Business Process Modeling with BPMN 2.0

2nd edition

gregor.polancic@gmail.com

About the Author

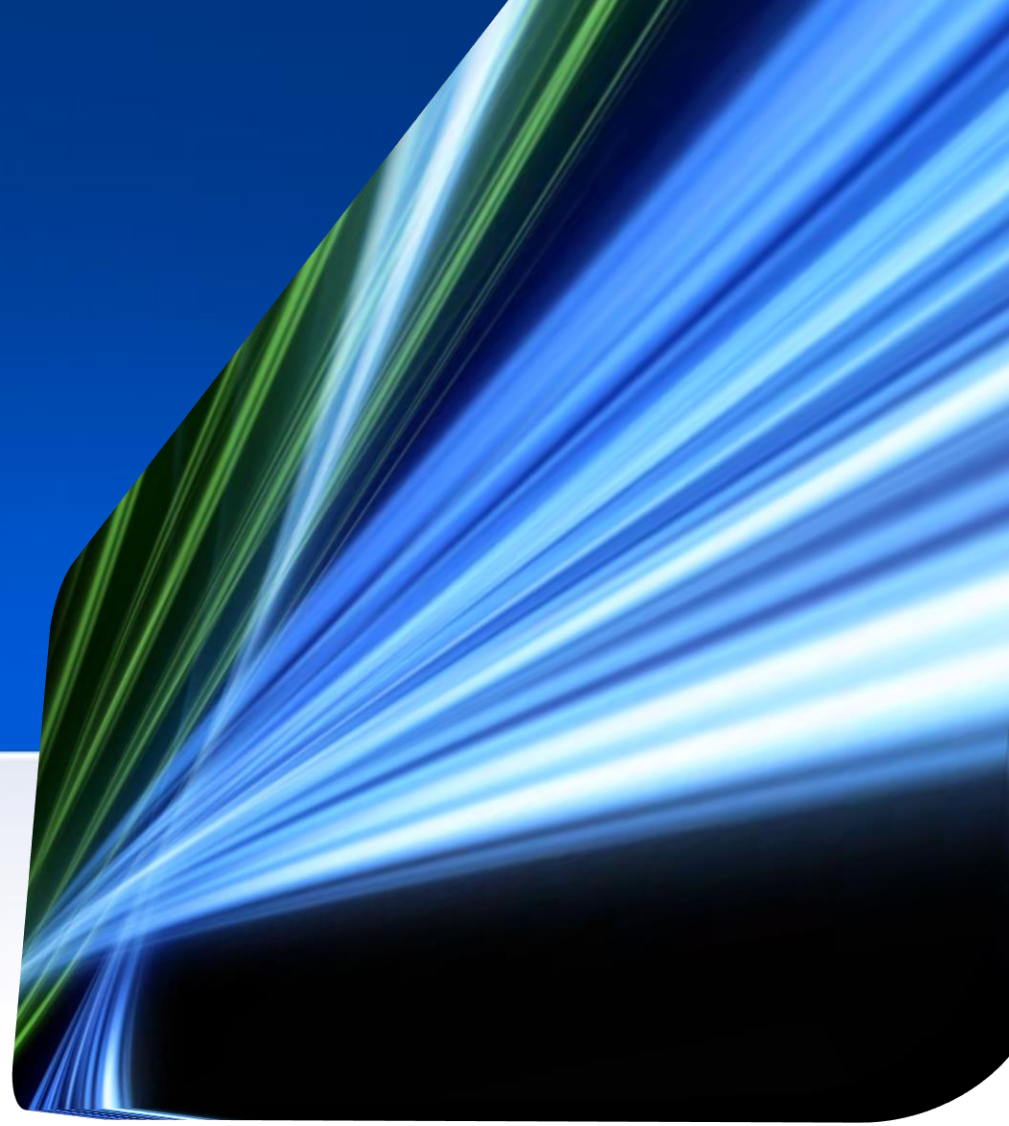
Gregor Polančič is an assistant professor at the University of Maribor, Slovenia. He received his PhD in 2008 in the fields of software engineering and information systems. He has a decade of experience in BPMN, starting to investigate and actively use BPMN since its first version in 2004. He has participated in the development of one of the first BPMN modeling utilities - a package of plugins for Visio, which were introduced in 2004 and is the main author of the first BPMN poster, which has been translated into several languages and already exceeded 50.000 downloads. In 2008, he was one of the first authors who published an article, dedicated to the experiences and practical use of BPMN. The article was published in 'BPM and Workflow Handbook' in association with the Workflow Management Coalition (WfMC). Gregor Polančič is teaching BPMN and BPM since 2005 in several undergraduate and postgraduate courses and has been an invited BPMN lecturer on several Universities. He has participated in several BPMN related local and international workshops and projects and is currently researching BPMN from different technological and user aspects. Gregor Polančič is also owner of the international certificate Certified Business Process Professional for Higher Education Institutions. Gregor Polančič is periodically authoring for Orbus Software and blogging as well preparing learning materials for Good e-learning.

http://www.linkedin.com/in/gregorpolancic

gregor.polancic@gmail.com

Content

- The basics of business processes and BPMN
- Basic set of process modeling elements
- Full set of process modeling elements
- BPMN beyond process modeling



Business Process Modeling with BPMN 2.0

THE BASICS OF BUSINESS PROCESSES AND BPMN

What is a Business Process?

- A business process consist of a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a business goal (product or service).

Business process

Realizes

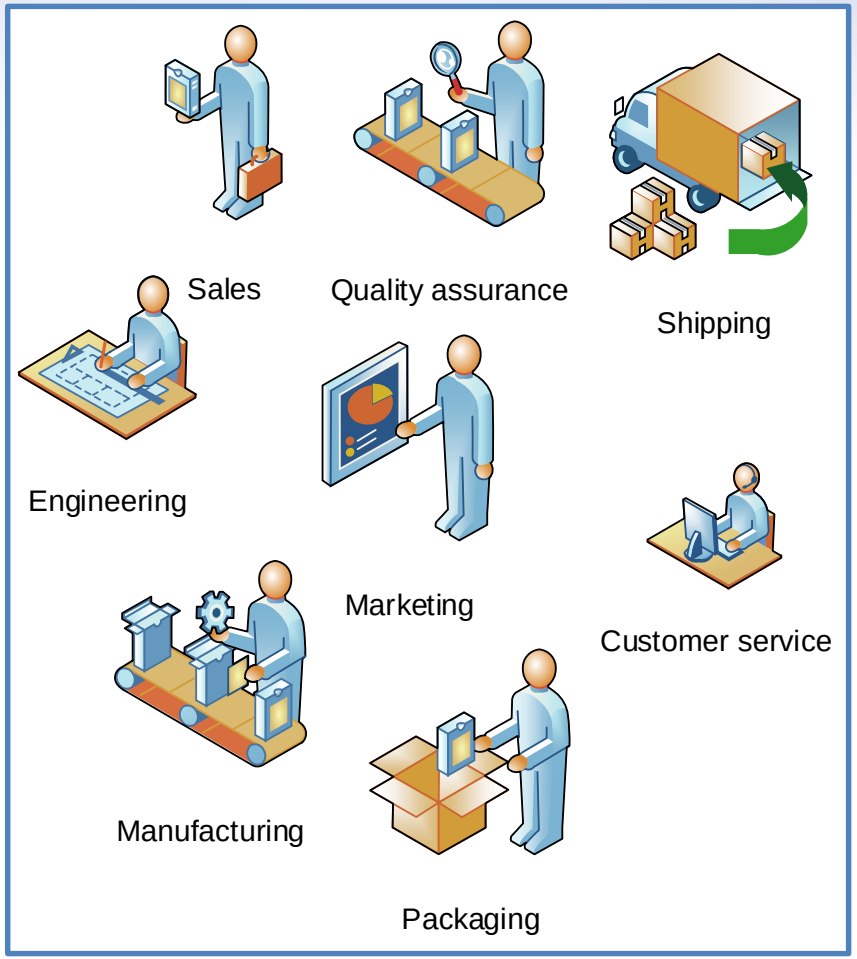
Product or service

- Each business process is enacted by a single organization, but it may interact with business processes performed by other organizations.

(Weske 2012)

Should you Care about Business Processes?

- Yes, because each organization produces something or services somebody.
- So, each company runs processes, however only some companies are aware of them. Don't agree? Try to find a counterexample!
- Did you know that business processes are organizational assets that are central to creating value for customers?



Examples of business processes

What Kind of Assets are Business Processes?

Tangible assets	Intangible assets
 Material, Machines, Infrastructure, Money,…	 Knowledge, Relationships, Processes, Information,…
These are organization's resources.	These are organization's capabilities.
Resources can be acquired.	Capabilities CANNOT be acquired. They have to be evolved.

- The product or service of an organization depends mainly on how capabilities use (develop, deploy and coordinate) resources.

(Orand 2011)

Are you Aware of Business Processes?

Process AWARE company

- A simple approach to evolve (i.e. improve) processes:
  1. If you are aware of processes you can identify them.
  2. If you can identify processes you can analyze them.
  3. If you can analyze processes, you can evolve (i.e. improve) them.

Deming (PDCA) circle

Process UNAWARE company

- Any ideas how to evolve processes?

?

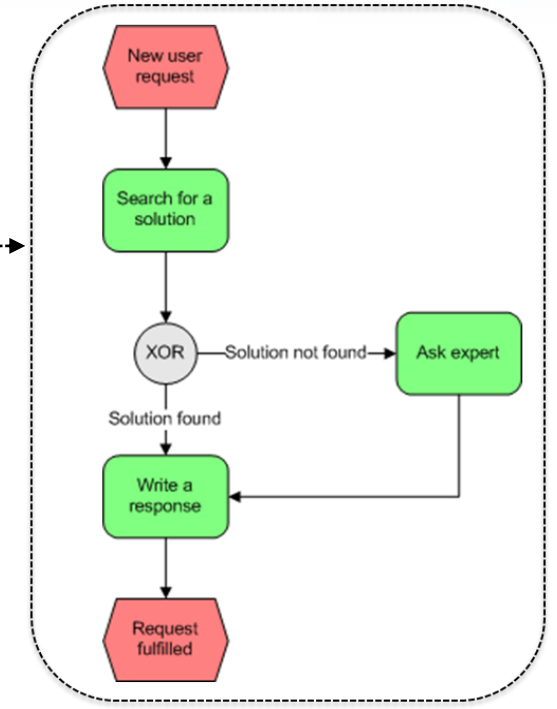
How Can we Work with Processes? With Modeling.

- Can we ‘touch’ a process? No, it actually exists only in our minds.
- So, we have to indirectly work with a process.
- The best way of working with a process is to model it.

Help desk center in a real world

Modeling...

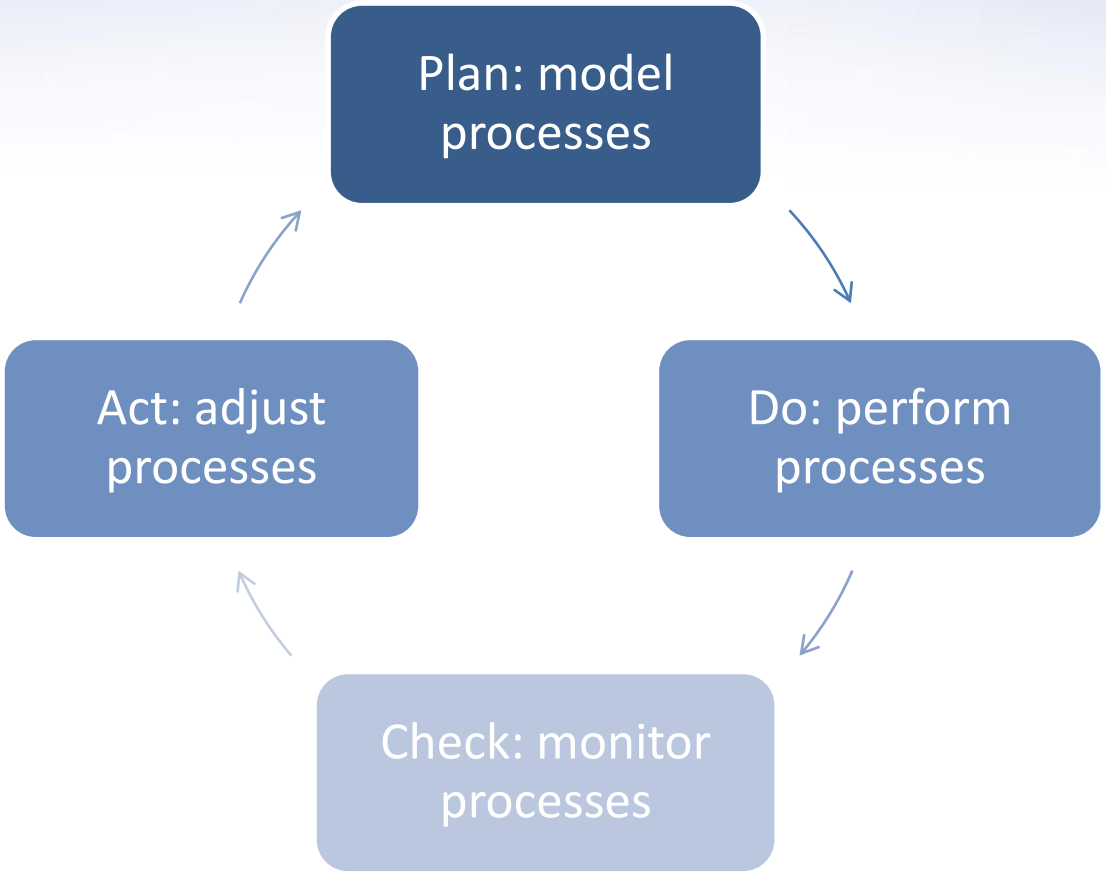
Transformation of a real-world business process into a process model.



Help desk process model

Benefits of Process Modeling

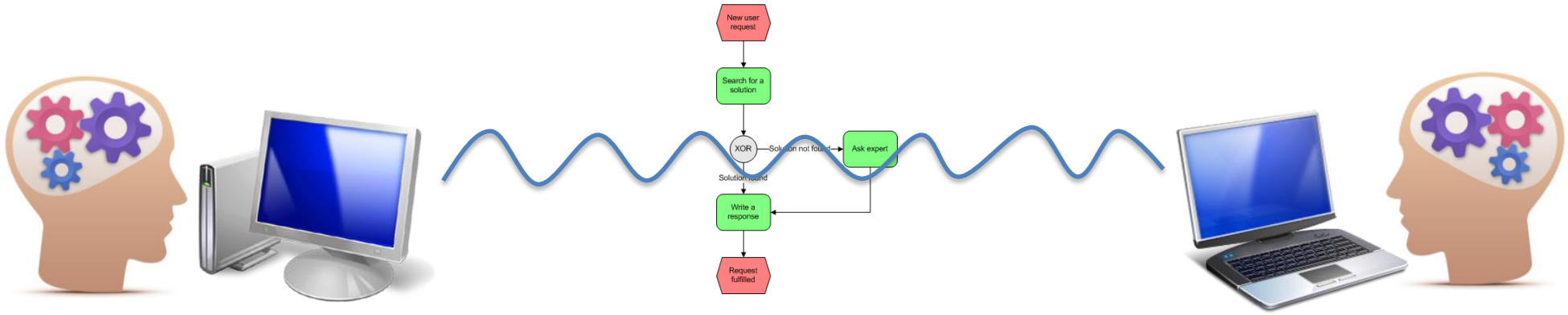
- Representation of processes.
- Analysis of processes.
- Continuous improvement of processes.
- A basis for IT support.
- A solid basis for process management.



Deming (PDCA) circle

Which „Modeling Language“ Should be Used?

- Process models are commonly used for communication between co-workers, customers or business partners.
- Besides, processes are commonly modeled, performed on computers in a collaborative environment.
- These requires a common - standardized modeling language.



Computers and humans can effectively communicate and collaborate in case of a common (process modeling) language.

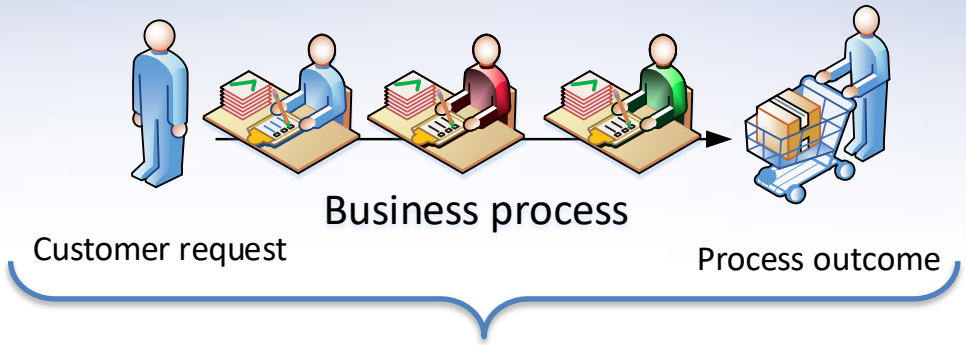
What We Have Learned so Far …

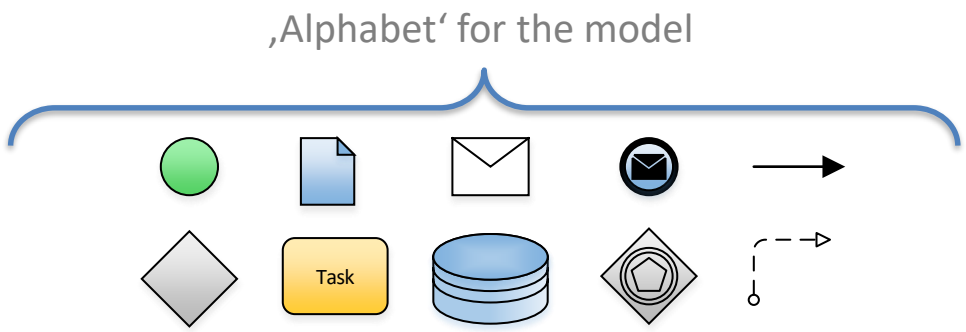
- Each company runs business processes.
- Business processes are organizational assets that are central to creating value for customers.
- In order to stay competitive, a company needs to continuously improve the processes.
- Processes are intangible, so we cannot manage them directly. An effective way to work with processes is to model them.
- To take the full potential out of process modeling, a standardized process modeling language should be used. This enables interoperability, communication and collaboration.

$$\sum = \text{BPMN}$$

What is BPMN?

- Business Process Model and Notation
  - Business Process - A collection of related, structured activities or tasks that produce a specific service or product for a particular customer.
  - Model –a representation of a business process.
    - Visual proces model – process diagram
    - Non-visual proces model (e.g. executable process model)
  - Notation – a set of elements (language) + rules used for representing a business process in a business process model (diagram).

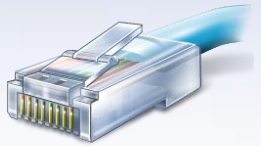




The current BPMN version is 2.0.2, released in January 2014.

Why BPMN?

- Standardized. The de-facto & ISO/IEC 19510:2013 standard in process modeling.
- Open. Created and controlled in an open and fair process.
- Simple & complete. Can be used in a simple or detailed way.
- Learnable. Based on previous notations.
- Interchangeable. Capable of being interchanged between IT solutions.
- Executable. Capable of being automated.



Standardization enables communication.

Openness enables transparency and democracy.

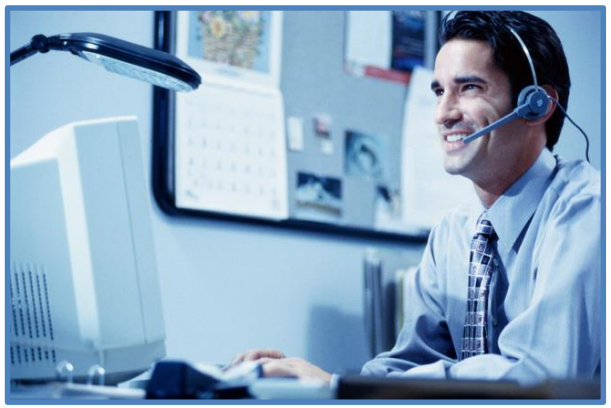
Simplicity and learnability enable acceptance and popularity.

Interoperability enables humans and IT collaboration.

