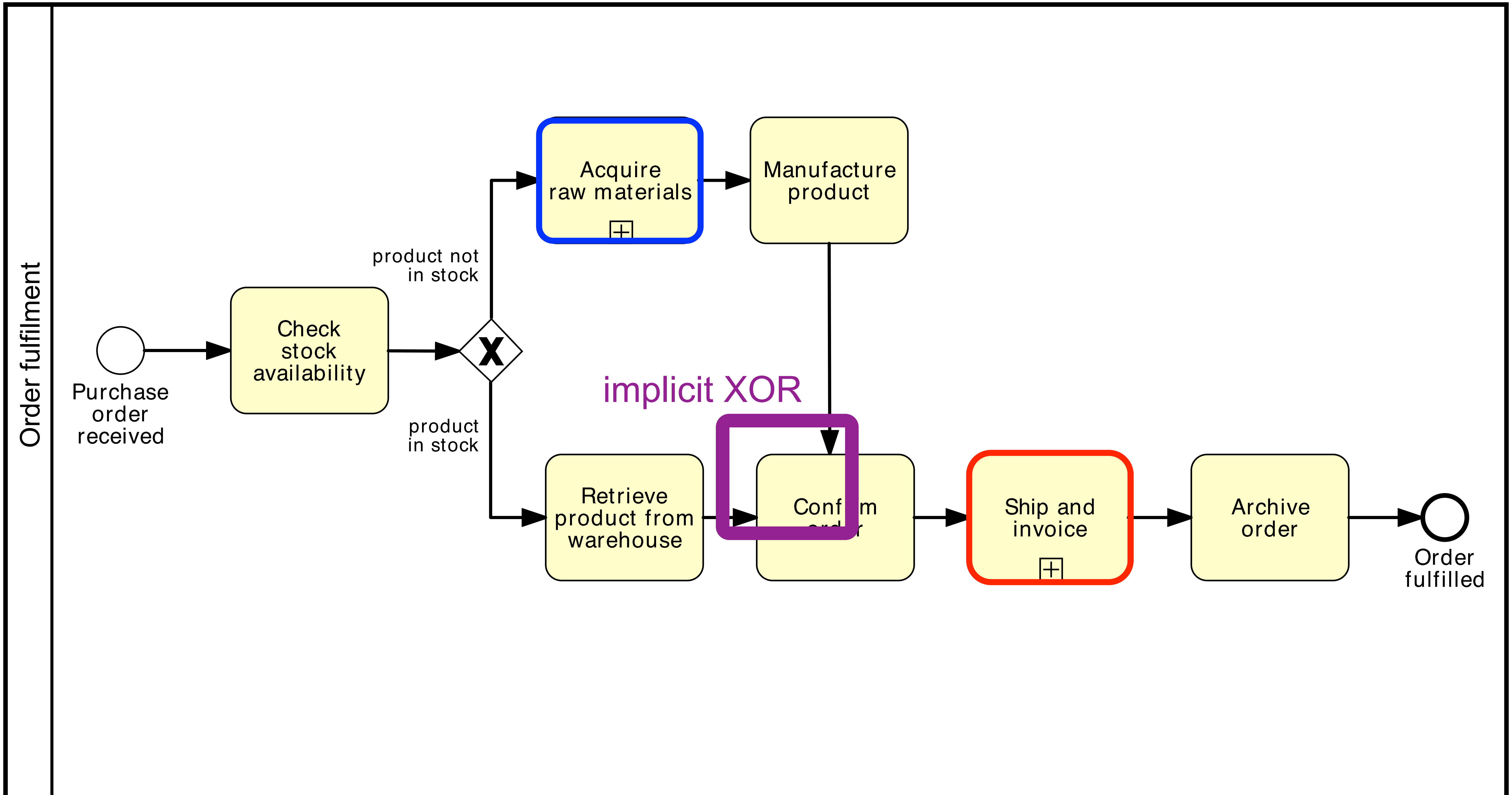
Resources as lanes: order fulfillment



Identify sub-processes

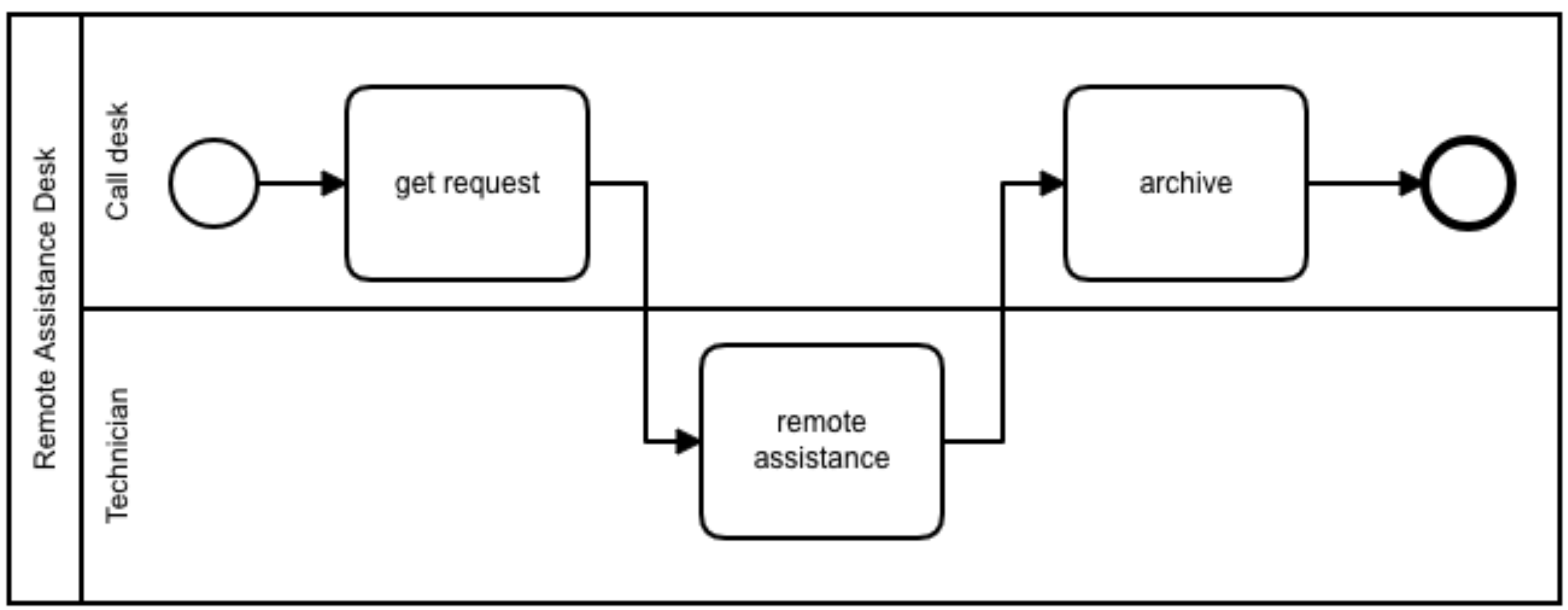