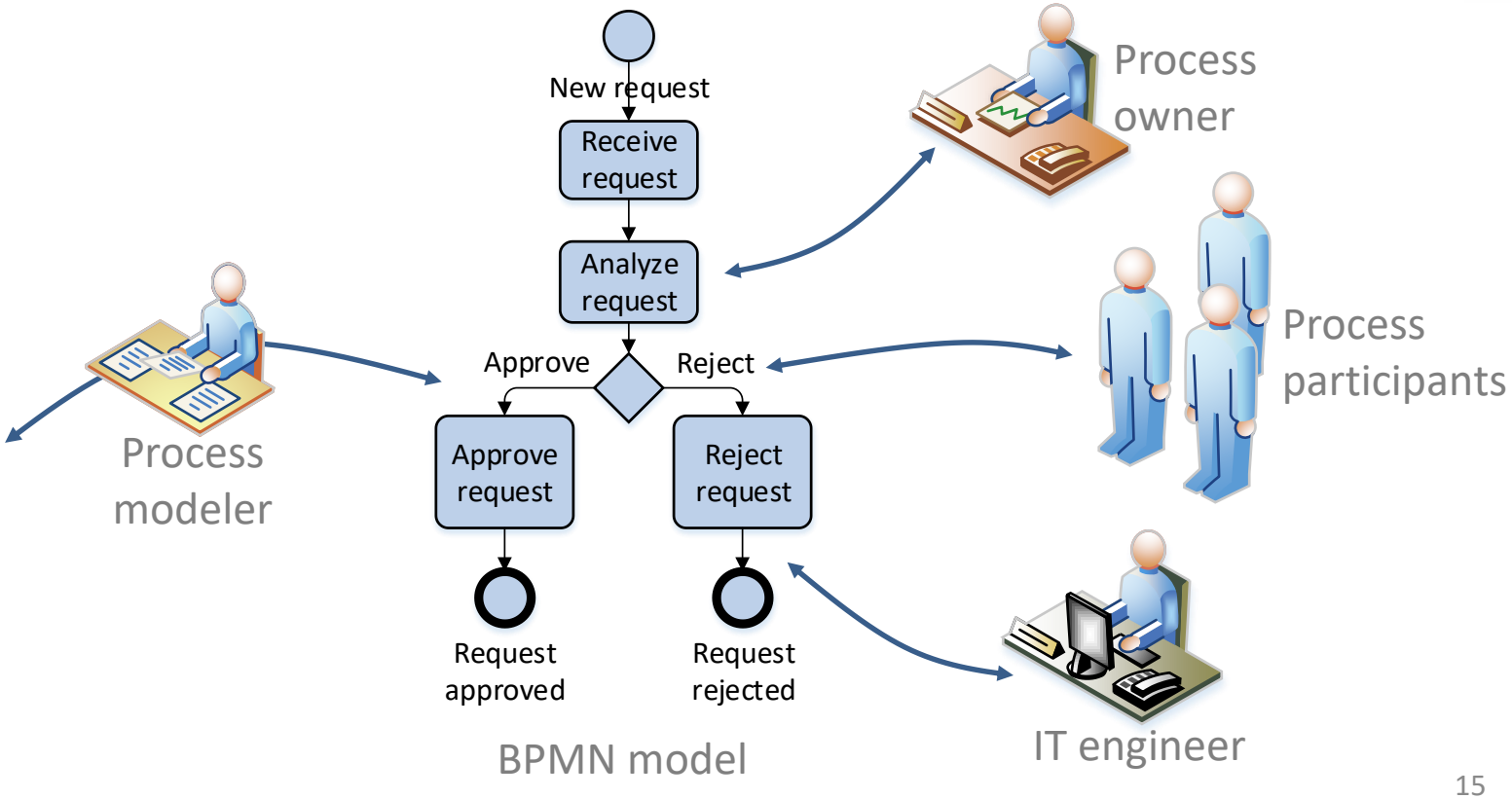
Automation improves process's efficiency and consistency.

Primary Goal of BPMN

- "The primary goal of BPMN is to provide a notation that is readily understandable by all business users /…/. Thus, BPMN creates a standardized bridge for the gap between the business process design and process implementation."



Real-world business process



In Scope of BPMN

BPMN 2.0
Page 1-12

Business process modeling

- Diagrams (e.g. process diagrams, collaboration diagrams).
- Syntax, semantics and visual appearance for process elements (e.g. events, activities and gateways).
- Attributes and properties of the semantic elements represented by the graphical process elements.
- Formats for exchanging diagrams.

Business process execution

- Execution semantics.
- Formats for exchanging executable models.
- Support for BPMN and BPEL process engines.

Focus of these slides

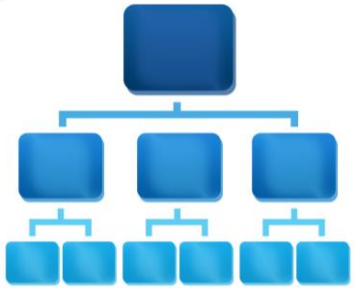
Introduced in these slides

Out-of-Scope of BPMN

- BPMN is constrained to support only the concepts of modeling that are applicable to business processes.
- Out of scope, but related domains are:
  - Definition of organizational models and resources.
  - Modeling of functional breakdowns.
  - Data and information models.
  - Modeling of strategy.
  - Business rules models.

Business rules

Data models



Organizational structures

Purposes of BPMN Use

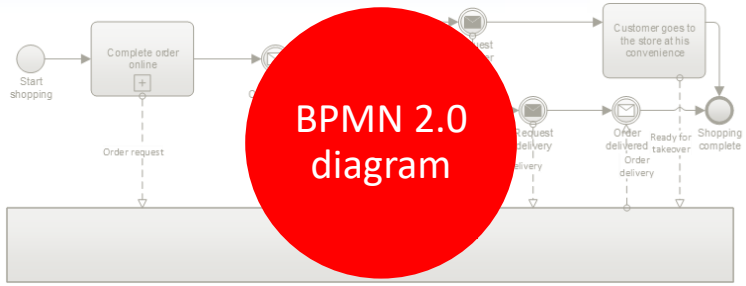
BPMN Conformance types define formal purposes of the use of BPMN and the corresponding software implementations. These slides focus on process modeling conformance.

Conformance type	Diagram types	Diagram elements	Execution
Process modeling	**Process, collaboration, conversation.**	**Full process modeling conformance.**	**Not available.**
Process execution with BPMN	Not available.	Not available.	Process diagrams.
Process execution with BPEL	Not available.	Not available.	Process diagrams.
Choreography	Choreography and collaboration (partially).	Choreography diagram elements and some basic process elements.	Choreography.

Focus of these slides

What is a BPMN Diagram?

- A BPMN diagram is actually a visual sentence expressed in BPMN notation. Similar to natural languages a notation (i.e. visual language) consists of a vocabulary, syntax and semantics.



BPMN 2.0 diagram

Expressed in

BPMN notation = BPMN vocabulary + BPMN syntax + BPMN semantics

Set of BPMN (visual) elements (i.e. symbols).

Set of BPMN rules for combining BPMN elements.

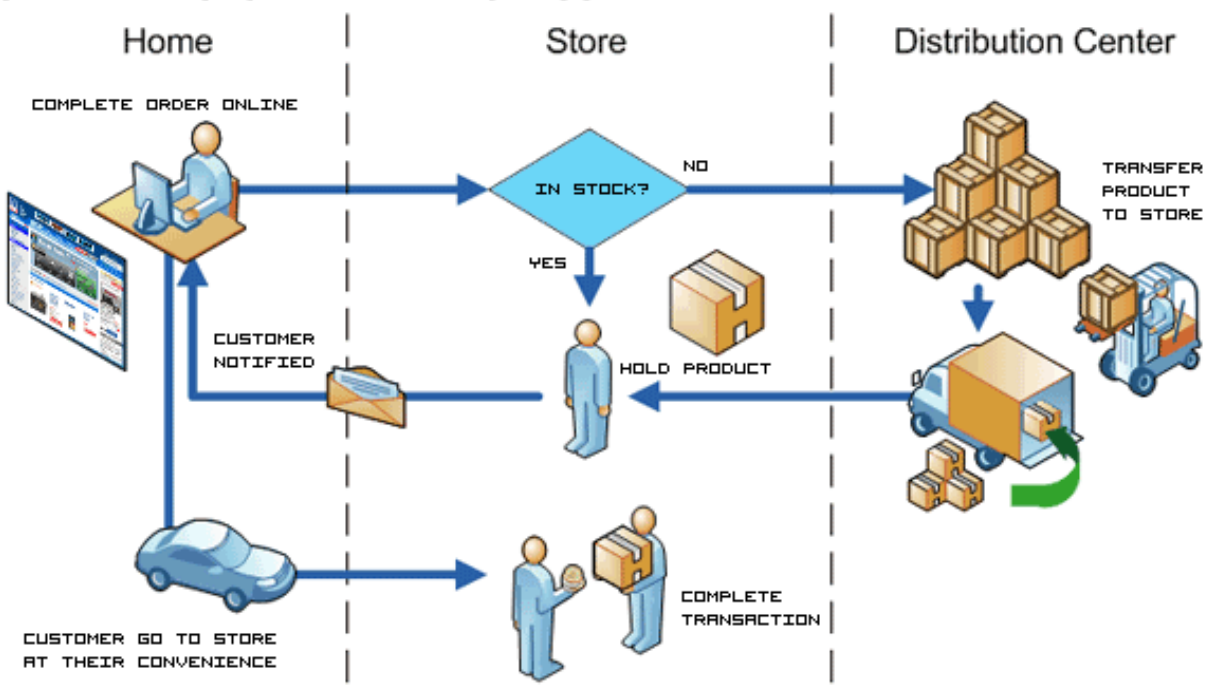
The meaning of BPMN elements and their combinations.

BPMN Diagram Types

BPMN 2.0
Page 22-23

- BPMN is designed to cover many types of modeling and allows the creation of ‚end-to-end' business processes.
- There are three basic types of sub-models within an ‚end-to-end' BPMN model:
  - Processes
  - Choreographies (not part of process modeling conformance)
  - Collaborations

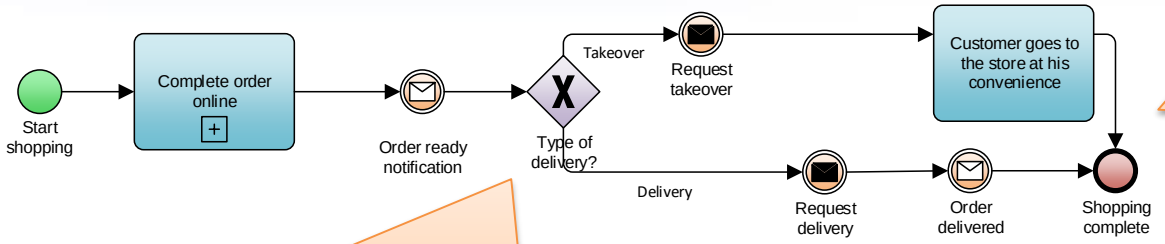
The majority of examples on the following slides is related to online store processes.



BPMN Process Diagrams

BPMN 2.0
Page 23,150

- In BPMN a process is depicted as a graph of flow elements (i.e. activities, events, gateways) and sequence flows that define finite execution semantics.



Private (internal) processes - represent a specific process (orchestration) in an organization. Can be executable or non-executable

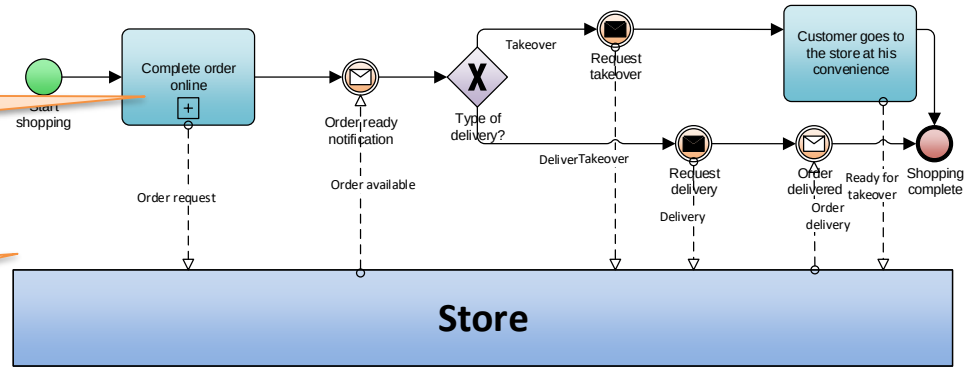
An **executable** process is modeled for the purpose of being executed.

A **non-executable** process modeled for the purpose of documenting process behavior at a modeler-defined level of detail.

Processes can be defined at any level: from enterprise-wide processes to processes performed by a single person.

Only those activities that are used to communicate to other participants are included in the public process.

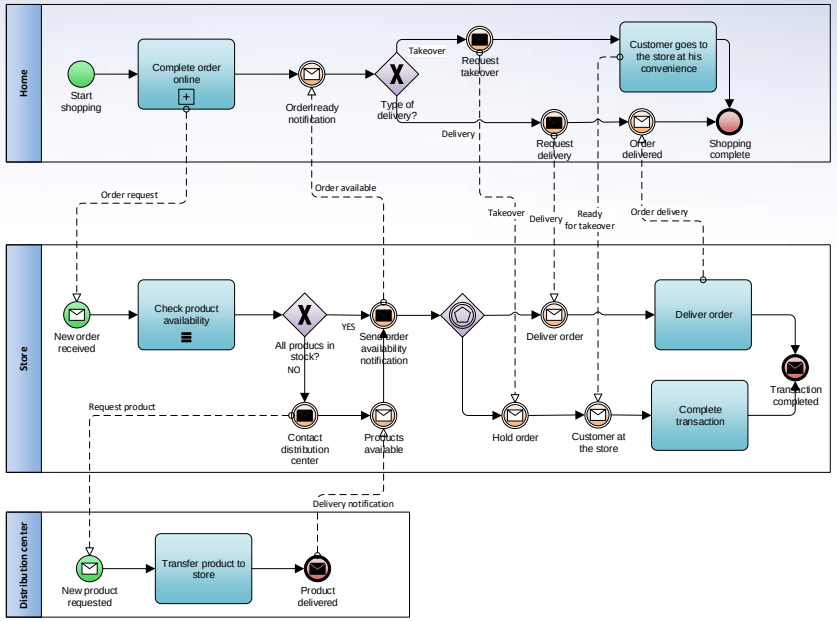
Public process - represents interactions between a private business process and another process or participant.



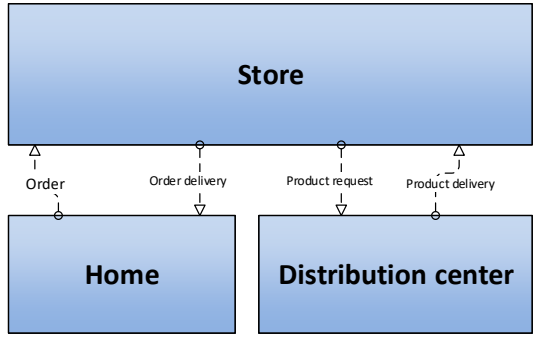
BPMN Collaboration Diagrams

- A collaboration represents the interactions between two or more business entities (e.g. processes).
- A collaboration usually contains two or more pools (black-box or white-box), representing the participants in the collaboration.
- The message exchange between the participants is shown by a message flow that connects two pools (or the elements within the pools).

Collaboration model with white-box pools (i.e. visible details)



Collaboration model with black-box pools (i.e. hidden details)



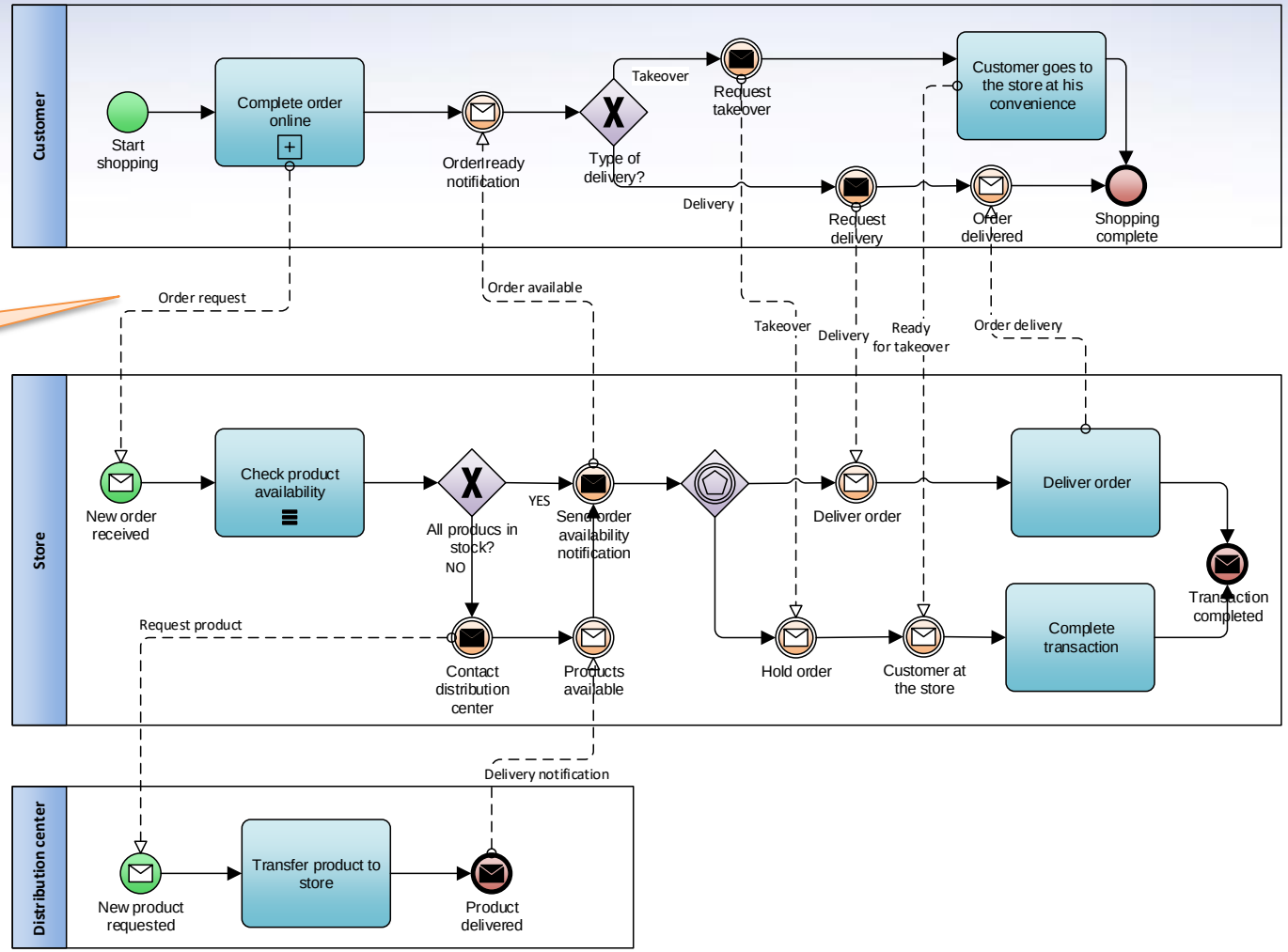
BPMN Collaboration Diagram Example

A customer starts shopping and completes order online. The order is sent to Store. Then the customer waits for a notification that the product is ready. Based on preferred type of delivery (takeover or delivery) the customer receives or gets the desired product.

Message flows are used for between-process interactions and synchronization.

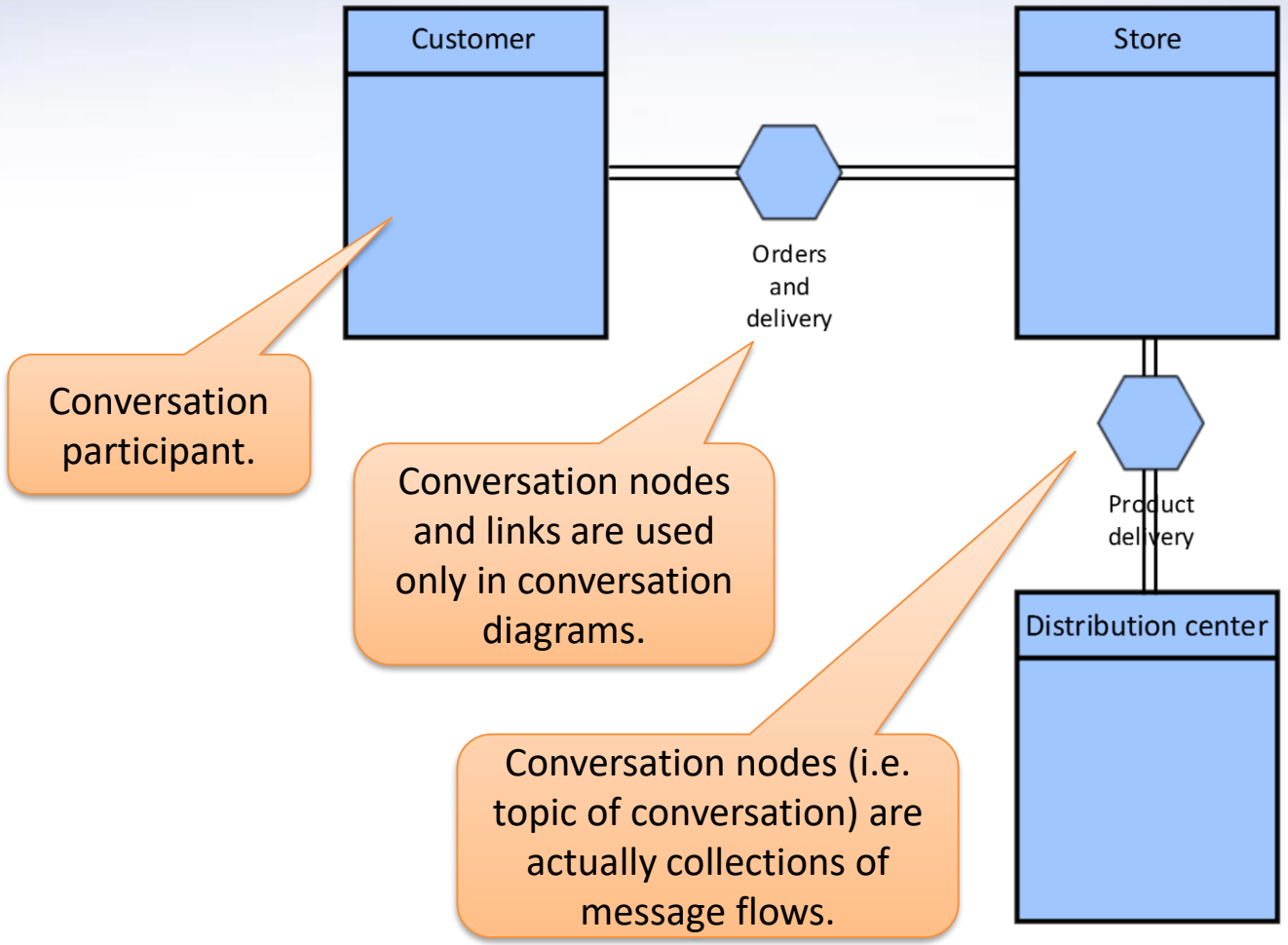
When the store receives a new order it checks for products availability. In case a product is not in stock, it contacts distribution center. When all order‘s products are available the customer is notified that the product is ready. Based on preferred type of delivery, the product is delivered to customer and the transaction completes.

In case of a new product request, the desired product is delivered to requesting store.



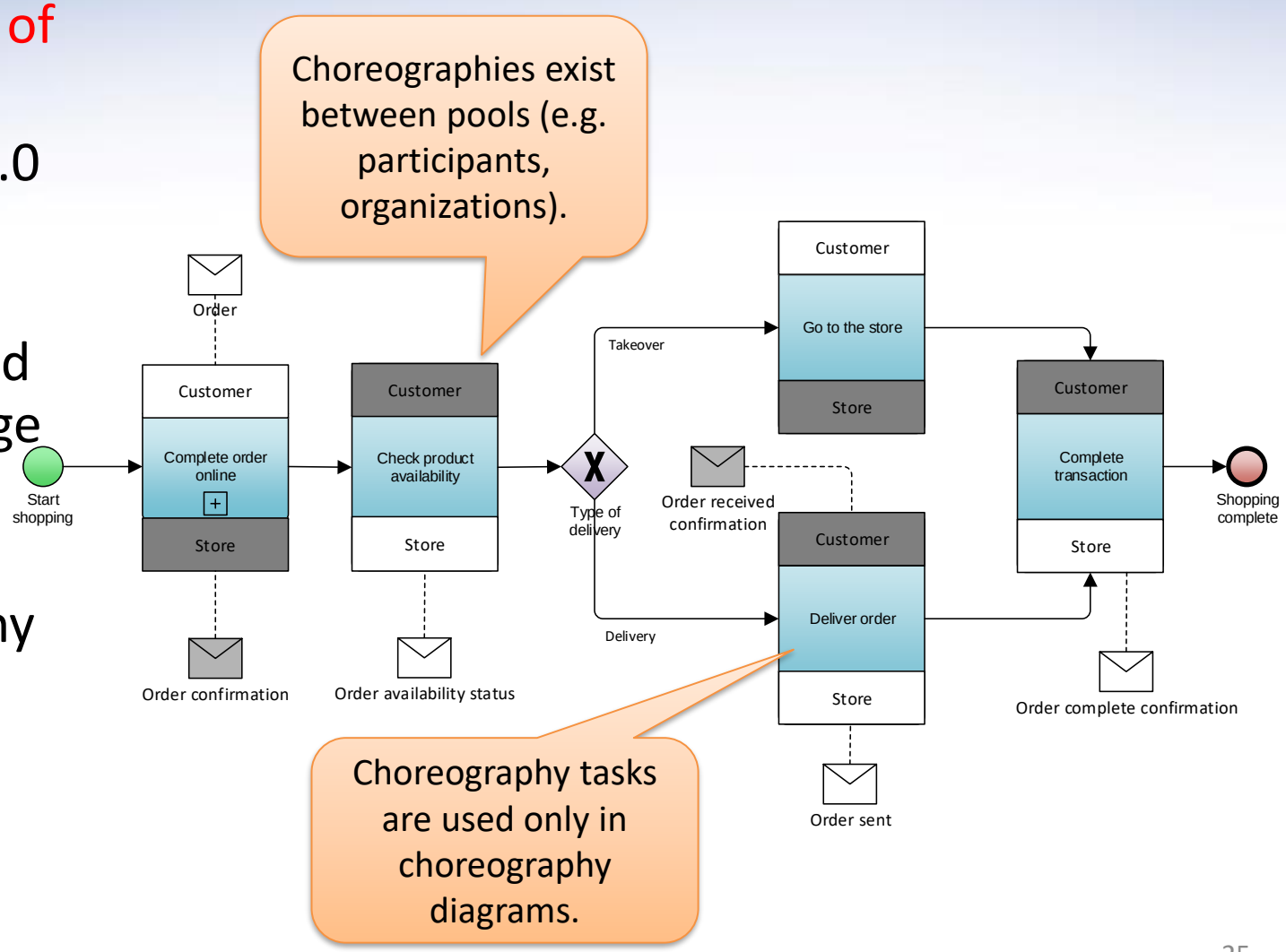
BPMN Conversation Diagrams

- A conversation diagram provides an overview of which partners of a certain domain co-operate on which tasks.
- Conversation diagrams represent a specific (i.e. top level) 'view' of collaboration diagrams.
- Conversation diagrams use simple notation: participants, conversation nodes (hexagons) and conversation links (i.e. a series of message exchanges).



BPMN Choreography Diagrams

- Choreography models are NOT part of process modelling conformance.
- Choreographies are new in BPMN 2.0 and focus on between-processes interactions and message flows.
- A choreography diagram can be used to analyse how participants exchange information to coordinate their interactions.
- Another way to look at choreography is to view it as a type of business ‘contract’ between two or more organizations.



Common Use of Different BPMN Diagrams

BPMN 2.0
Page 133, 138

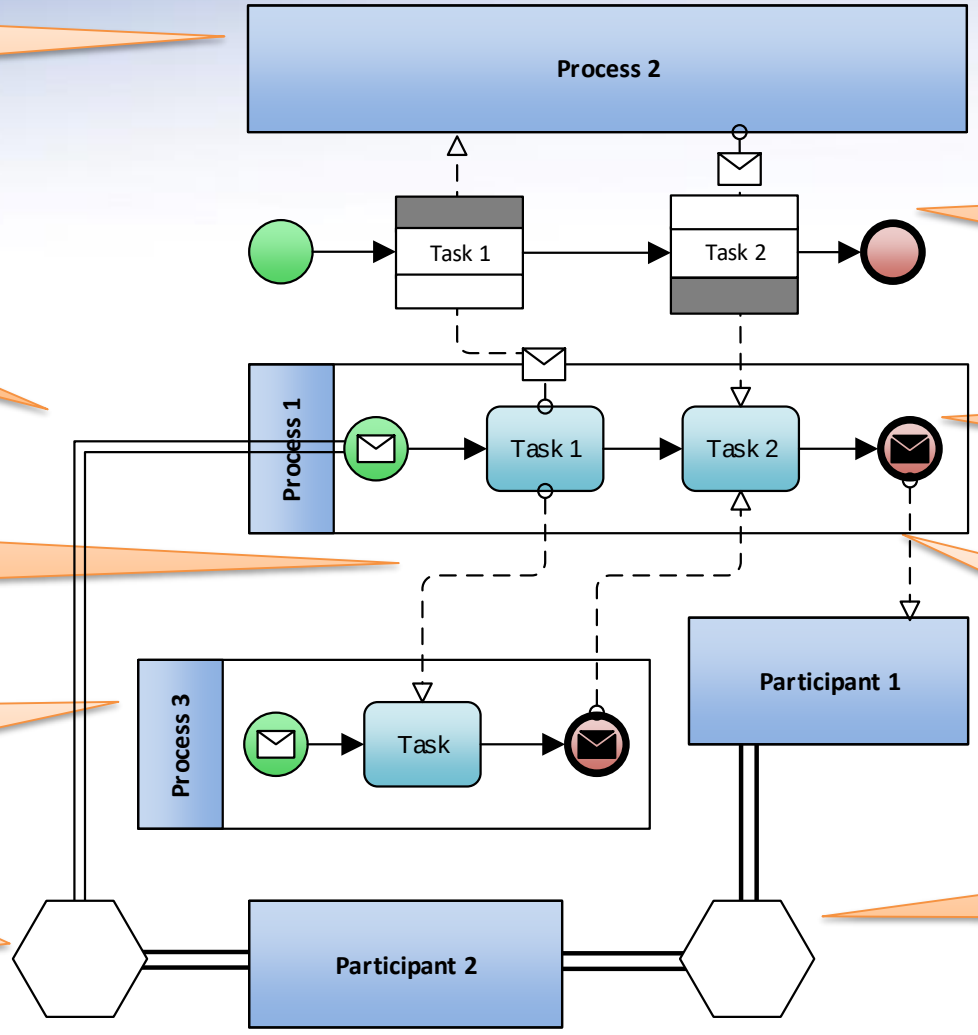
Black-box process 2.

Conversation link to Process 1 event.

Message exchanges in Process 1 - Process 3 collaboration.

Process model for Process 3.

Conversation between Participant 2 and Process 1.



Choreography model within Process 1 – Process 2 collaboration model.

Process model for Process 1.

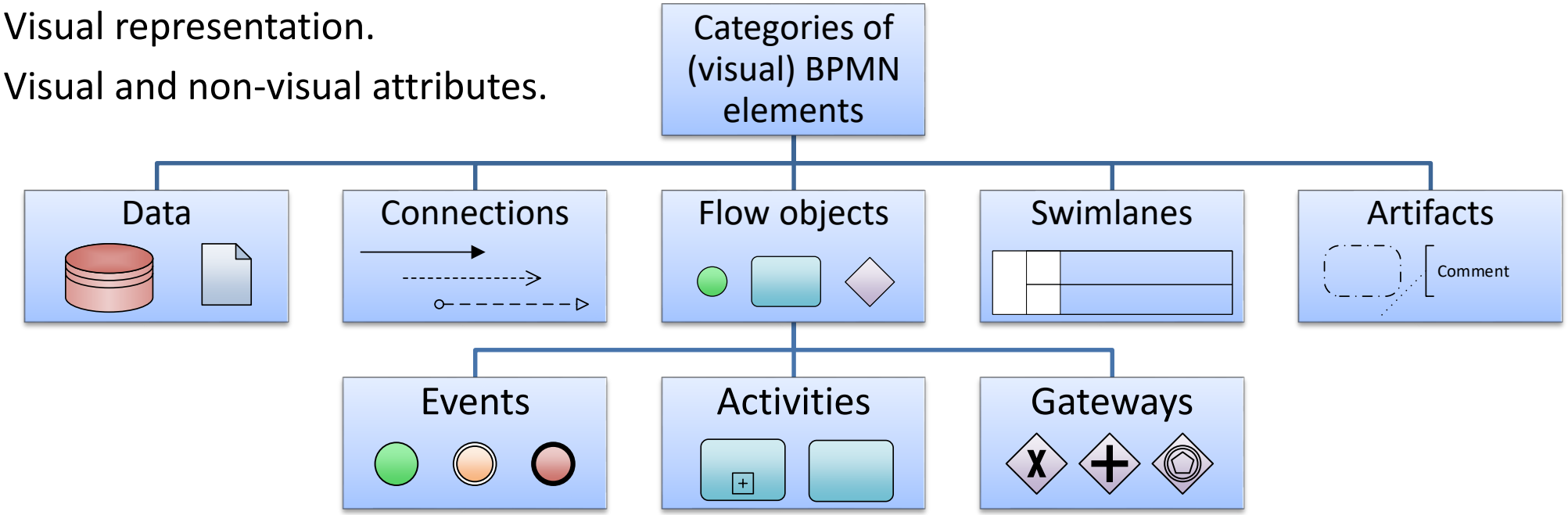
Collaboration between Process 1 and Participant 1.

Conversation model between Participant 1 and 2.

BPMN Process Modeling Elements

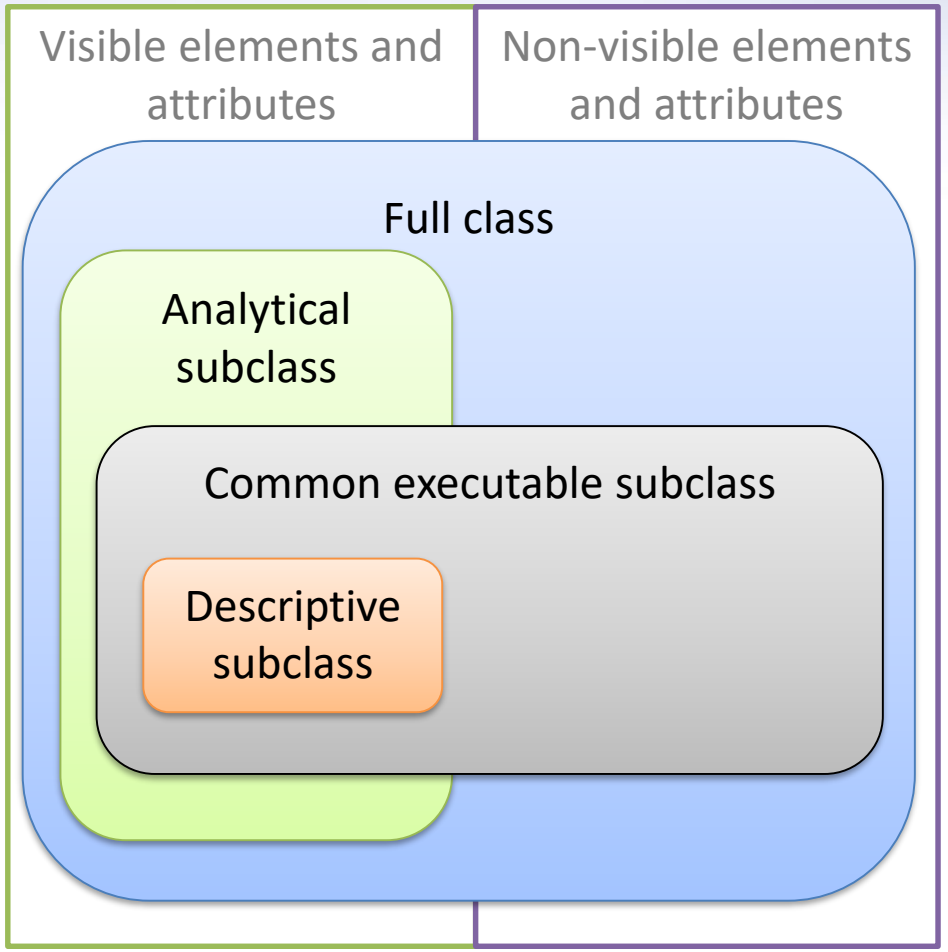
BPMN 2.0
Page 27-28

- BPMN diagrams are ‘graphs’ of BPMN elements.
- BPMN elements have defined:
  - Syntax – rules about how to use BPMN elements in BPMN diagrams.
  - Semantics – meaning of BPMN elements.
- BPMN elements may have:
  - Visual representation.
  - Visual and non-visual attributes.



Classes of BPMN Elements

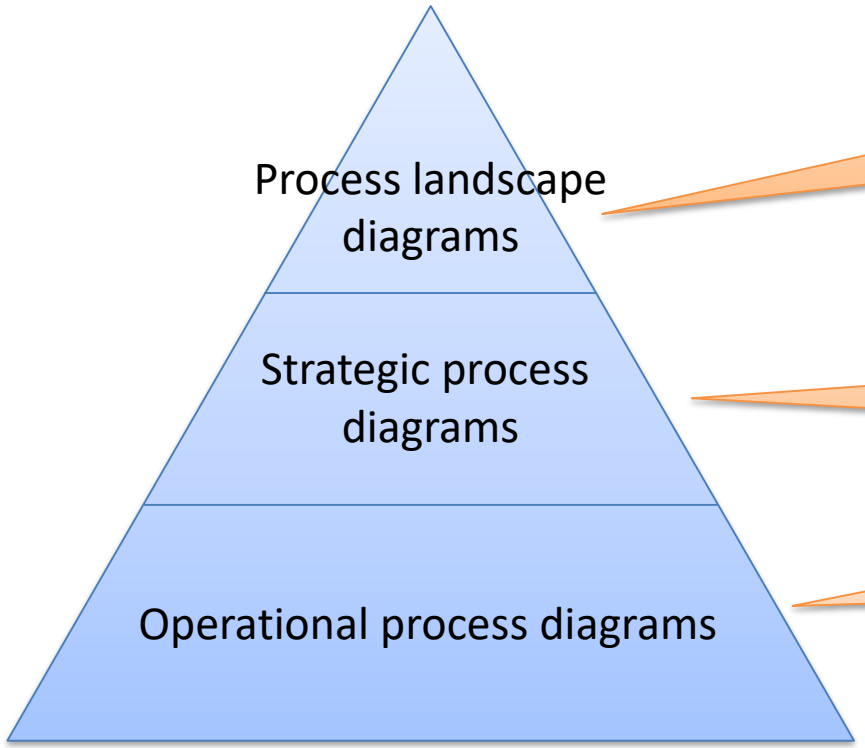
- The full set of BPMN process modeling elements is additionally divided into three subclasses:
  - Descriptive subclass of elements is concerned with visible elements and attributes used in high-level modeling. It should be comfortable for majority of process analysts.
  - Analytic subclass of elements is concerned with visible elements and attributes used in detailed process modeling. Includes all elements from the descriptive level and about half of all process modeling elements.
  - Common executable subclass of elements focuses on what is needed for executable process models. The palette of elements is between descriptive and analytical, but includes attributes related to executable details.



(Silver 2011)

BPMN in a Process Architecture

- A process architecture represent a hierarchically (i.e. process abstraction levels) and horizontally (i.e. relationships between processes) organized system of an organization's business process models.



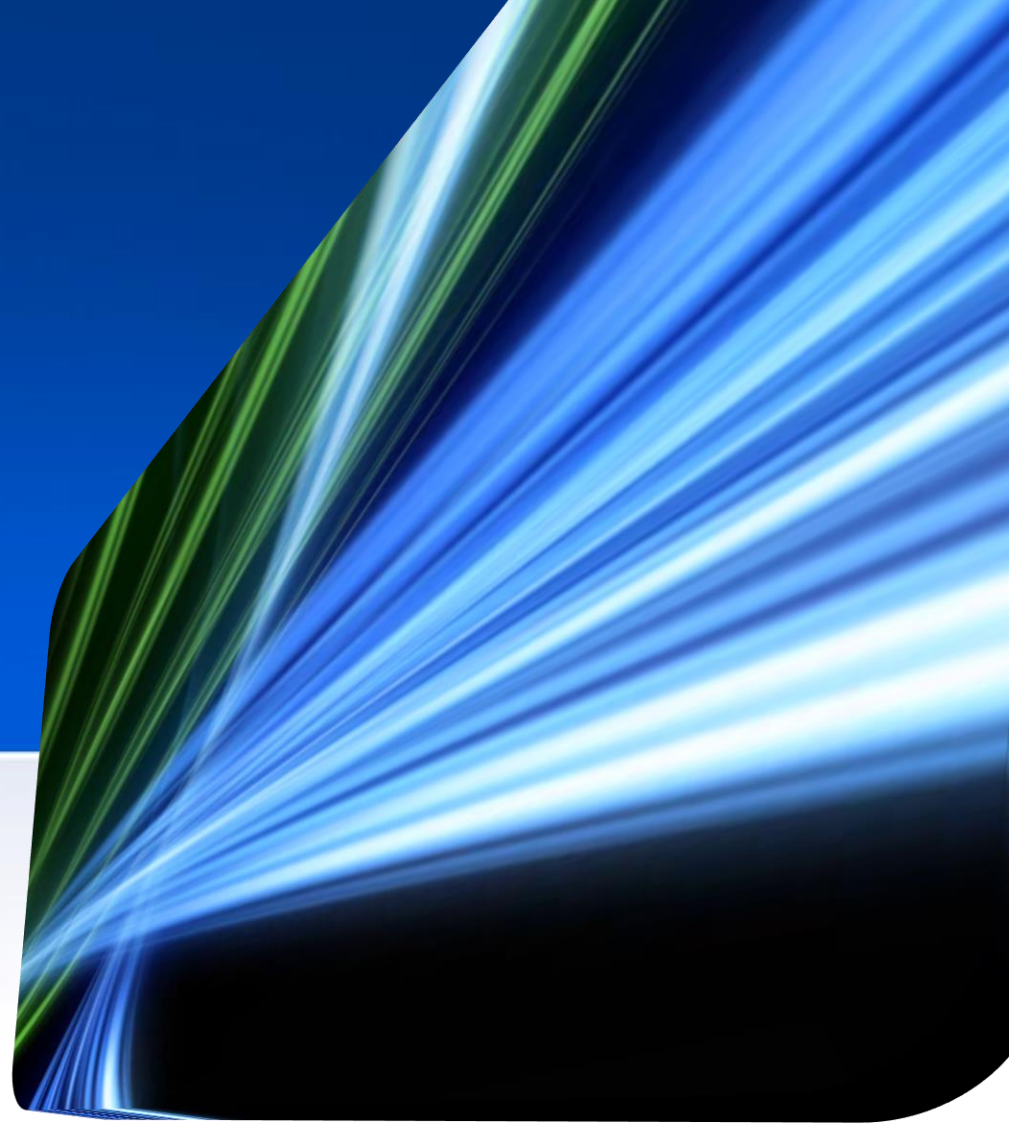
Process architecture levels

Used for representing an overview of organizational processes and their relationships. Partially supported by BPMN (i.e. Conversation diagrams).