Collapsed sub-processes



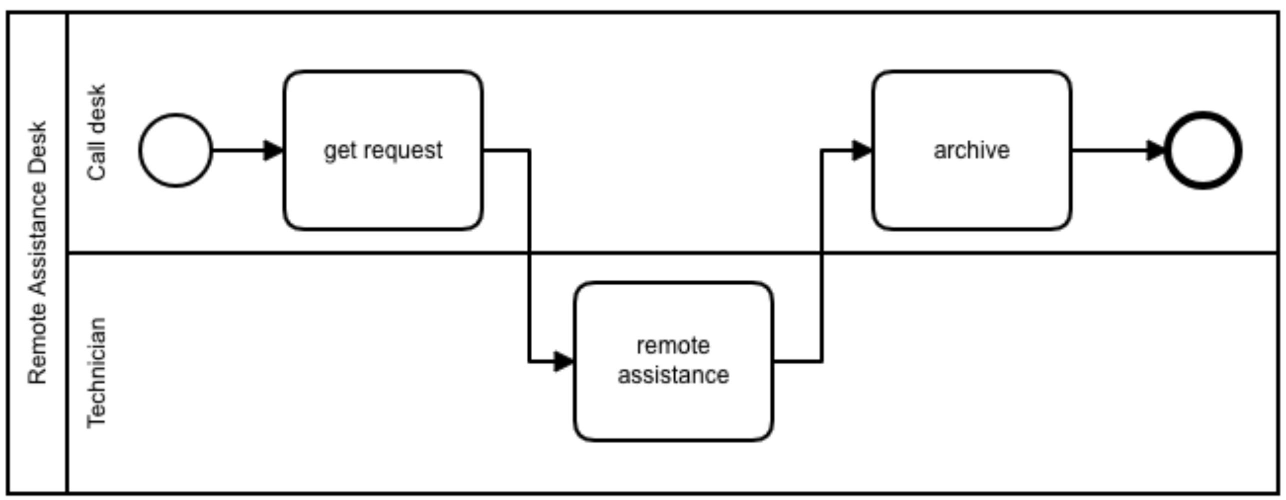
Exercise: basics

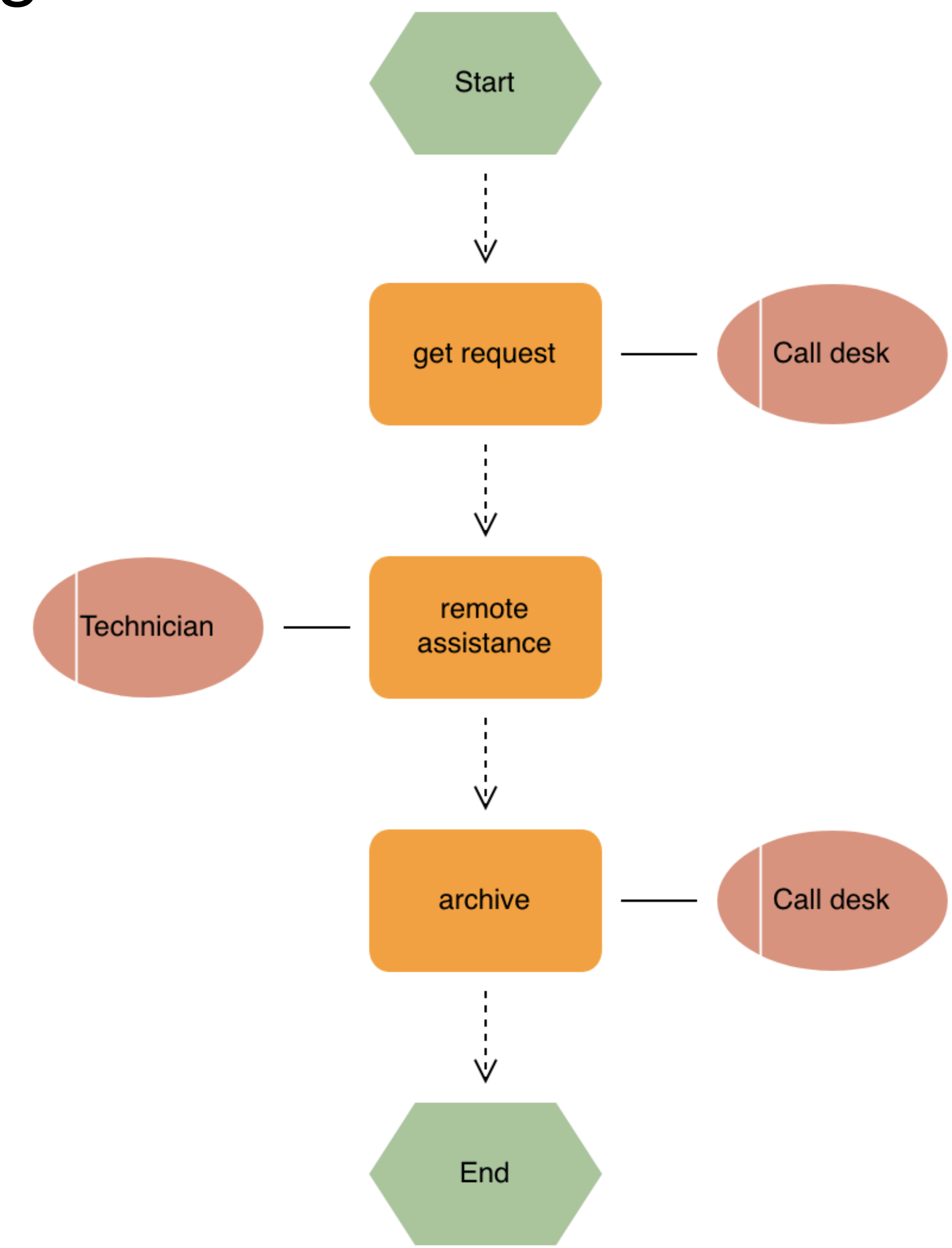
Draw the EPC diagram that corresponds