Used for a conceptual overview of individual business processes. Supported by BPMN with the use of descriptive set of BPMN elements.

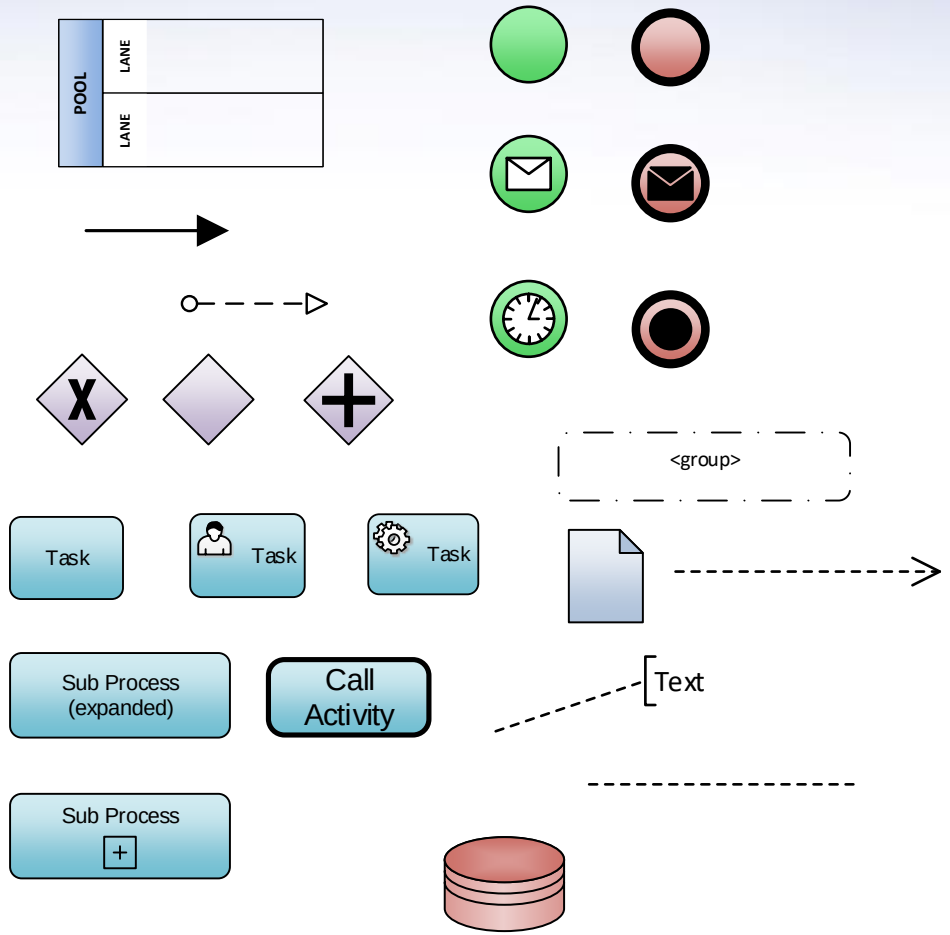
Used for detailed business process descriptions (i.e. workflows) intended for (technical or human) operation. Supported by BPMN with the use of the full set of BPMN elements.

BASIC SET OF PROCESS MODELING ELEMENTS



Descriptive Subclass of Process Modeling Elements

- Used to create process diagrams which are readily understood by almost any business person and supported by almost all BPMN tools.
- Suited for high-level process modeling.
- Should be comfortable for analyst that have used “flowcharts”.

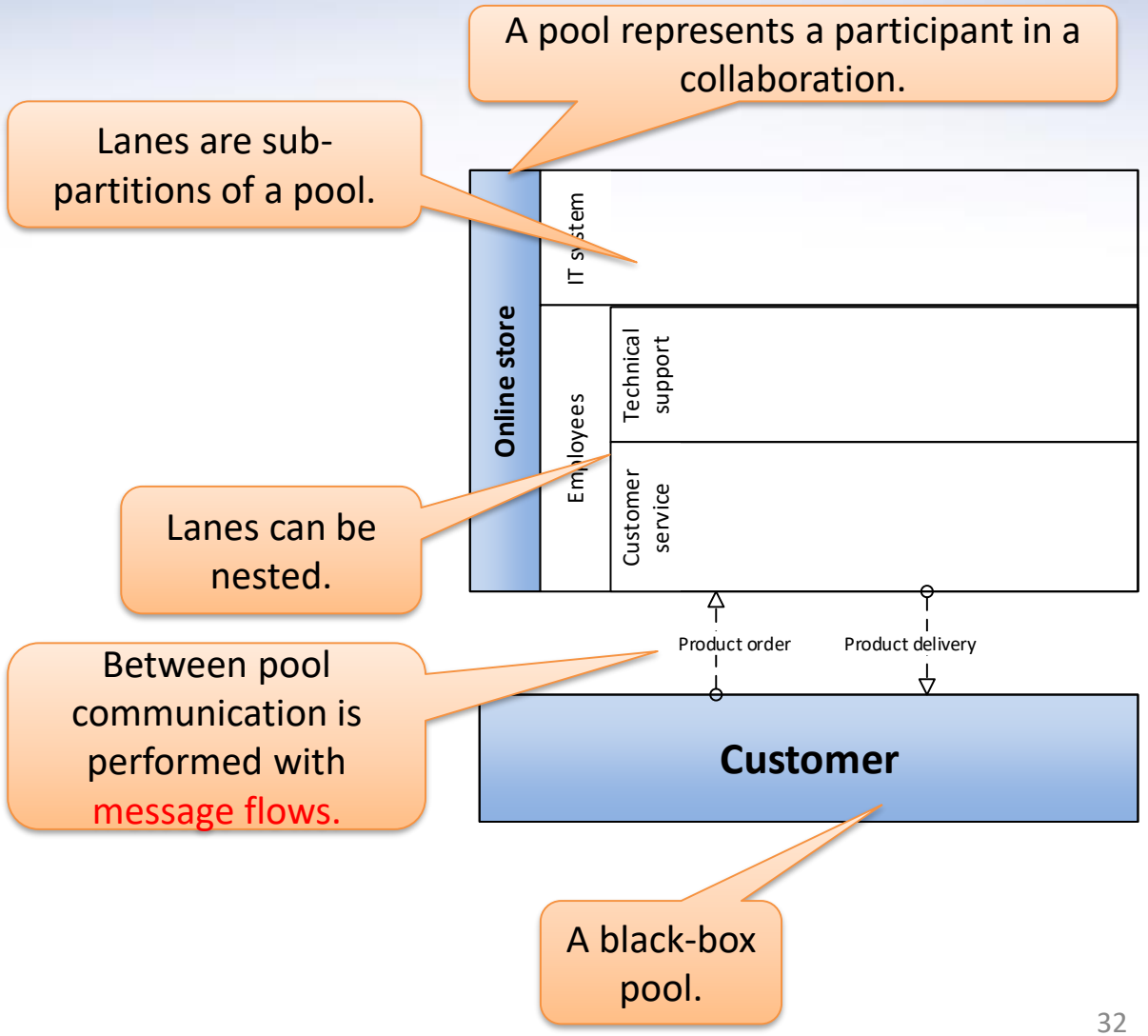


Descriptive subclass of BPMN elements

Swimlanes and Message Flows

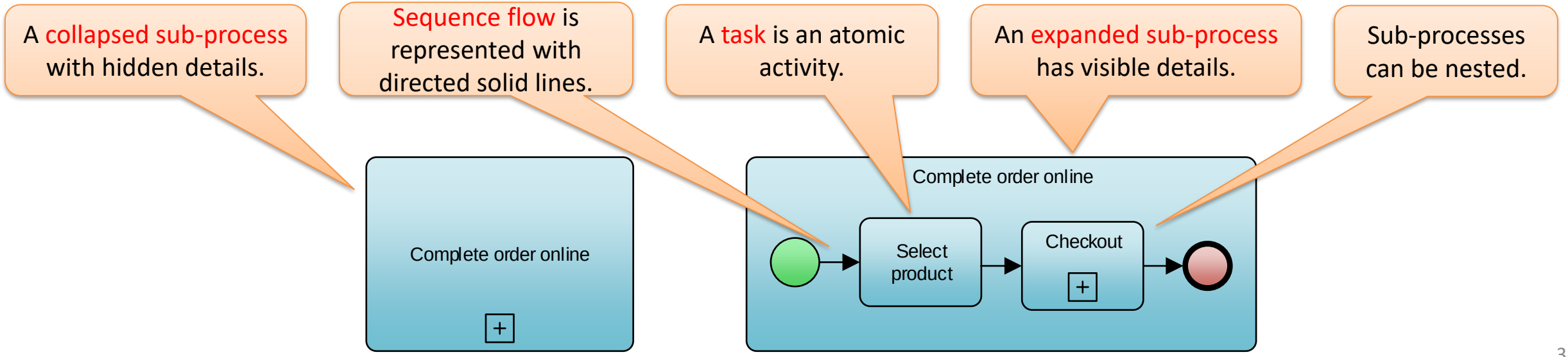
BPMN 2.0
Page 29-30

- A swimlane is a graphical container for partitioning a set of activities from other activities.
  - A pool is a container for partitioning a Process from other Processes or Participants.
  - Lanes are used to organize and categorize activities within a Pool.
- Between pools communication is modeled with message flows.
- Pools are used in process and collaboration diagrams.



Activities and Sequence Flows

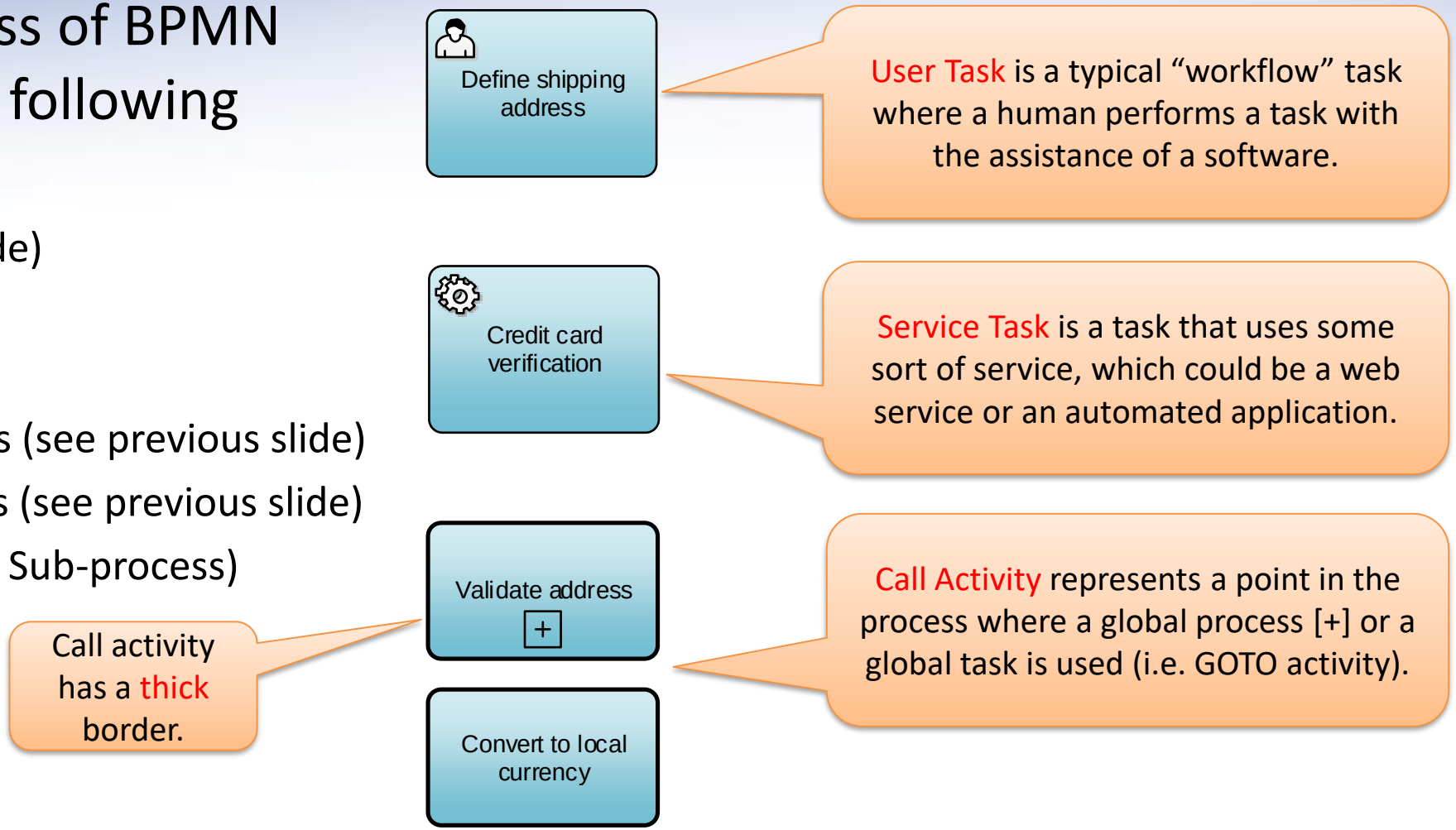
- An activity is a generic type of work that an individual or company performs.
- An activity can be:
  - Atomic (task) or
  - Compound (process, sub-process) – uses a “+” sign.
- The sequence of activities is represented with sequence flow connections.



Descriptive Subclass of Activities

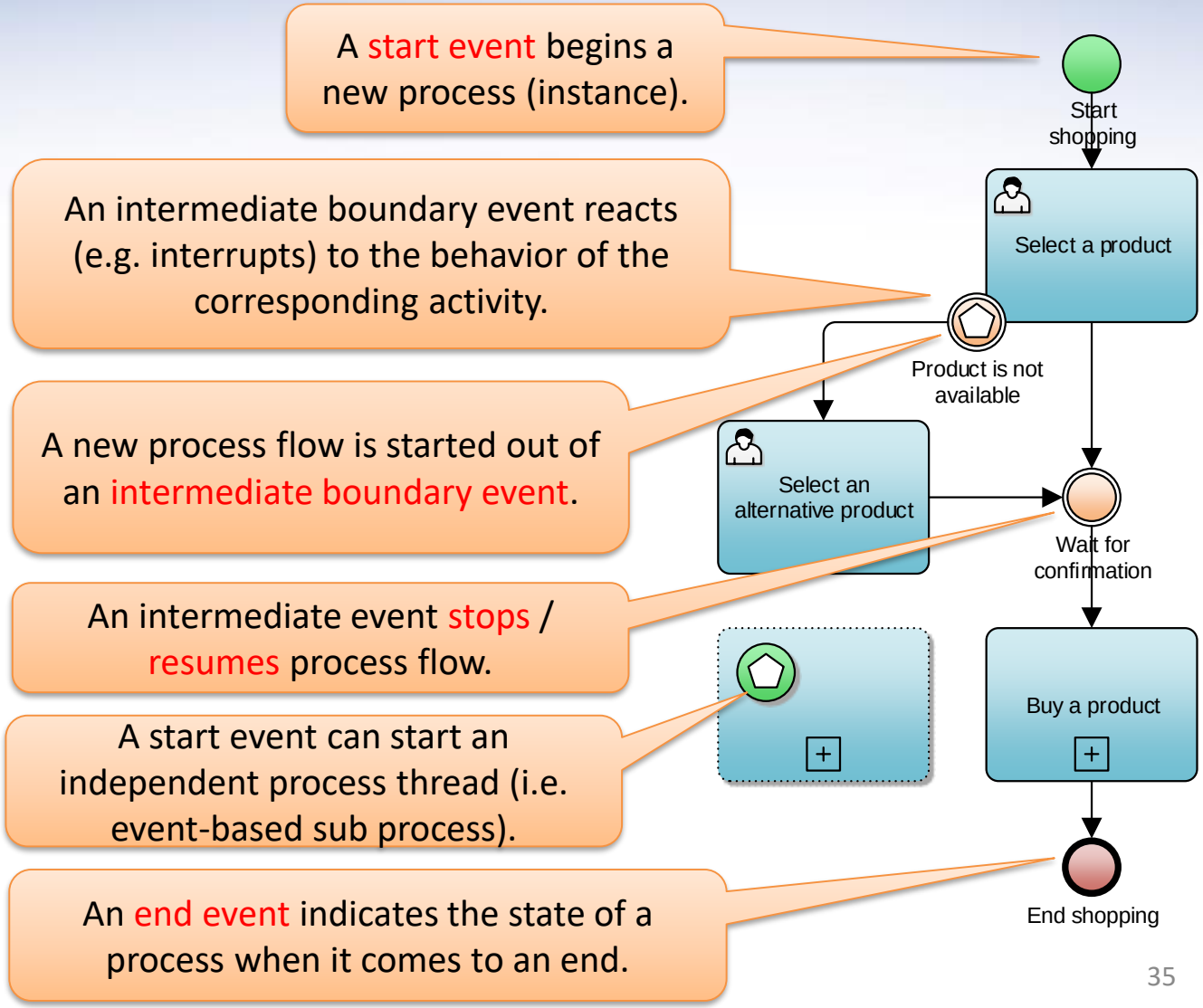
BPMN 2.0
Page 158-184

- Descriptive subclass of BPMN elements includes following activities:
  - Task (see previous slide)
  - User Task
  - Service Task
  - Expanded sub-process (see previous slide)
  - Collapsed sub-process (see previous slide)
  - Call Activity (Task and Sub-process)



Events

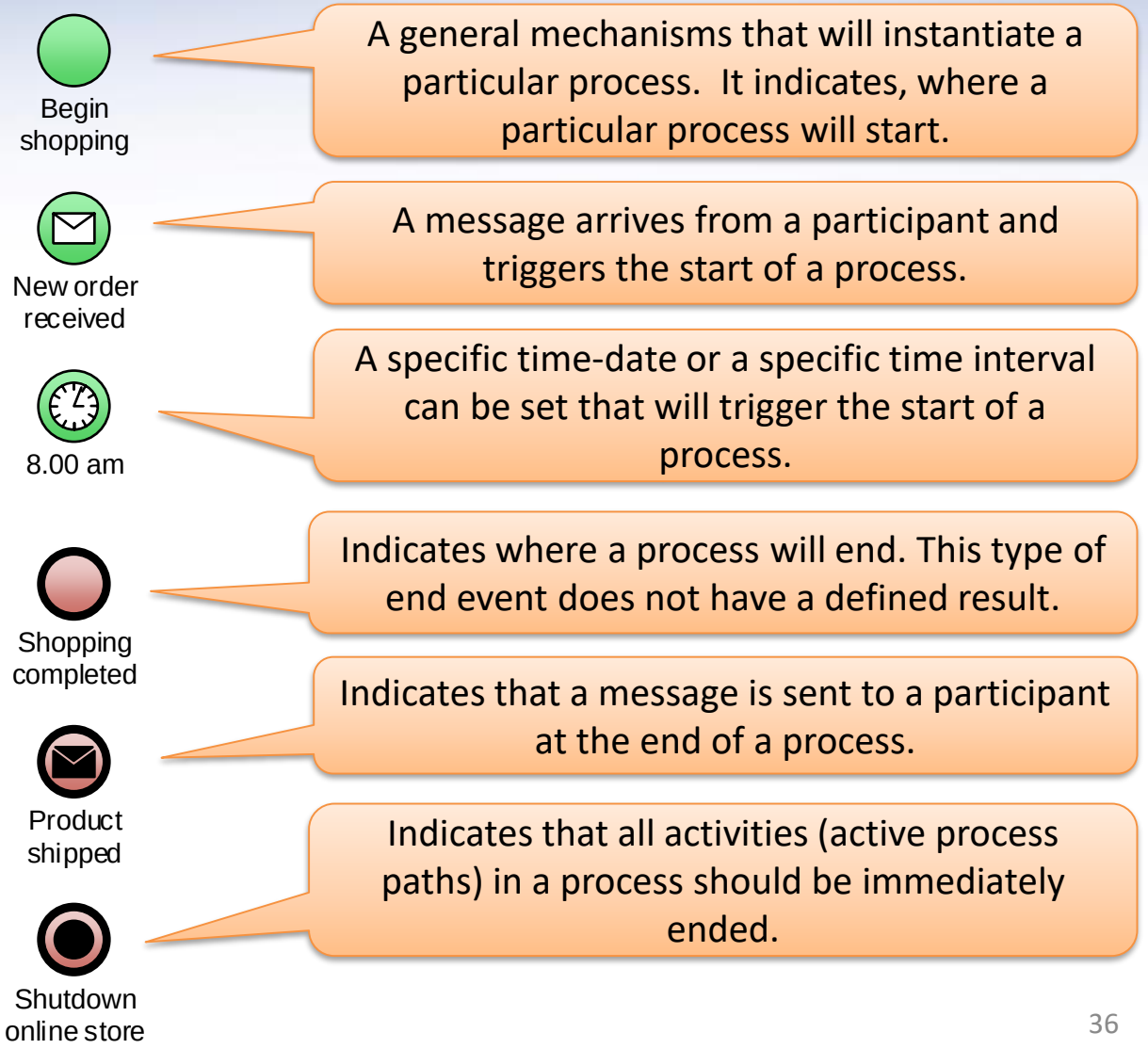
- An event is something that »happens« during the process.
- Basic (descriptive) events can start a process, as well, they occur at the end of a process.
- Icons within the circle shape of an event can indicate the type of triggering (e.g. a message, time condition, rule, etc.).



Descriptive Subclass of Events

BPMN 2.0
Page 233-287

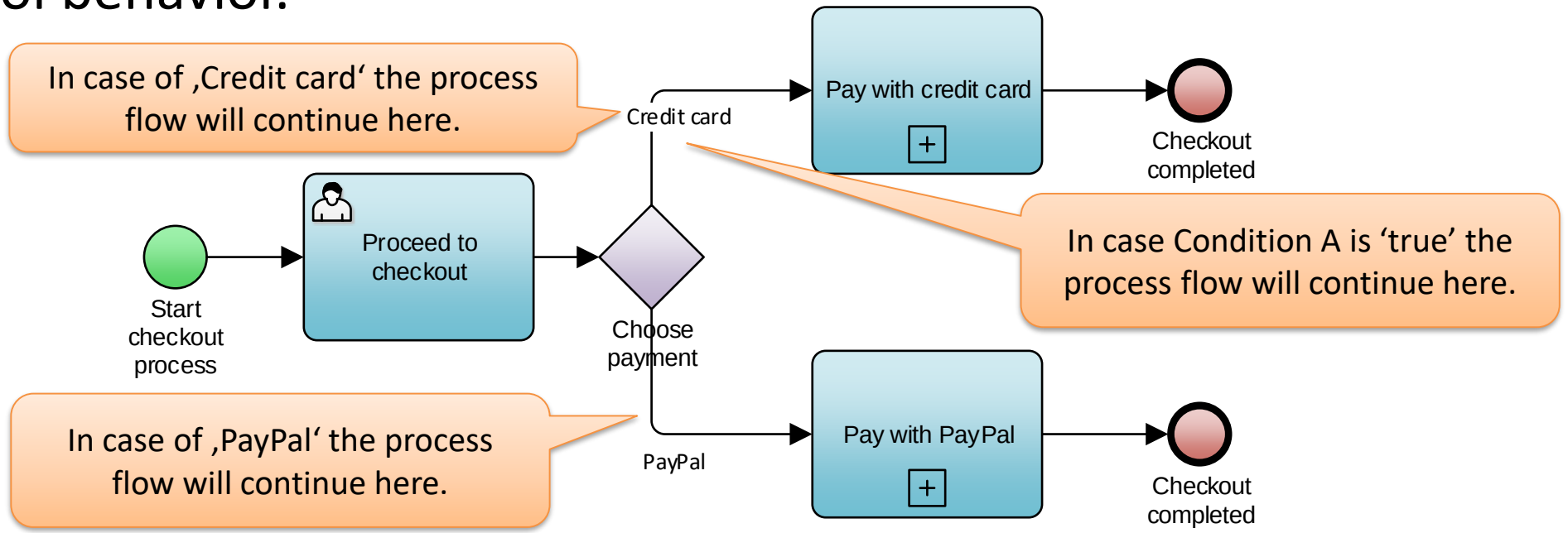
- Descriptive subclass of BPMN elements includes following events:
  - Generic start event
  - Generic end event
  - Message start event
  - Message end event
  - Timer start event
  - Terminate event



Gateways

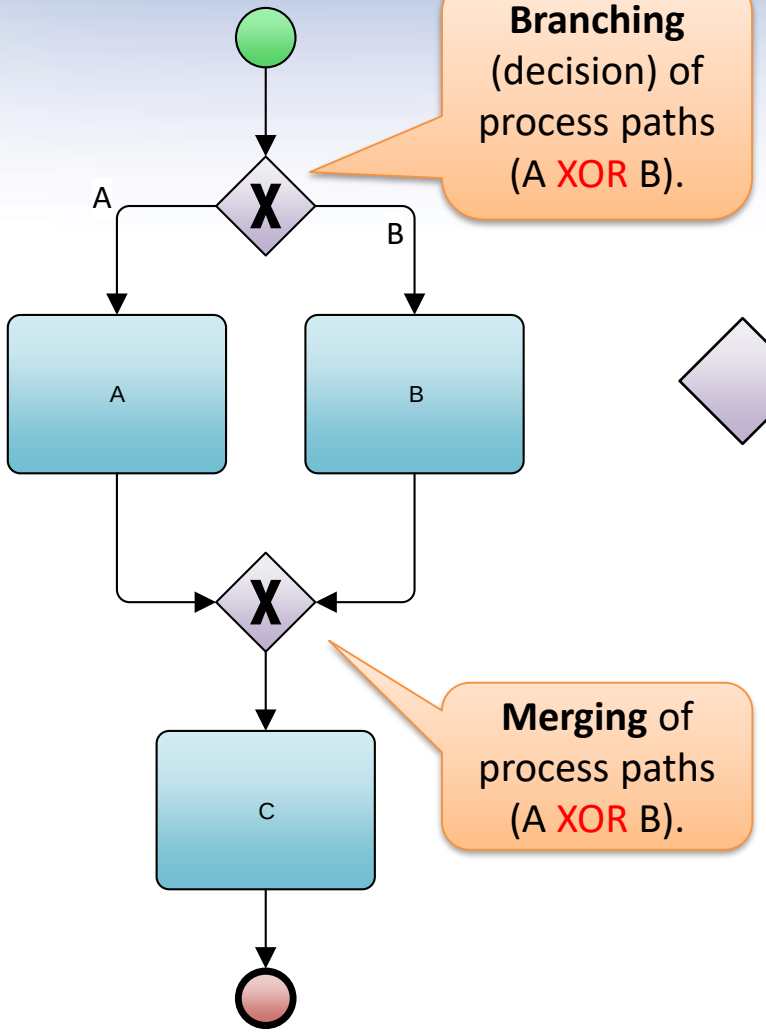
BPMN 2.0
Page 29, 34

- A gateway is used to control process flow.
- A gateway determines branching, forking, merging and joining of business process's paths.
- Icons within the diamond shape of a gateway indicate the type of flow control behavior.

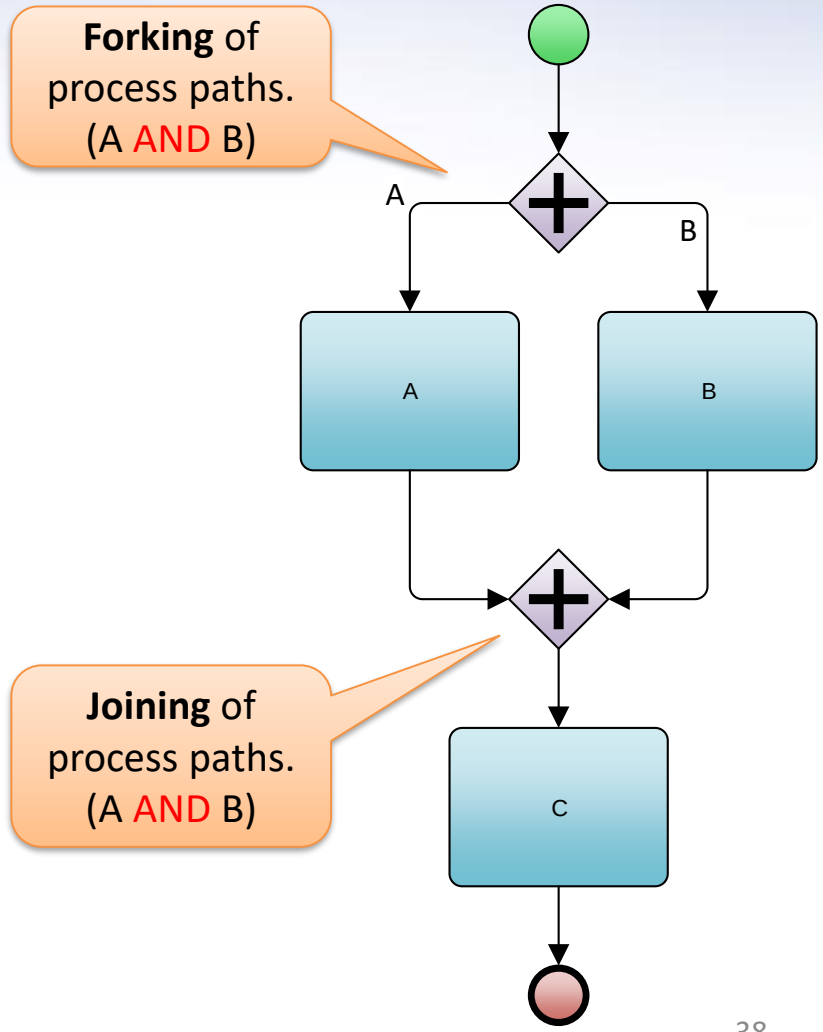


Descriptive Subclass of Gateways

BPMN 2.0
Page 287-302



- Descriptive subclass of BPMN elements includes following Gateways:
  - Exclusive (XOR) gateway. It is used to create alternative paths within a process flow.
  - The Exclusive (XOR) gateway MAY or MAY NOT use a marker that is shaped like an “X”.
  - Parallel (AND) gateway. It is used to create parallel flows and to synchronize (combine) parallel flows.

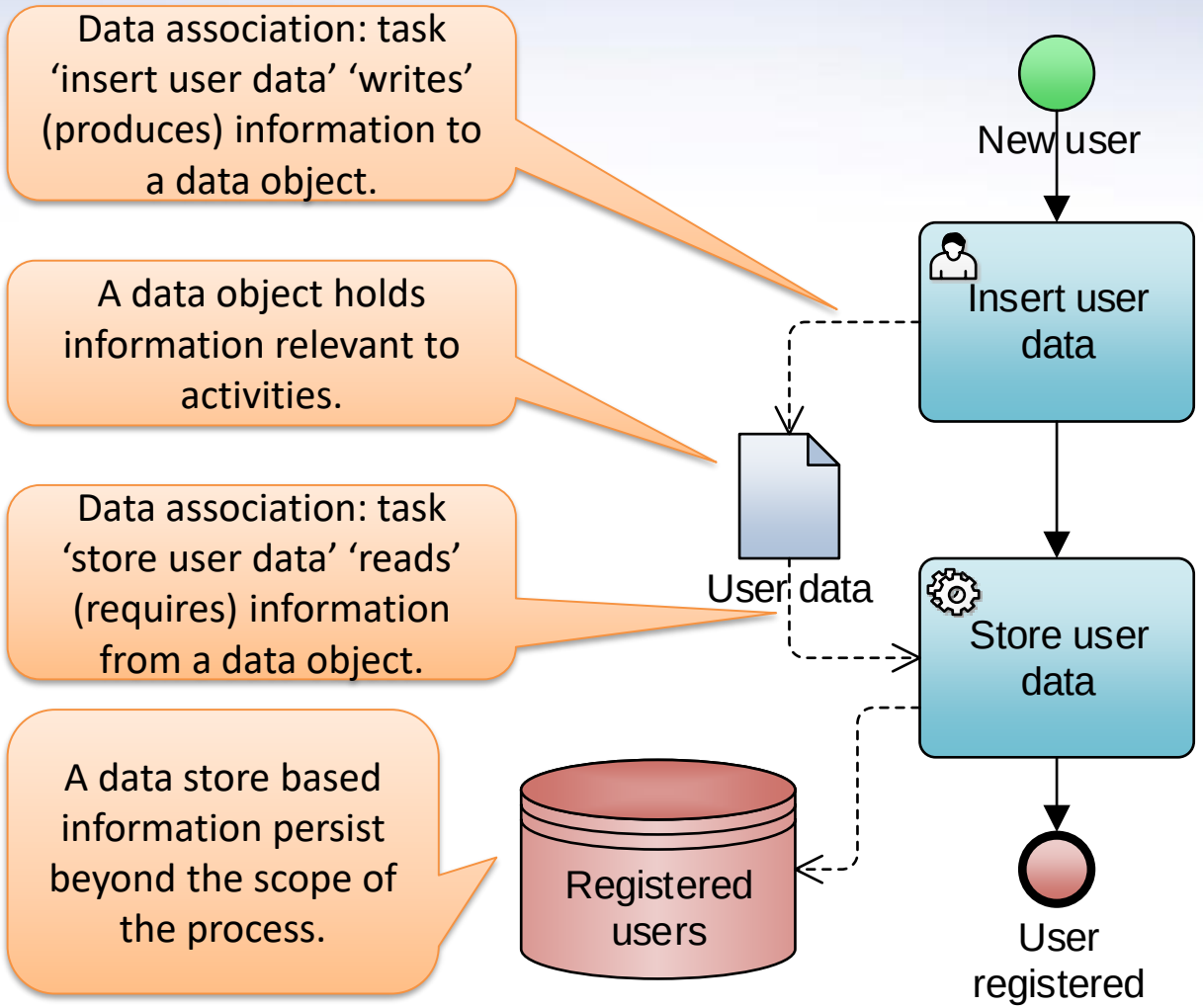


Data and Data Associations

Bpmn 2.0
Page 30, 208

- Data objects provide information about what activities require to be performed and/or what they produce.
  - Local variable in a process level.

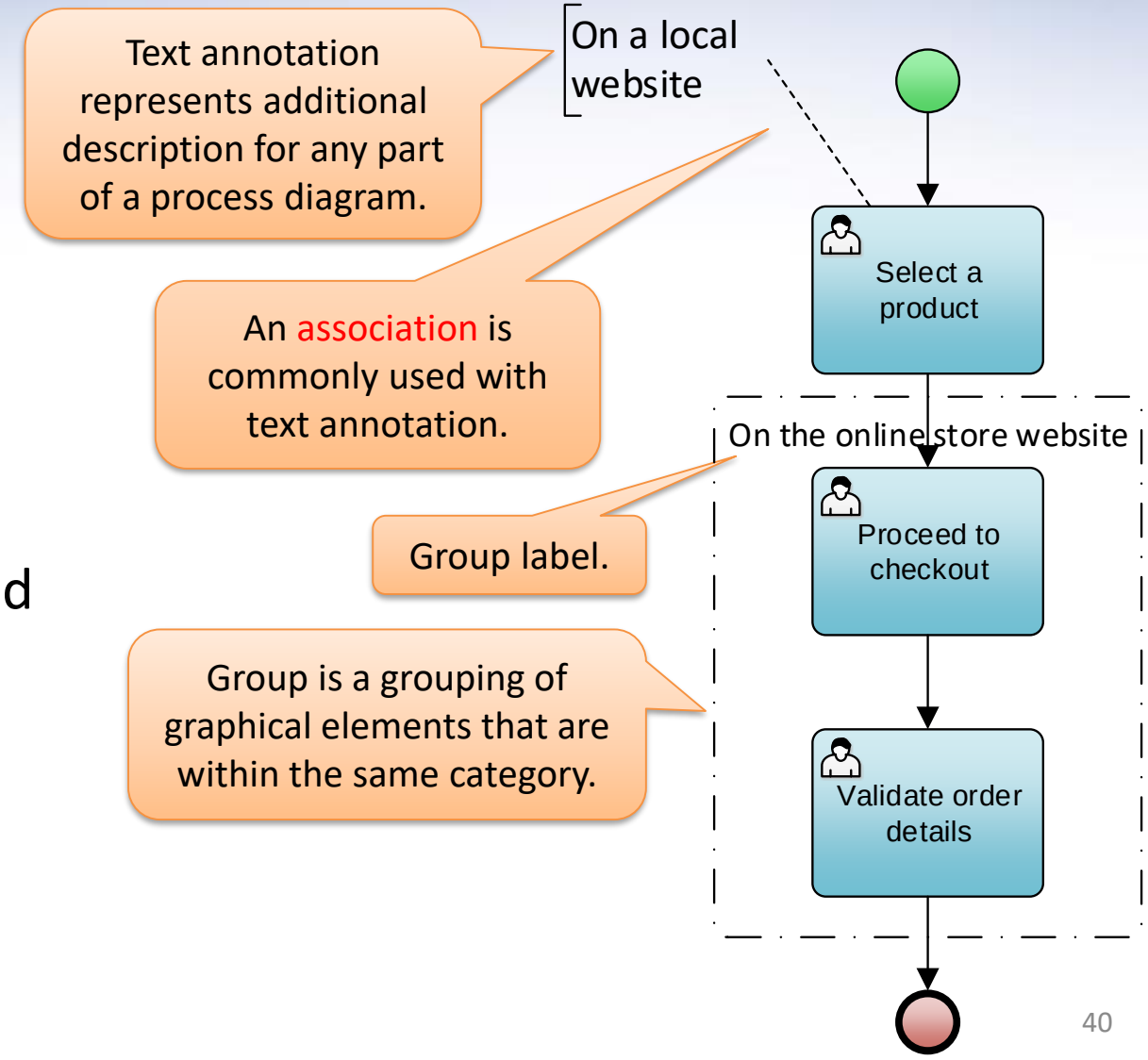
- Data store represents persistent data, which exists outside the scope of the process.
- Data objects and data store elements are connected to other process elements (i.e. activities, events) with data associations (dotted arrows).



Artifacts and Associations

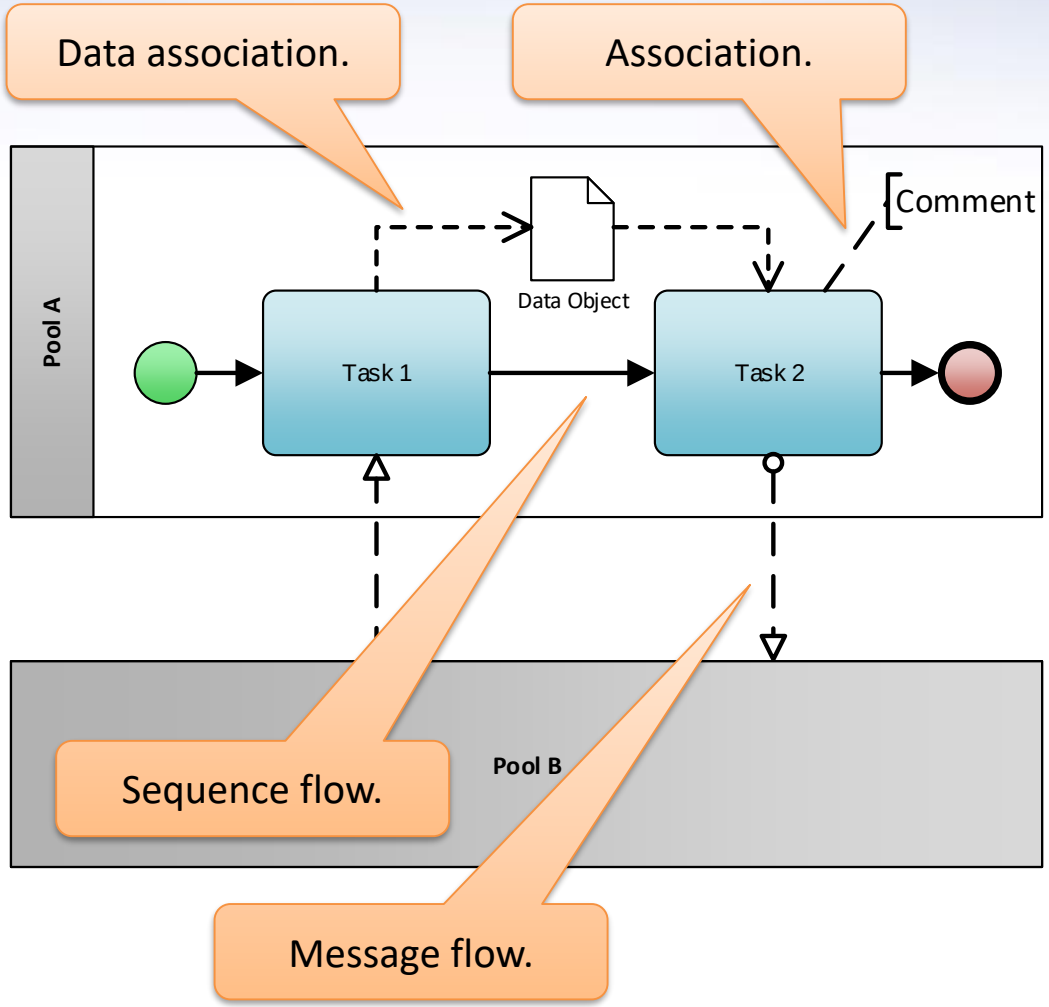
BPMN 2.0
Page 29,30

- Artifacts are used to provide additional information about the process to modelers.
- Artifacts DO NOT affect process flow.
- Artifacts are: text annotation, group and documentation.
  - Documentation is not a visible element. It is an attribute of most elements.
- An association is used to link information and artifacts with BPMN graphical elements.