to the BPMN diagram below



Exercise: basics

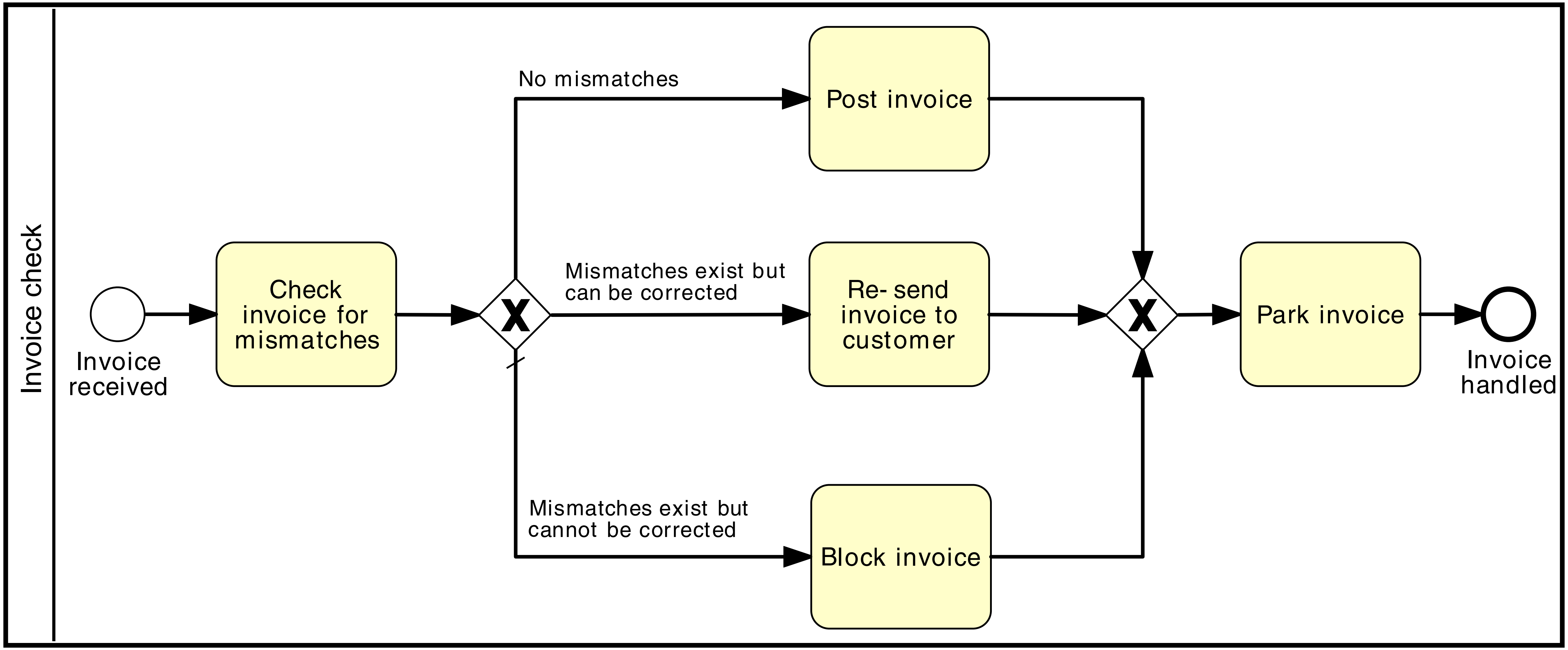
Draw the EPC diagram that corresponds to the BPMN diagram below