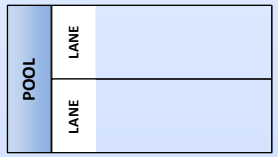
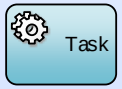
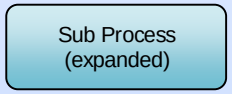
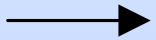
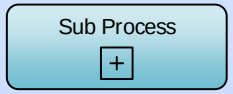
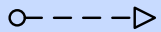
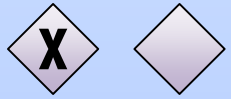
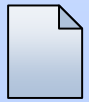
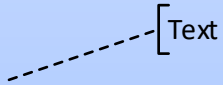
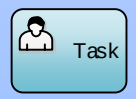
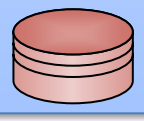
Connecting Objects

- Connecting object are visually represented with different styles of non-directed or directed lines.
- Sequence flows represent the order of performed activities.
- Data associations represent information flows.
- Associations connect artifacts with other bpmn elements.
- Message flows represent exchange of messages between pools.

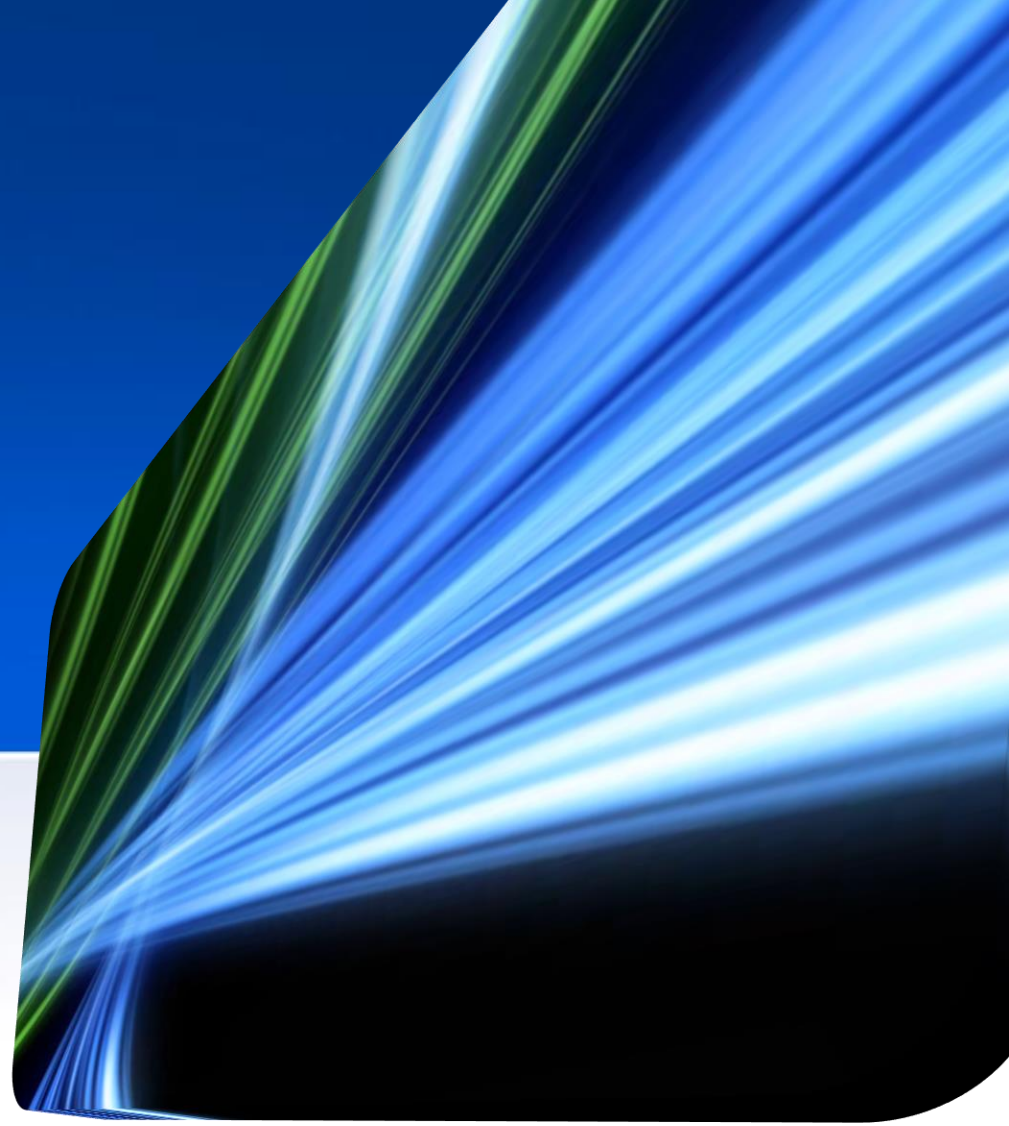


Descriptive Process Modeling Elements

BPMN 2.0
Page 3-4

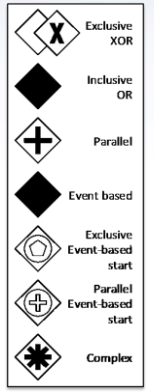
Element	Graphical representation	Element	Graphical representation	Element	Graphical representation
participant (pool)		serviceTask		startEvent (None)	
laneSet		subProcess (expanded)		endEvent (None)	
sequenceFlow (unconditional)		subProcess (collapsed)		messageStartEvent	
messageFlow		CallActivity		messageEndEvent	
exclusiveGateway		DataObject		timerStartEvent	
parallelGateway		TextAnnotation		terminateEndEvent	
task (None)		association/dataAssociation		Documentation	Non-visual element
userTask		dataStoreReference		Group	

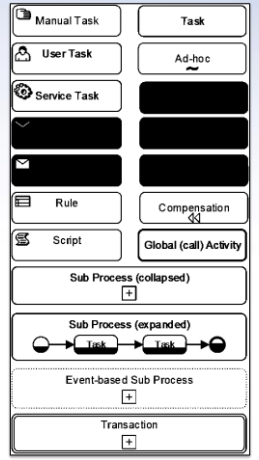
FULL SET OF PROCESS MODELING ELEMENTS

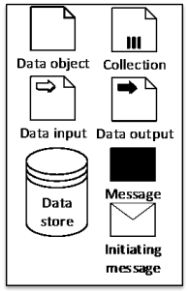


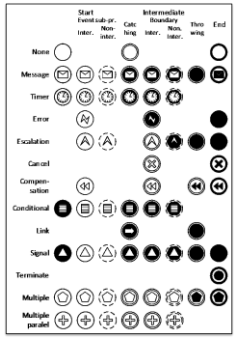
Full Process Modeling Conformance Elements

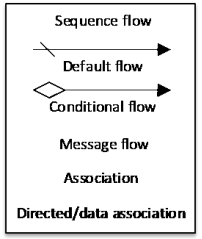
- Defines all elements, which can be used in process, collaboration and conversation diagrams:
  - All task types, embedded sub-processes, call activity,
  - All gateway types,
  - All event types (start, intermediate, and end),
  - Lane, participants, data object (including data input and data output), message, group, text annotation,
  - Sequence flow (including conditional and default flows), message flow,
  - Conversations (limited to grouping message flow, and associating correlations), correlation, and association (including compensation association).
  - Markers (loop, multi-instance, transaction, compensation) for tasks and embedded sub-processes).
- Out of scope: choreography modeling elements (e.g. choreography task and sub-choreography).







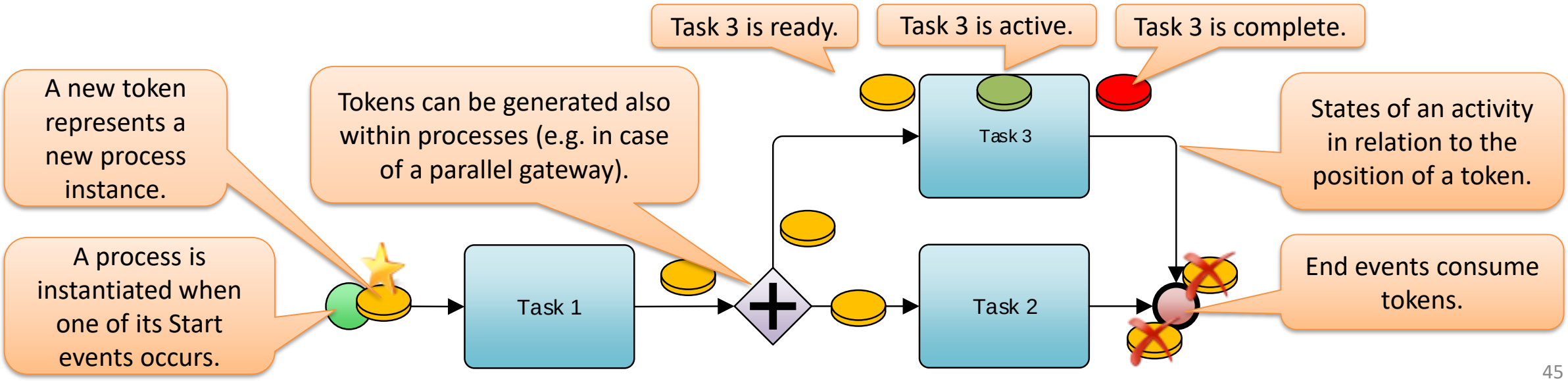




About Process Instances and Tokens

BPMN 2.0
Page 456

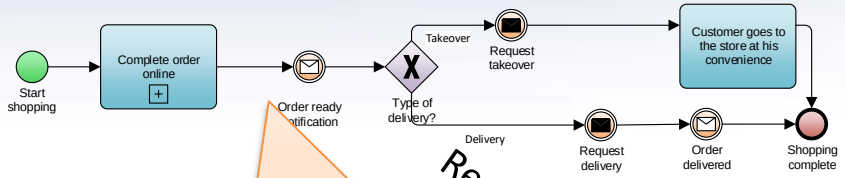
- A process can be executed or performed many times, but each time is expected to follow the steps defined in the process model. A single process performance represents a process instance.
- The behavior of a process instance is commonly represented with the flow of tokens.



BPMN Process, Pool and Participant

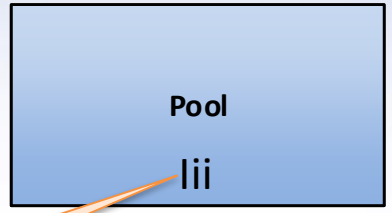
Bpmn 2.0
Page 112, 145

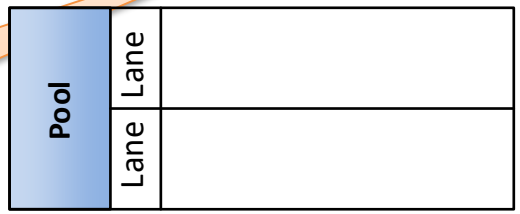
Process

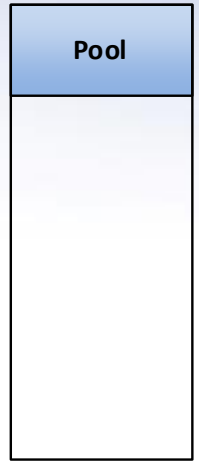


A pool may include a process.

A pool is not required to contain a process (i.e. it can be “black box”).







This symbol (III) annotates that a pool has multiple participants.

In BPMN a **process** is depicted as a graph of flow elements, which are a set of activities, events, gateways, and sequence flows that define finite execution semantics.

A participant may be Responsible for executing a process.

In BPMN, a participant is graphically represented with a pool.

Lanes are used to organize and categorize activities within a pool.

A pool can be positioned vertically or horizontally.

A **participant** represents a specific partner entity or a more general partner role.

Participant

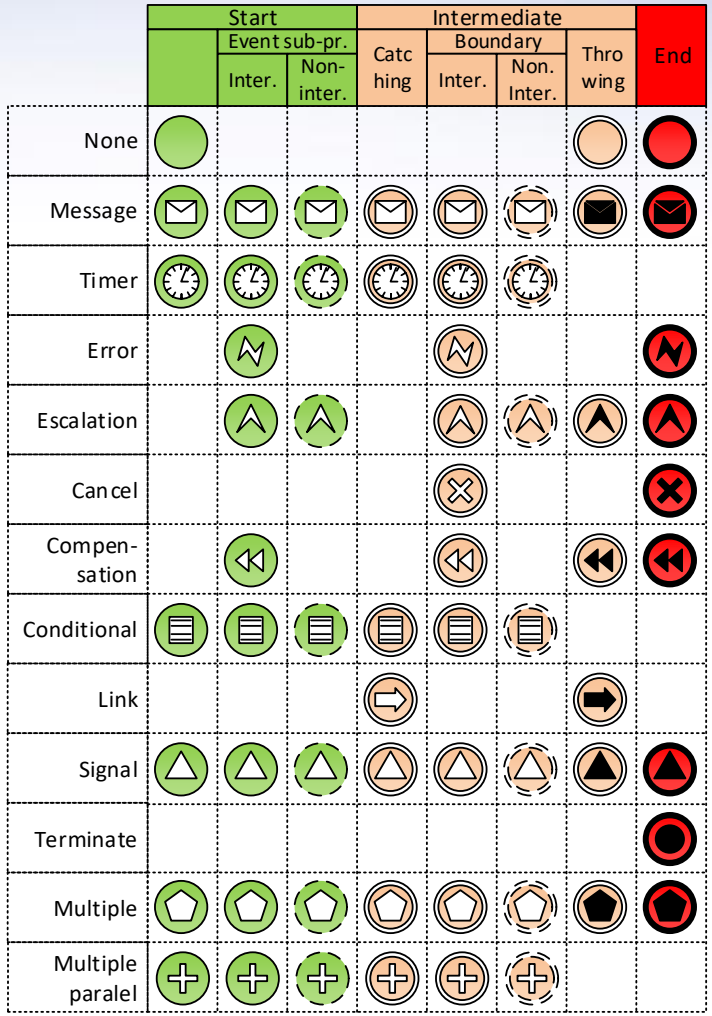
Person or role

Organization

BPMN Events

BPMN 2.0
Page 233

- BPMN 2.0 defines 12 different triggers of events: none, message, timer, error, escalation, cancel, compensation, conditional, link, signal, terminate, multiple and multiple parallel.
- An event may ‘catch’ a trigger or ’throw’ it as a result.
- Events may be of the following types:
  - Start events, which indicate where a process will start.
    - Start events are always of ‘catch’ type.
  - End events, which indicate process’s end state.
    - End events are always of ‘throw’ type.
  - Intermediate events, which indicate where something happens within a process.
    - Intermediate events may be of ‘catch’ or ‘throw’ type.



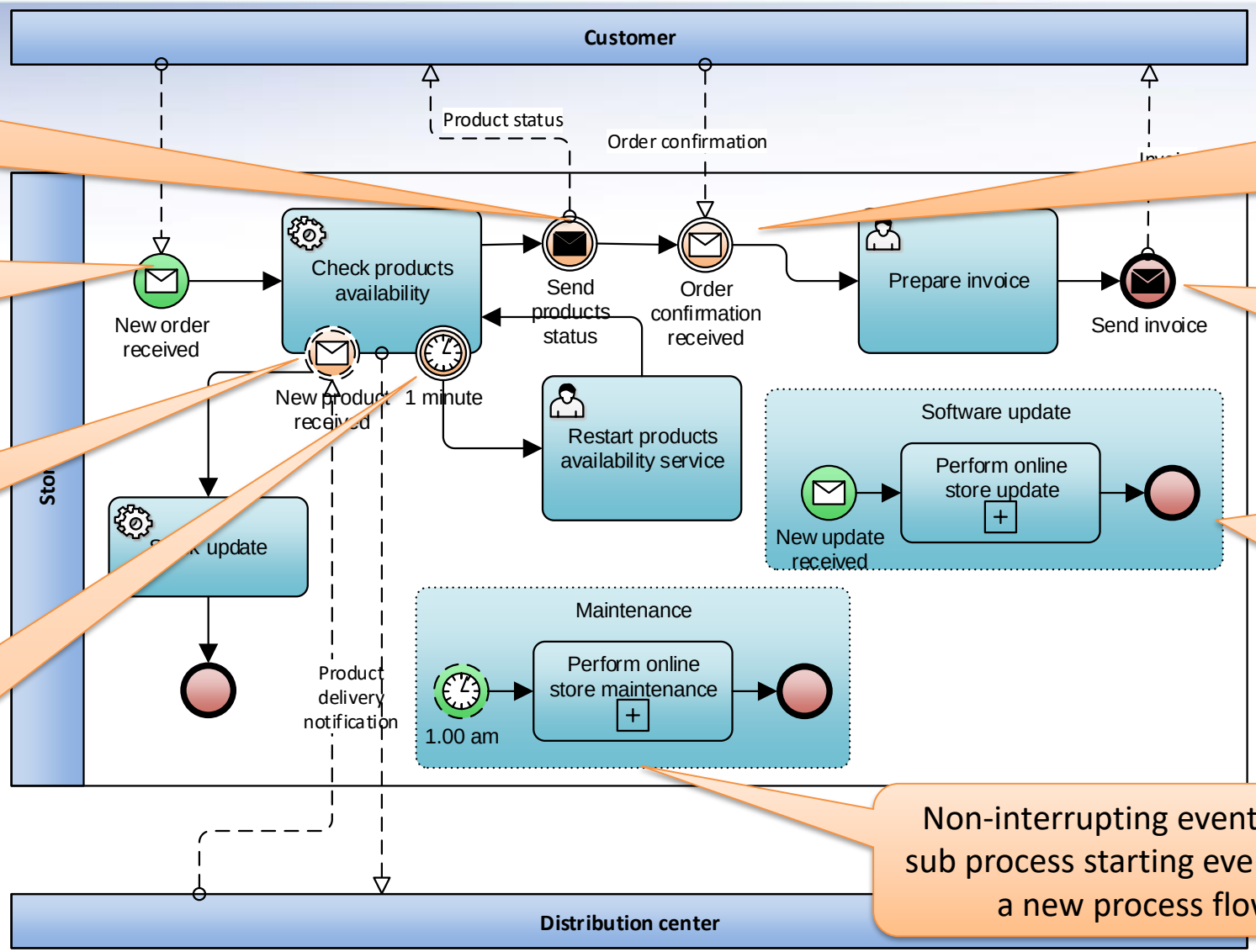
BPMN 'Events by Type' Example

Intermediate throwing events 'produce' a new event.

Start events begin (instantiate) a new process instance

Non-interrupting boundary events start a new process flow.

Interrupting boundary events interrupt existing and start a new process flow.



Intermediate catching events 'wait' for a specific event to continue with process flow.

End events represent final process instance state.

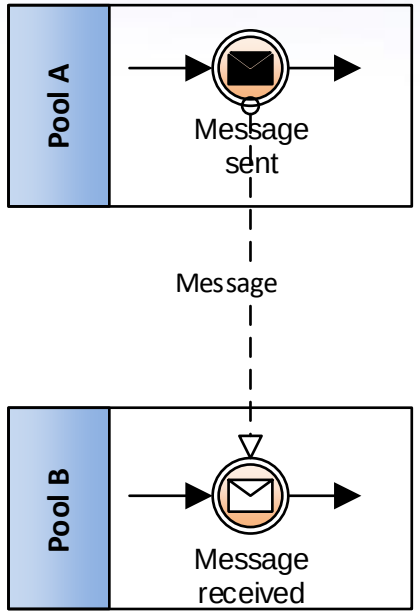
Interrupting event-based sub process starting events start a new process flow.