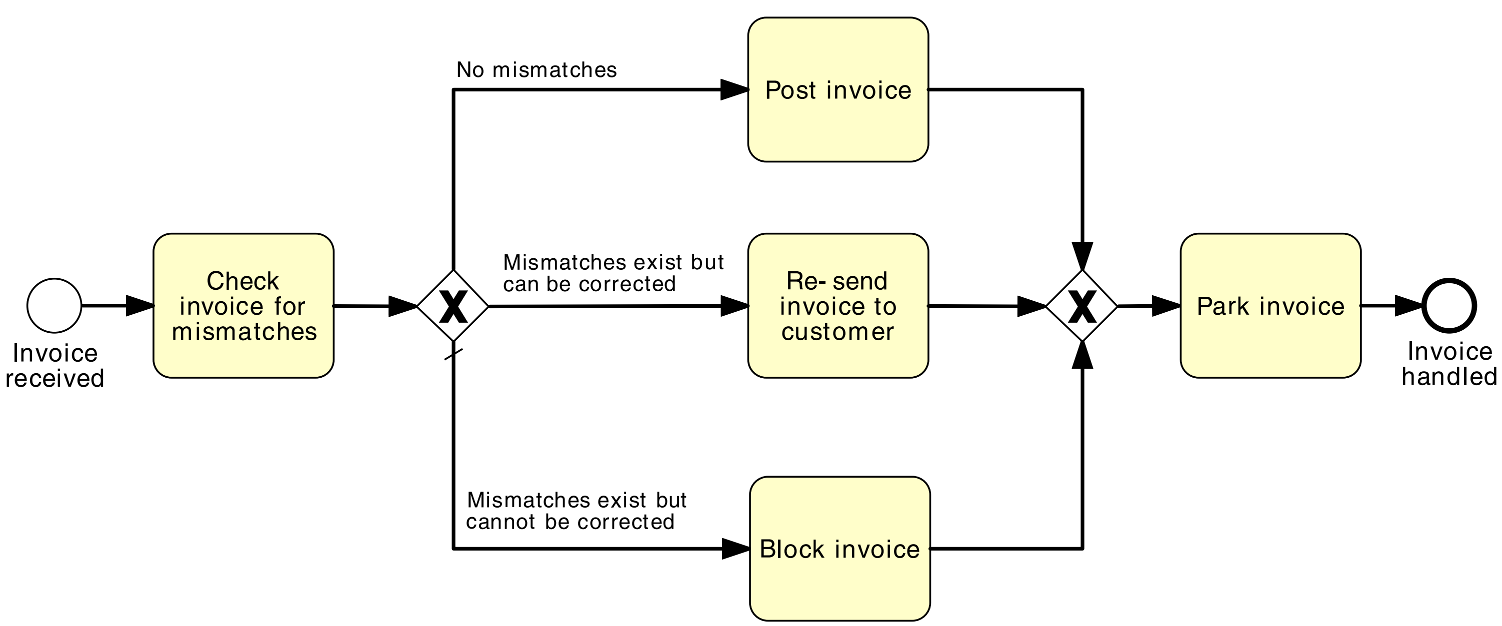
Exercise: basics

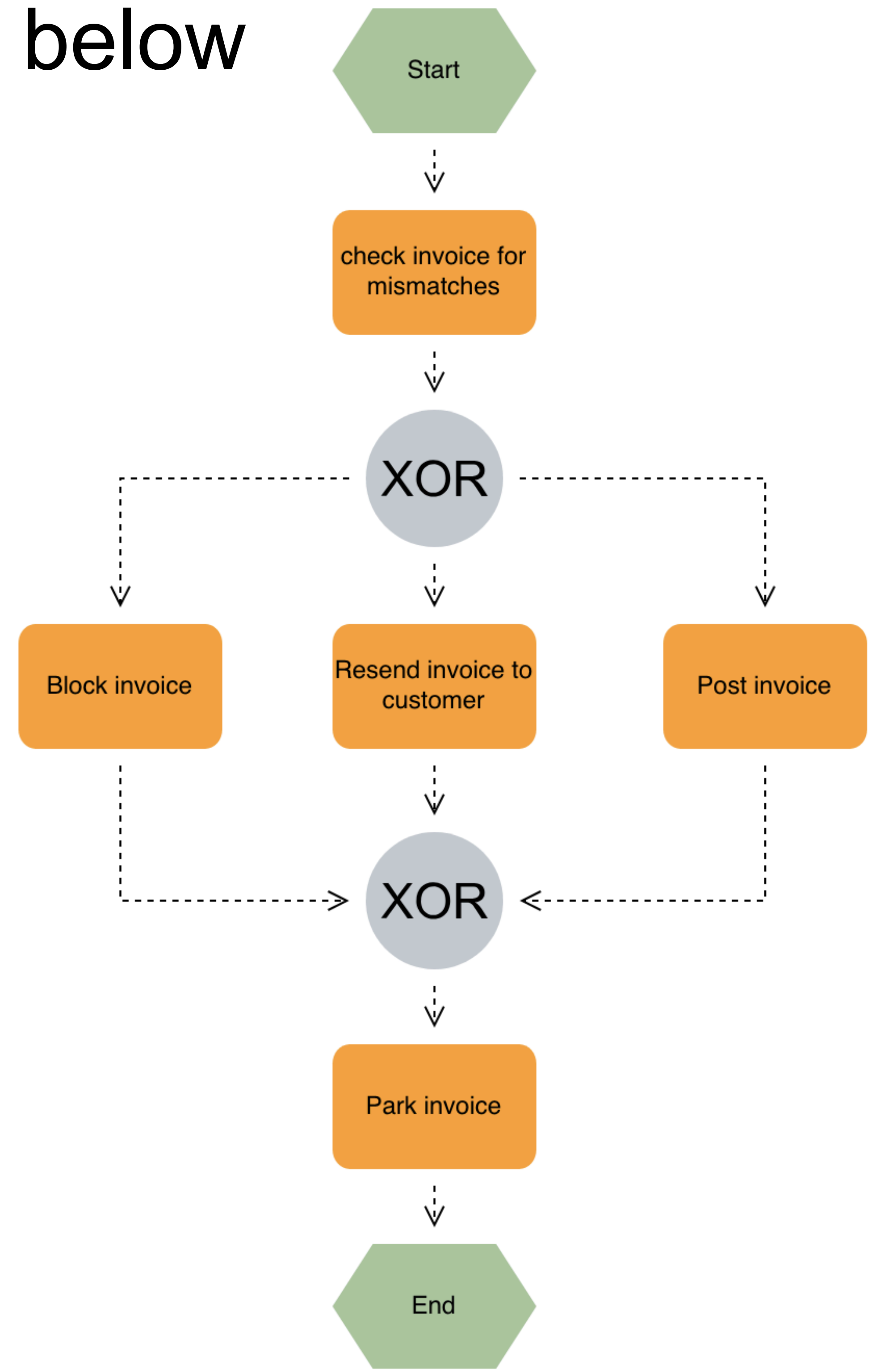
Draw the EPC diagram that corresponds to the BPMN diagram below



Exercise: basics

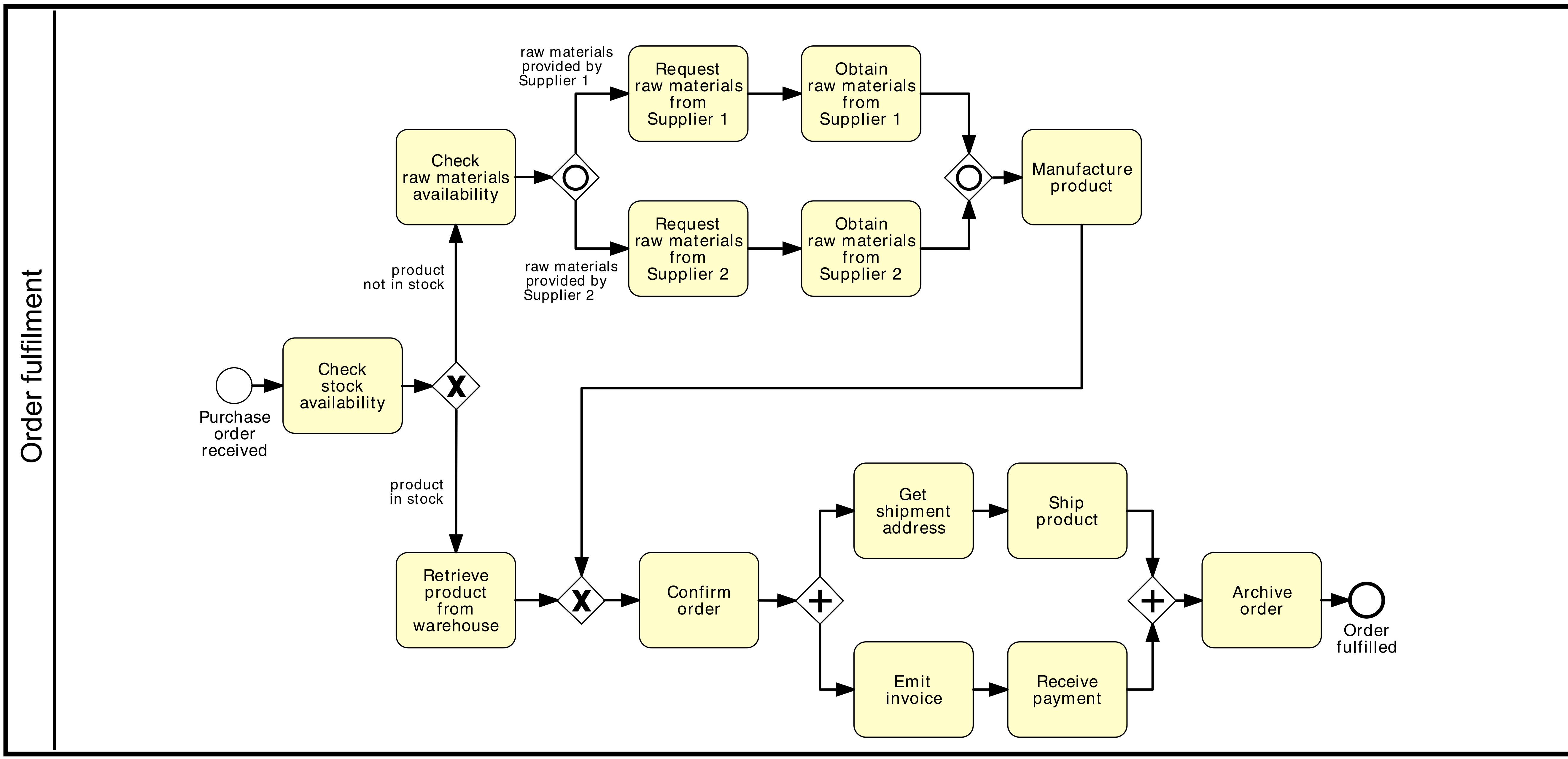
Draw the EPC diagram that corresponds to the BPMN diagram below