Non-interrupting event-based sub process starting events start a new process flow.

BPMN Message Events

BPMN 2.0
Page 240-251

- Message events are used for communication between processes.
- Message events send (i.e. throw) or receive (i.e. catch) messages
- Message events can be used for modeling:
  - Asynchronous process start,
  - Waiting,
  - Interrupting current work in case a message receives,
  - Informing other about process results and process end,
  - Start of an asynchronous task, etc.



Common use of message events

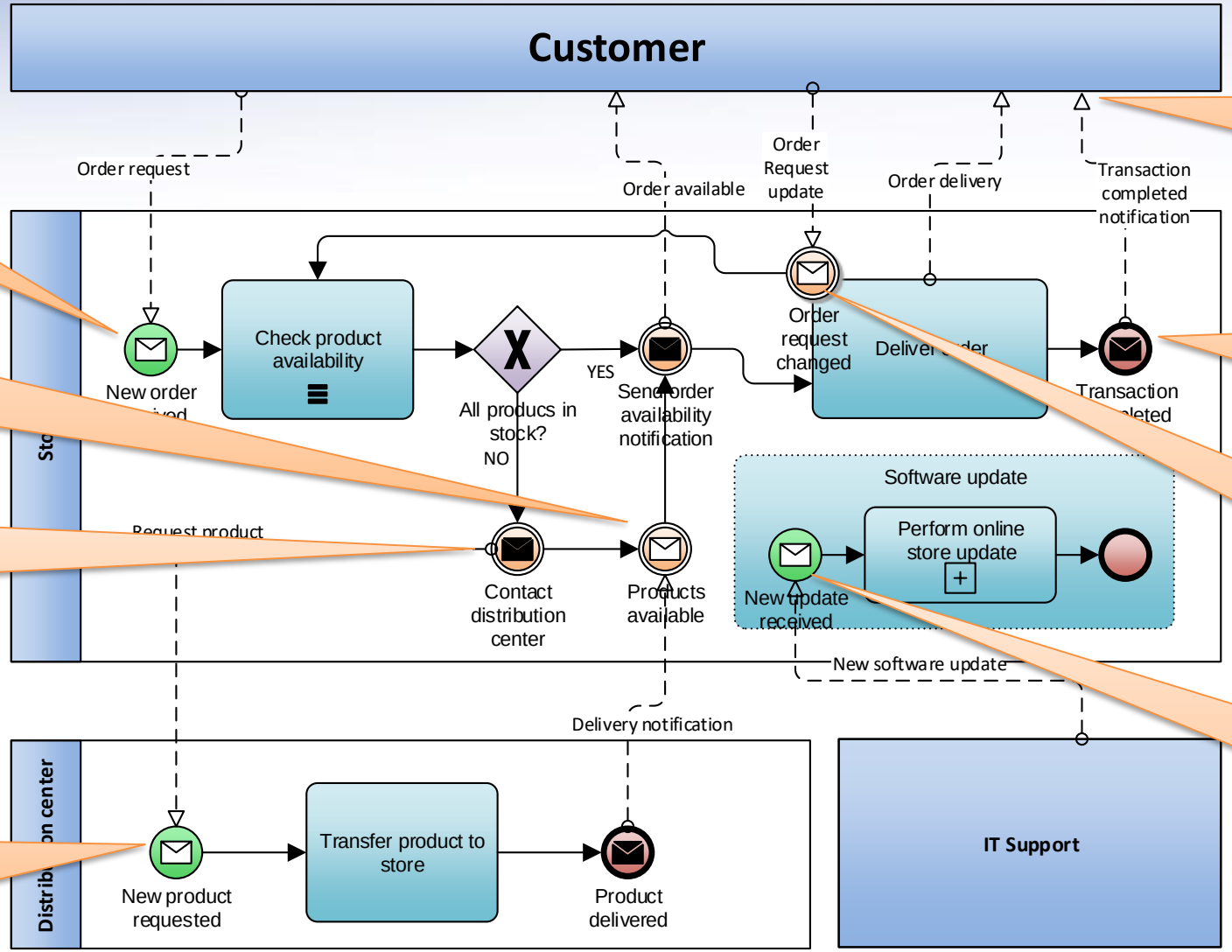
BPMN Message Events Example

New process instance starts with a new order.

Intermediate catching events wait for a message. When message receives, the flow continues.

Intermediate throwing events are source (i.e. generate, trigger) a message.

In case of a white-box pool, the messages are connected to events and activities.



In case of a black-box pool, a message is connected to pool's boundary.

In case of reaching end state, the message is triggered.

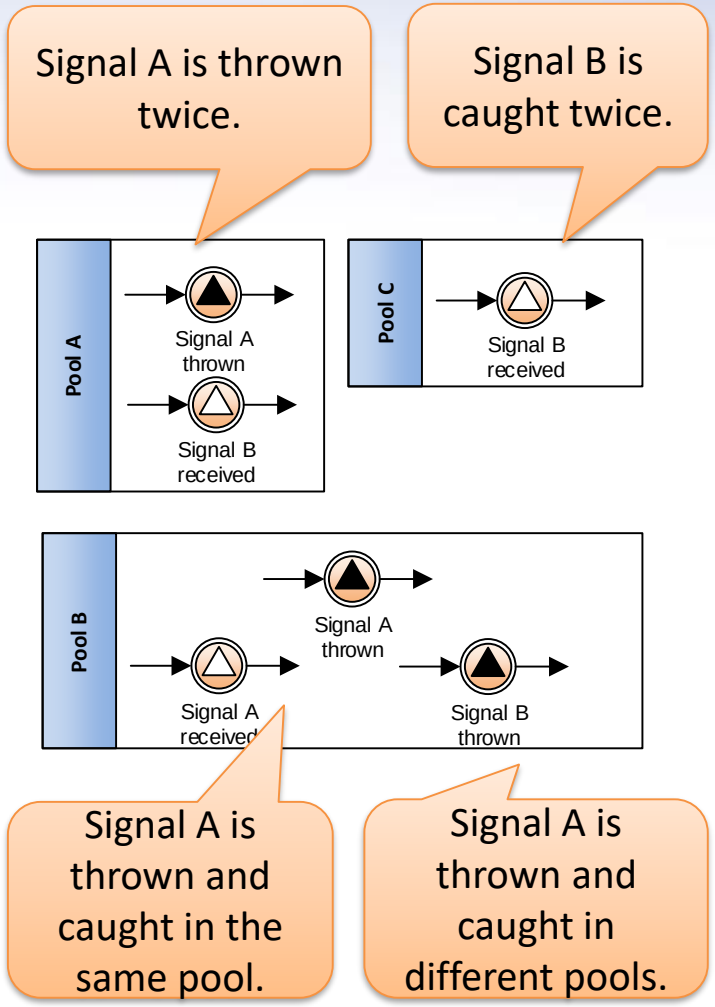
In case a message receives a boundary message event start a new process flow.

This sub-process is started with a message and interrupts the 'main' process flow.

BPMN Signal Events

Bpmn 2.0
Page 241

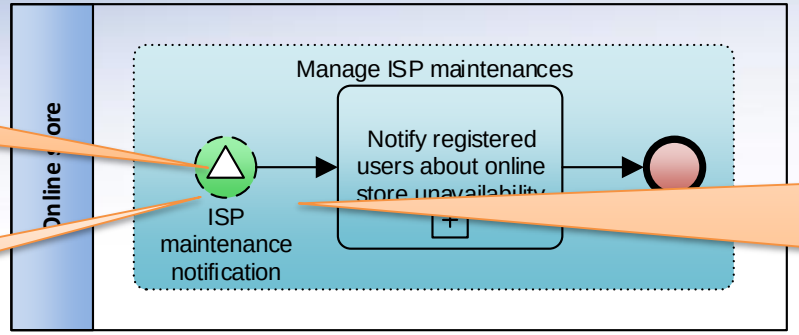
- Signals are generated by throwing signal events and caught by catching signal events.
- A signal differs from a message, since it has no specific target.
  - Throw → catch behavior: message events, error events, escalation events.
  - Publish → subscribe behavior: signal events.
- Same signals can be thrown or received several times.
- Signals have within pool and between pools scope.



BPMN Signal Events Example

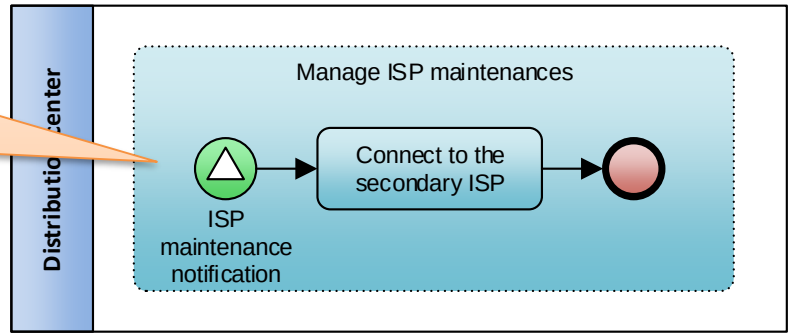
Non interrupting event based sub process signal start event.

Signal catching events are similar to radio receivers.



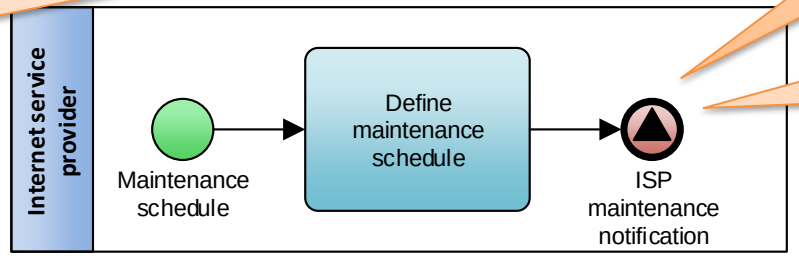
Online store is subscribed to ISP's notifications. In case a 'signal' is received, the work continues, however customers are notified about the potential online store unavailability.

Distribution center is subscribed to ISP's notifications. In case a 'signal' is received, the distribution center connects to the secondary ISP provider.



Signal throwing events are similar to a radio transmitter.

Note that signals are NOT directed to a specific target (i.e. no connections).

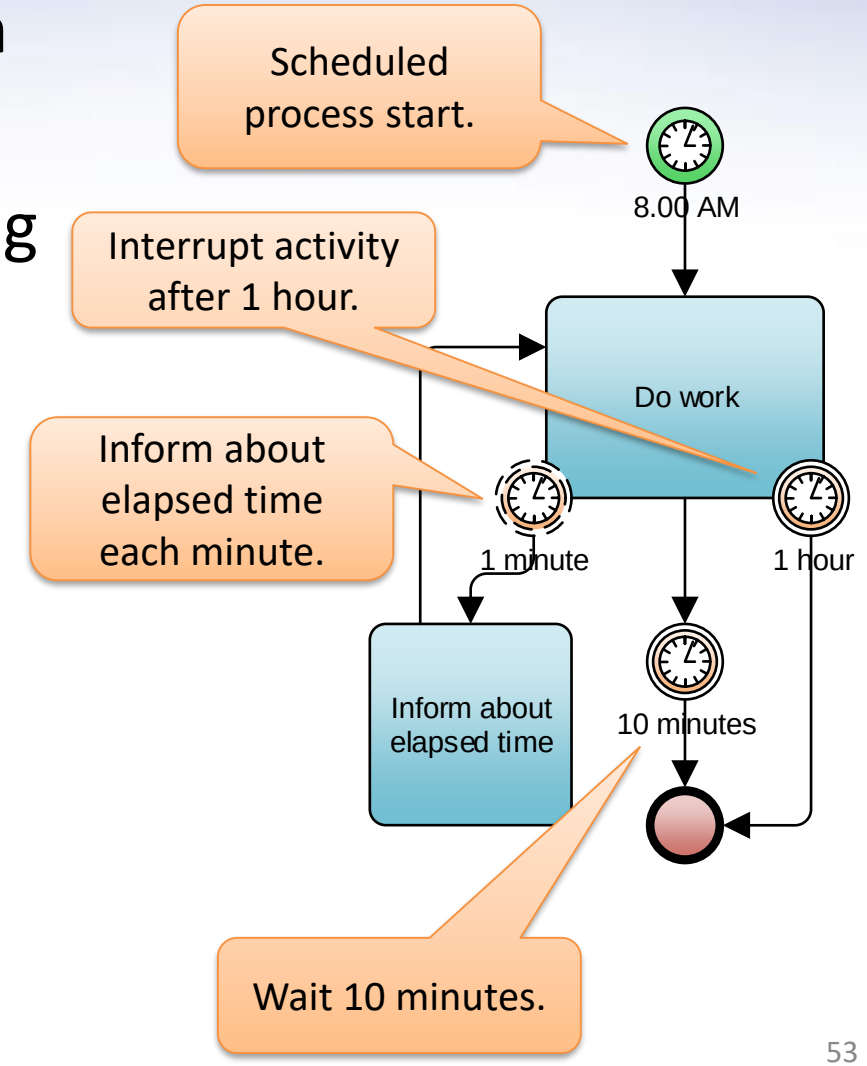


In case an Internet Service Provider (ISP) plans a new maintenance it published maintenance notification.

BPMN Timer Events

BPMN 2.0
Page 253,254

- A timer event occurs by a specific time condition (e.g. specific time, interval, duration)
- Timer events can only react on the corresponding condition (i.e. ‘catch type’).
- Timer events can be used for modeling:
  - Scheduled process start,
  - Waiting,
  - Interrupting current activity after the elapsed time,
  - Informing about the elapsed time,
  - Start of a pre-scheduled job, etc.



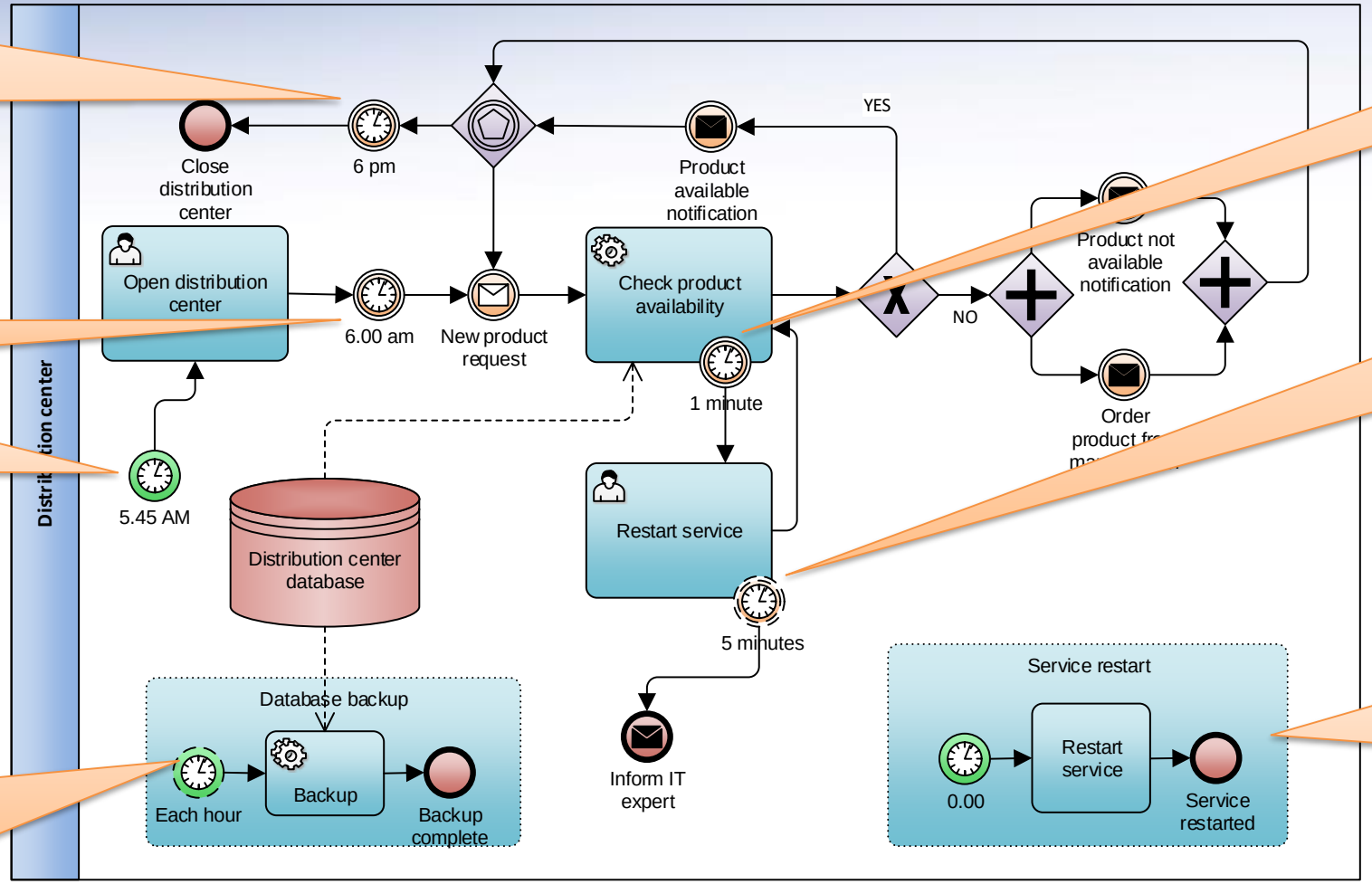
BPMN Timer Events Example

At 6.00 pm, the distribution center stops receiving requests.

Wait until 6.am.

At 5.45 AM the distribution center starts working.

Each hour, the database is backed-up where the current work continues



In case the service checks for products more than a minute, it is restarted.

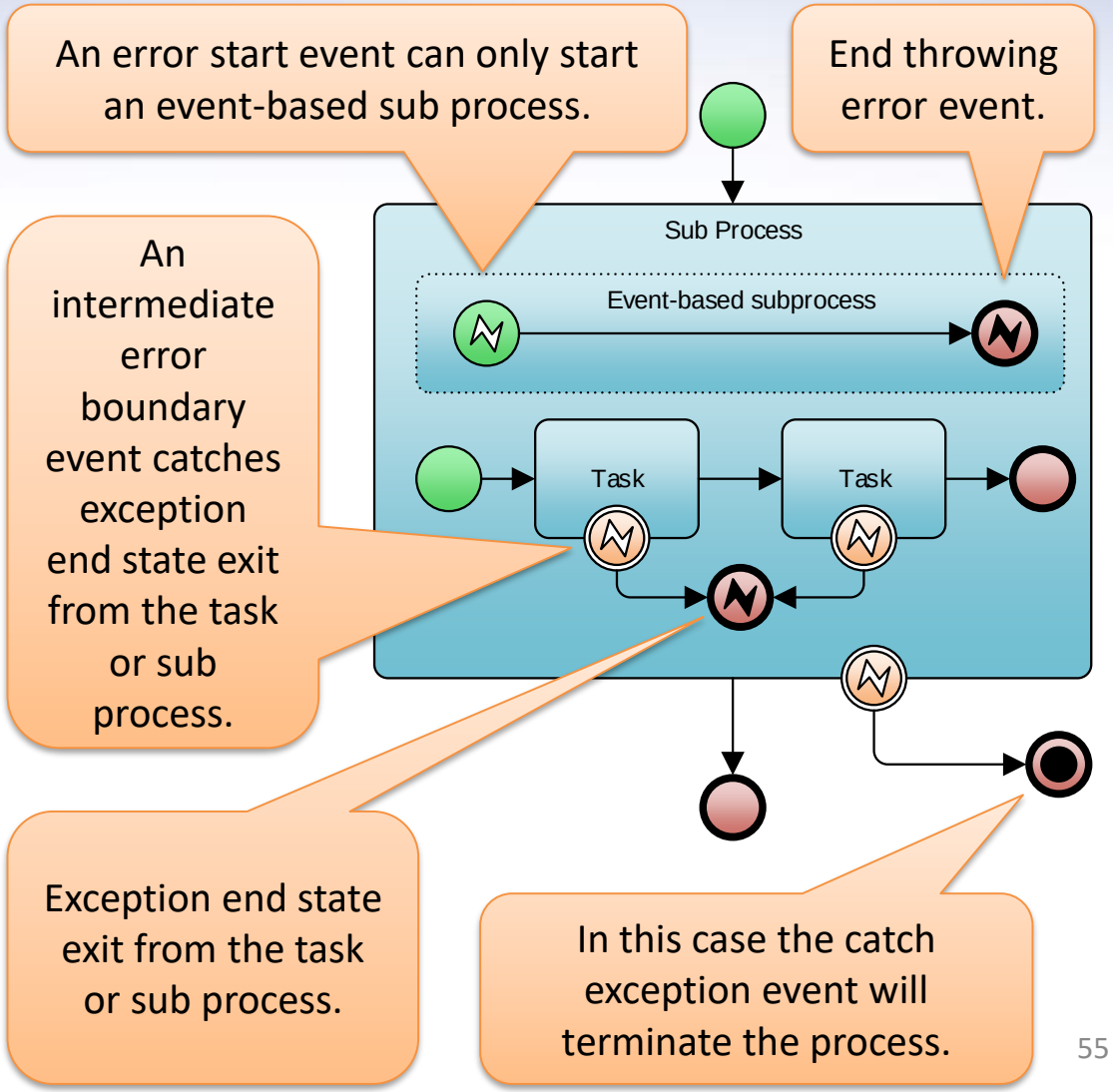
In case restart takes more than 5 minutes, IT expert is contacted.