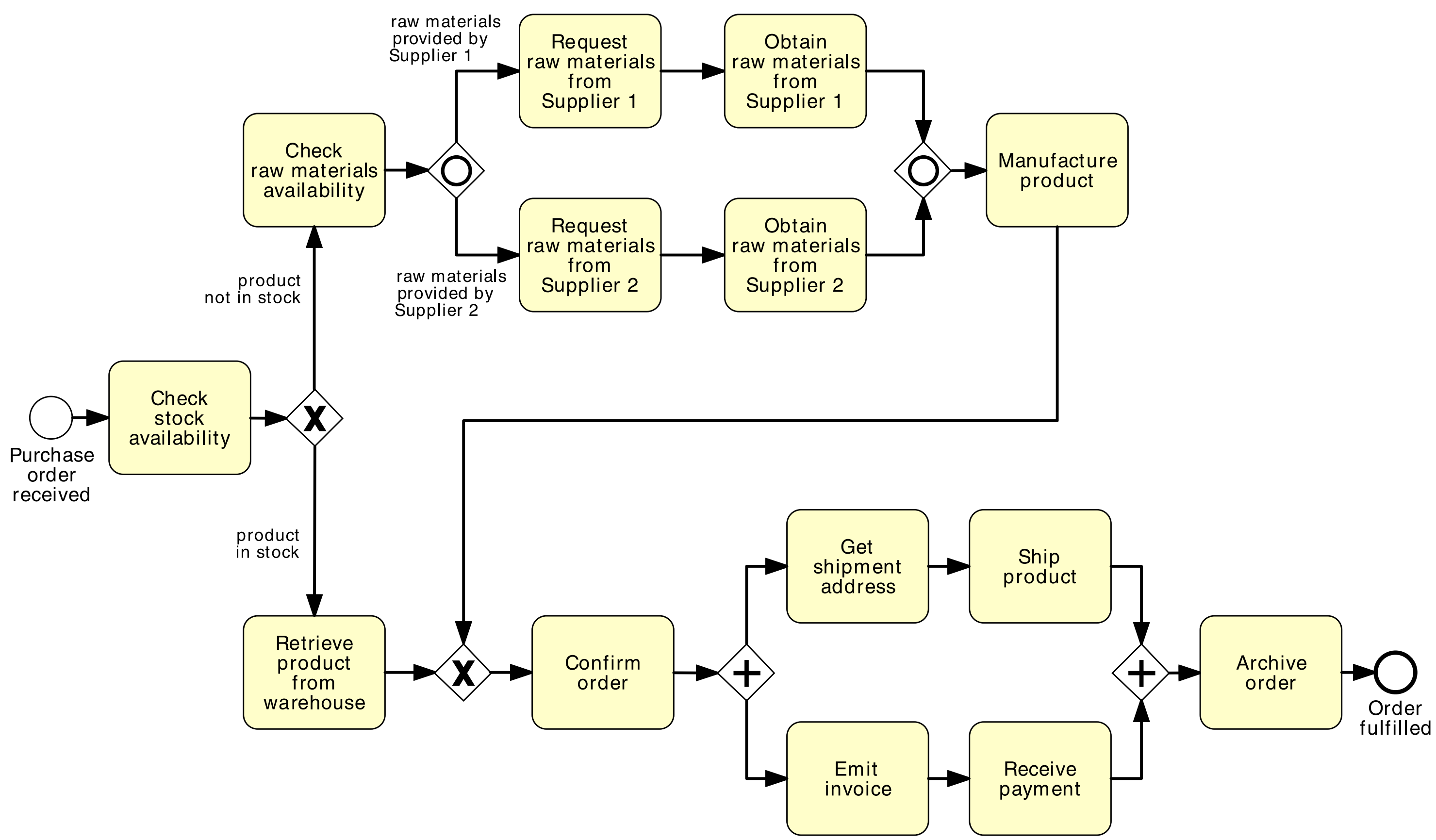
Exercise: basics

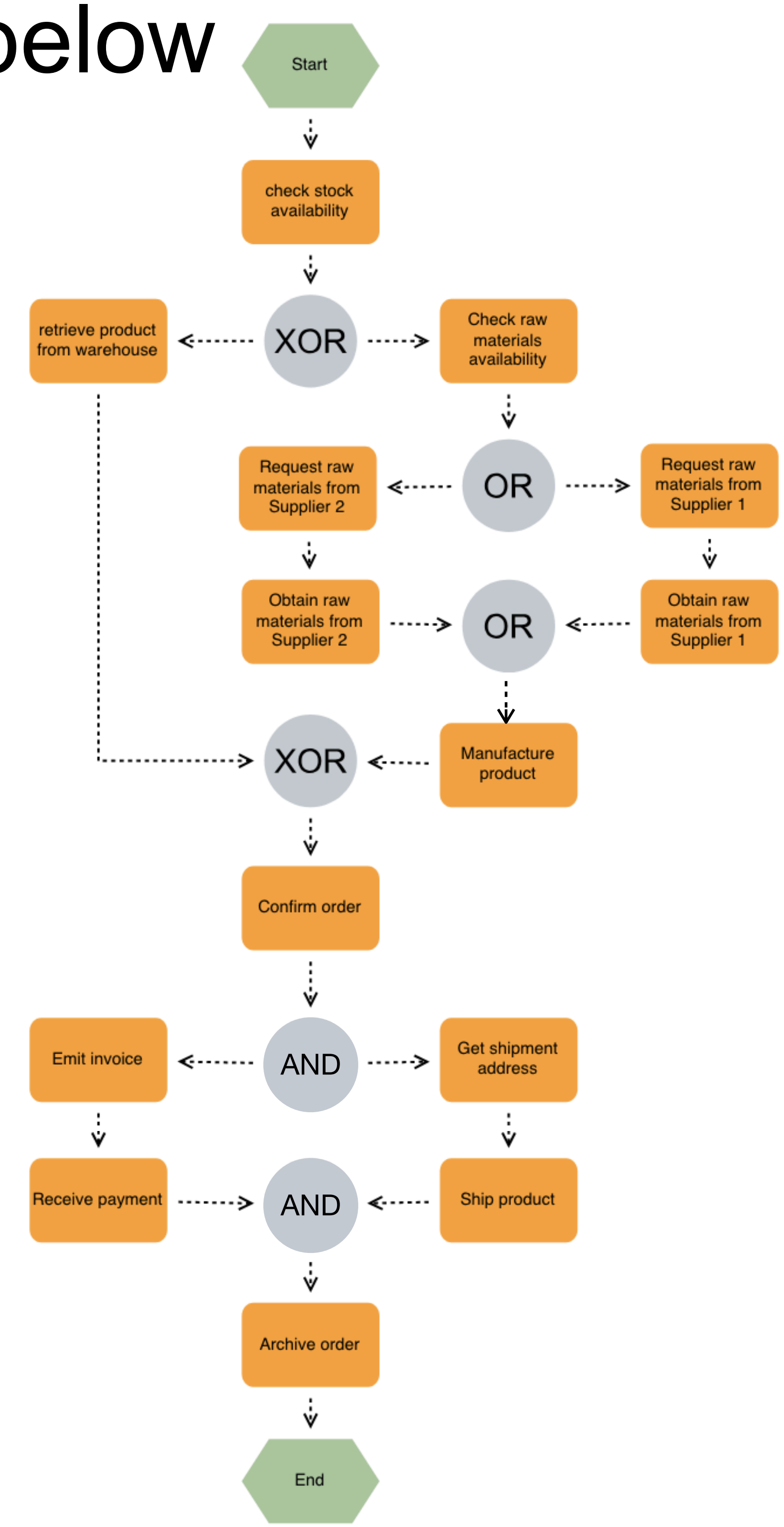
Draw the EPC diagram that corresponds to the BPMN diagram below



Exercise: basics

Draw the EPC diagram that corresponds to the BPMN diagram below





Exercises

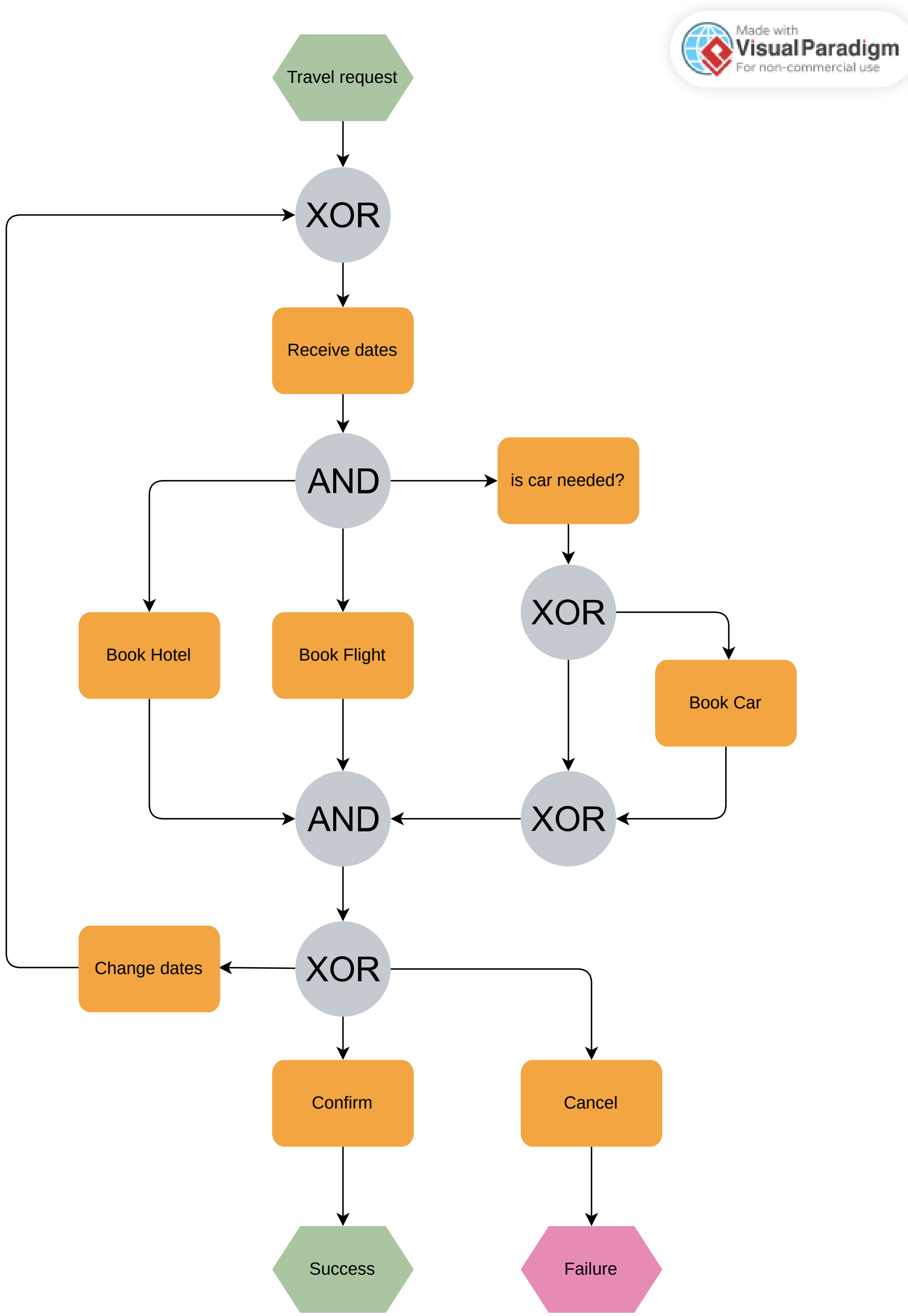
Transfer the following verbal description into an EPC diagram

You are tasked with modeling the Customer Order Process of a small e-commerce company.
The process starts when a customer places an order online and ends when the order is successfully delivered.
The process must involve at least the following activities:
checking if the items are available in stock,
a notification to the customer if the items are not available,
the preparation of the order for shipment,
and the processing of the payment.

Send your solutions to: `bruni@di.unipi.it`

Exercises

Draw the BPMN diagram that corresponds to the EPC diagram below



Send your solutions to:

bruni@di.unipi.it