At 0.00, the distribution center service is restarted.

BPMN Error Events

BPMN 2.0
Page 247,255

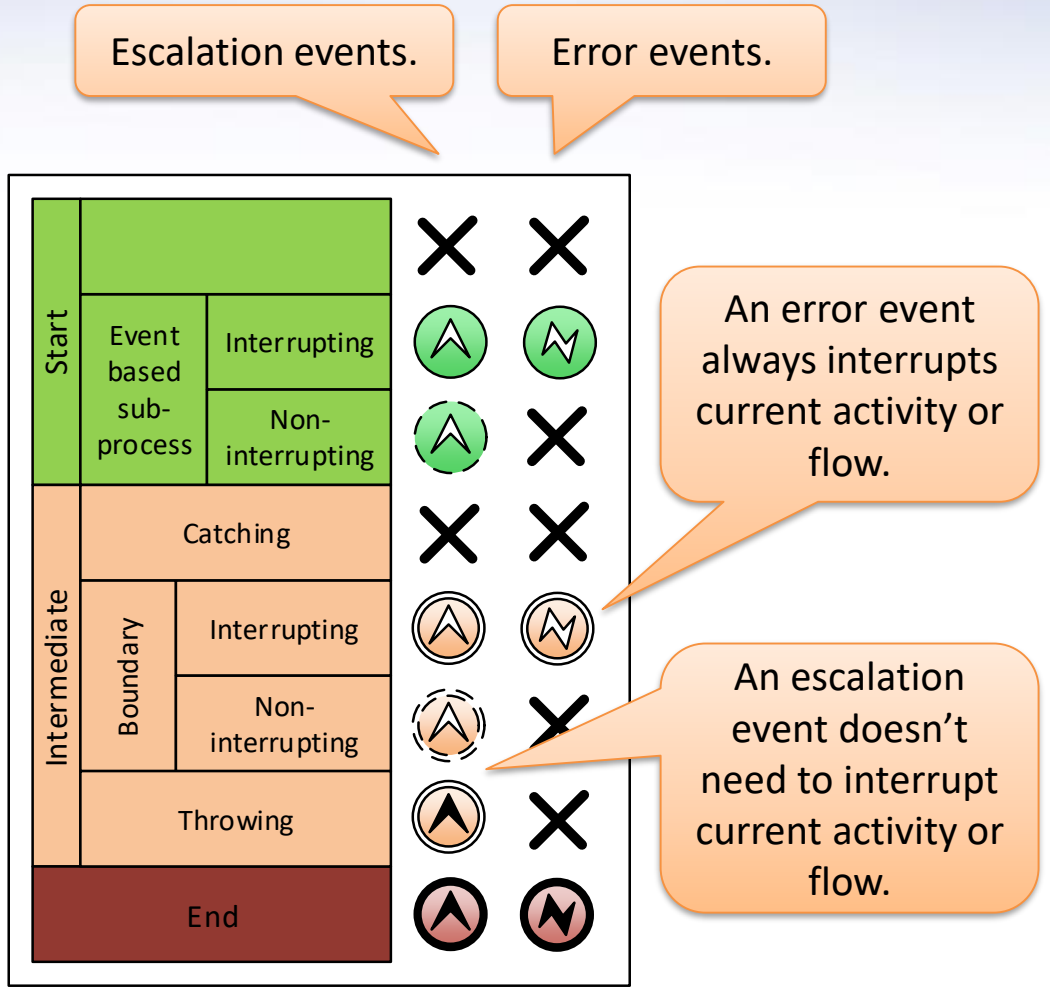
- An error (end) event is thrown when an activity (i.e. task, sub process) does not end successfully.
  - All currently active threads in the particular sub-process are terminated as a result.
- An exceptional end state can be caught by an intermediate error event attached to the boundary of an activity.
  - An activity can have several boundary error events attached (each representing distinct end exception states).
  - Note that an error event always interrupts the activity to which it is attached.



BPMN Escalation Events

BPMN 2.0
Page 247,255

- Escalations events are useful for the communication between sub-processes and processes.
  - Escalation events enable “throw-catch” behavior (i.E. In case an escalation happens, the next higher level of responsibility is involved).
- In concept, escalations are similar to errors, but are generally less critical.
  - In contrast to error events escalation events can be non-interrupting.
- Escalations events usually represent a situation where human intervention is required.

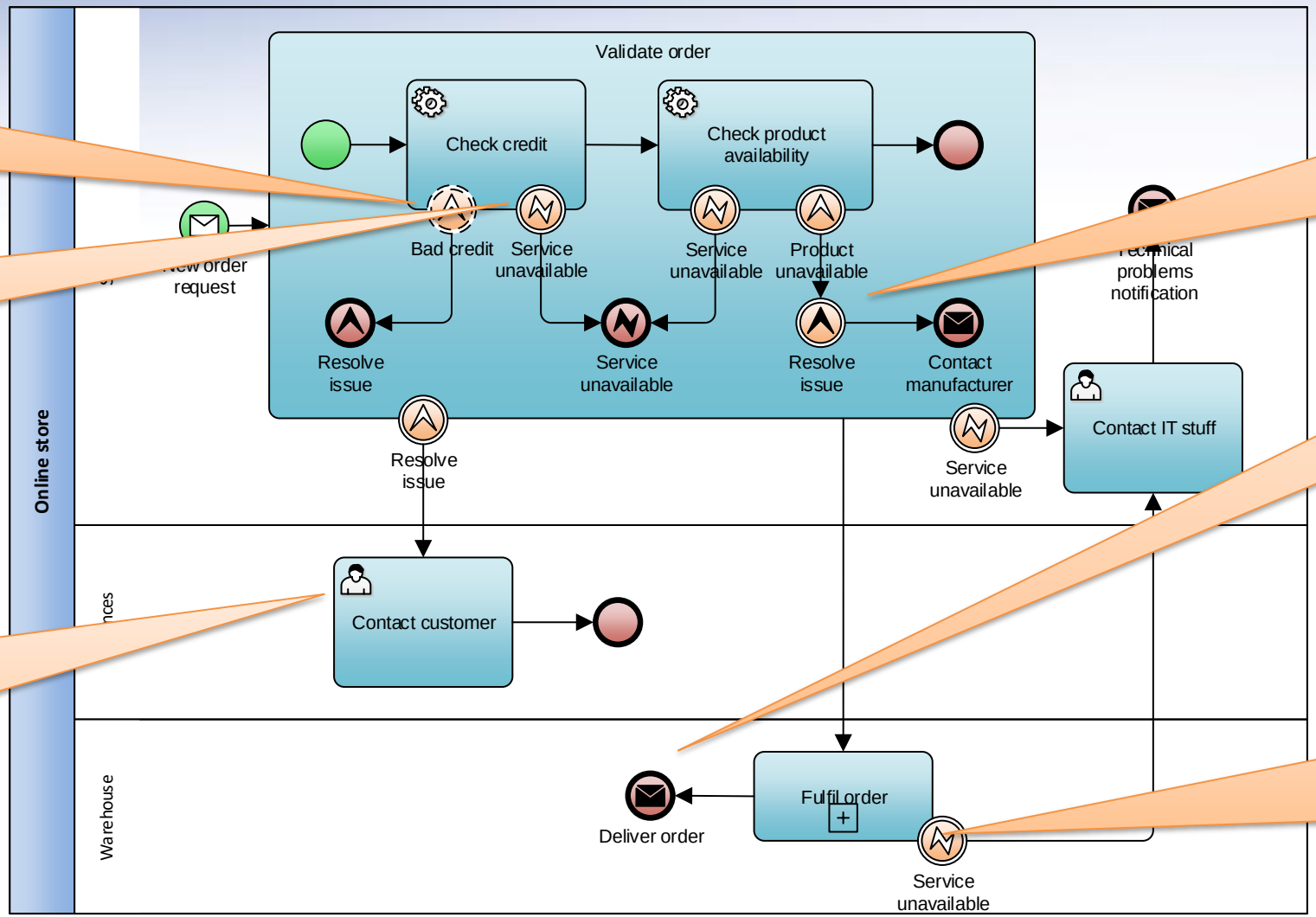


BPMN Error and Escalation Events Example

In case of bad credit escalation is thrown but the sub process continues.

In case of unavailability the sub process cannot continue normally.

Escalation is commonly manually resolved by another role or department.



Intermediate escalation events can throw escalations and then continue.

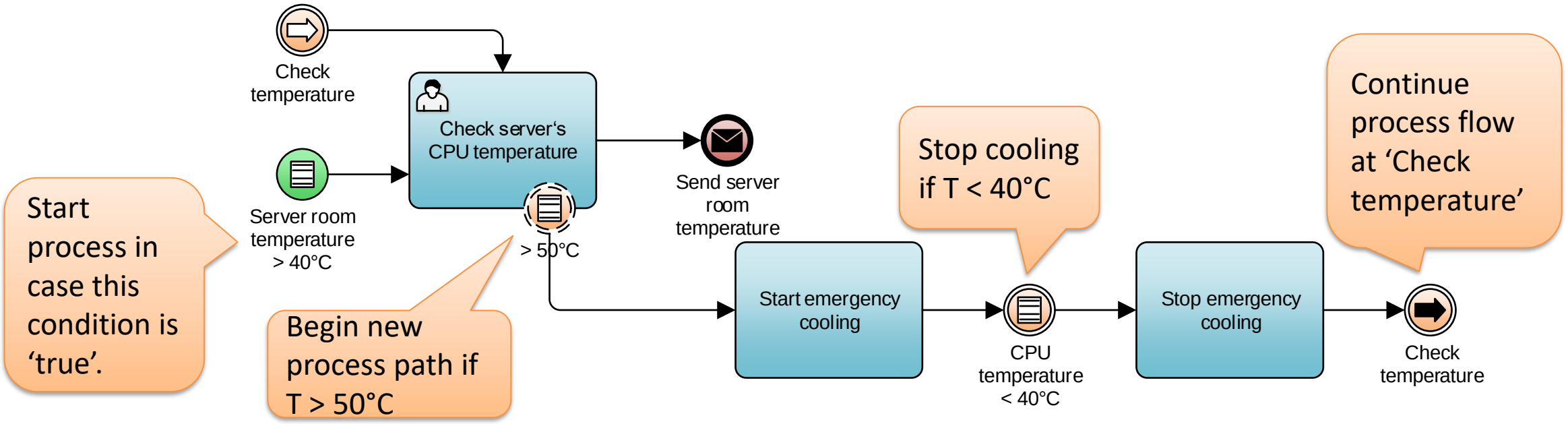
This end event represents successful process end.

In case an error event is caught the corresponding activity is interrupted.

BPMN Conditional and Link Events

BPMN 2.0
Page 267-270

- A conditional event signifies a continuously monitored condition.
- When condition becomes ‘true’ the event is triggered.
- Link events pair (i.e. throw – catch) is a visual shortcut for a sequence flow.
  - Link events may not be used between pools or parent-child process levels.

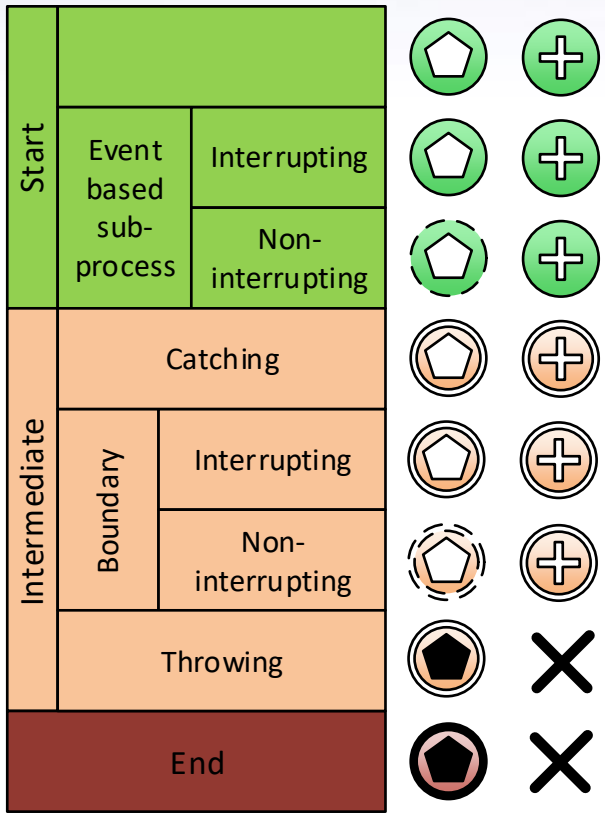


BPMN Multiple Events

BPMN 2.0
Page 271-273

Multiple (OR)

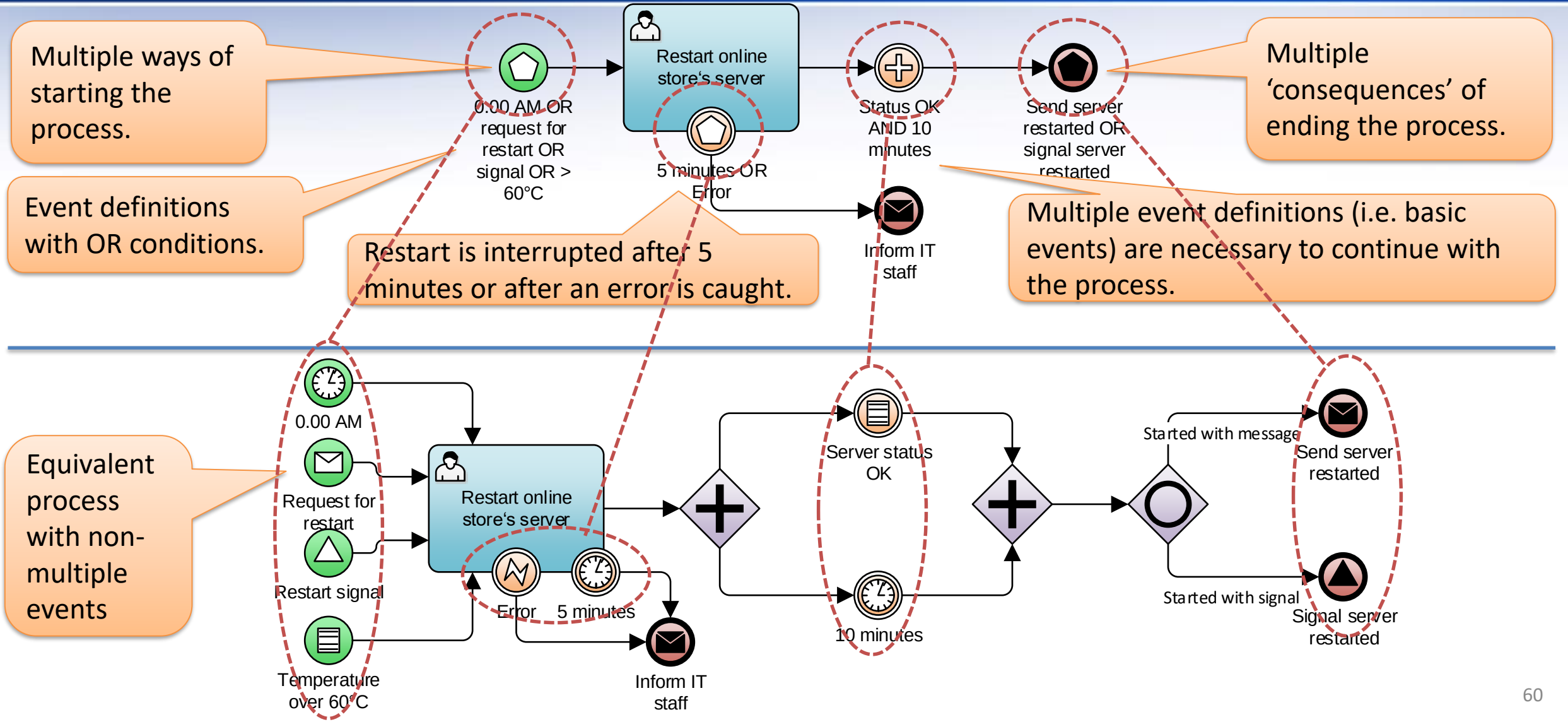
- Start
  - There are multiple ways of starting the process.
- Intermediate / boundary
  - Catch / boundary: only one event definition is required to catch the trigger.
  - Throw: all of the event definitions are considered and the subclasses will define which results apply.
- End
  - There are multiple consequences of ending the process.



Parallel multiple (AND)

- Start
  - There are multiple triggers required to start the process.
- Intermediate / boundary
  - All of the defined event definitions are required to trigger the event.
- Parallel multiple events are only of type ‘receive’ (catch).

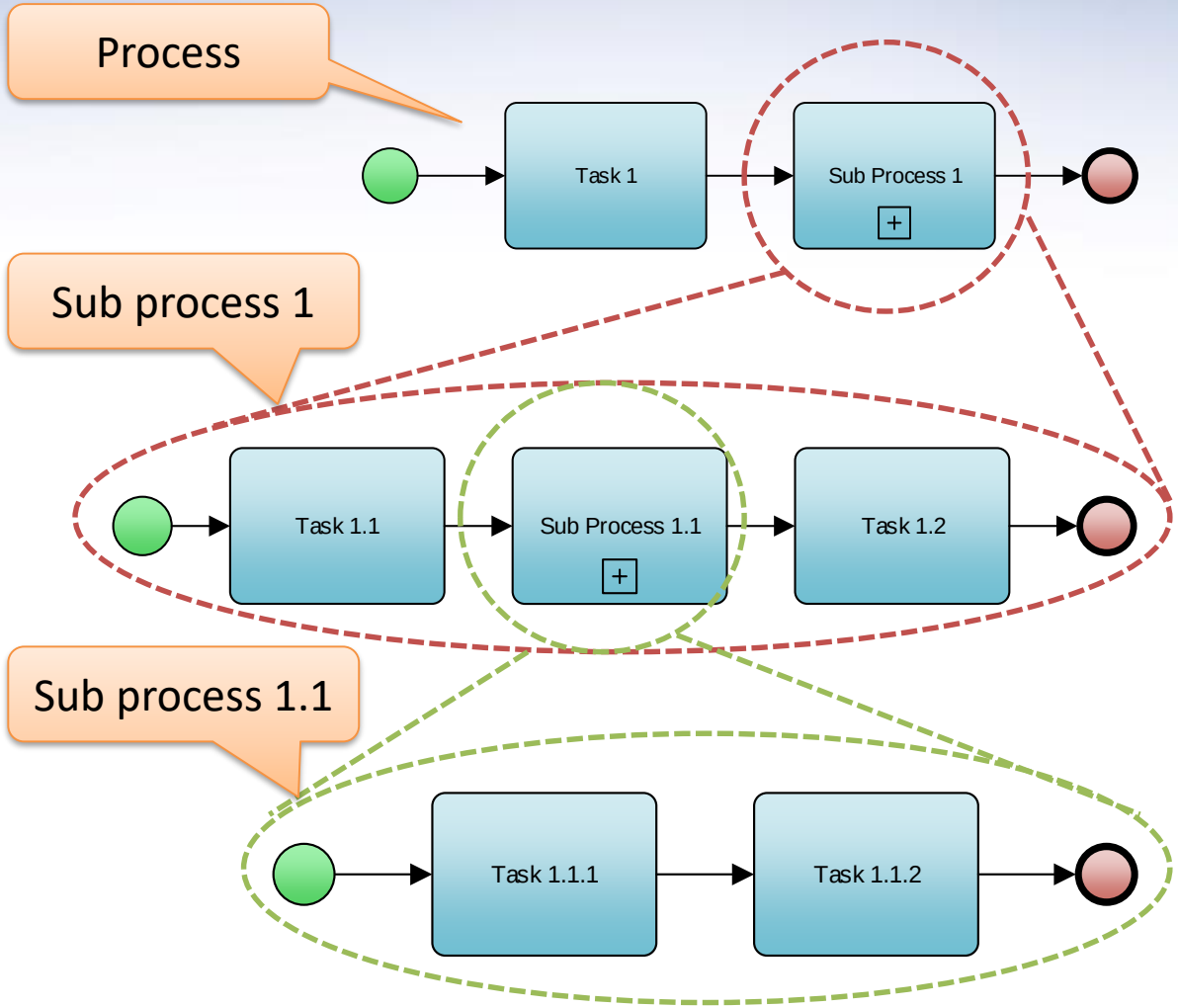
Multiple Events Example



BPMN Activities

BPMN 2.0
Page 151

- Activities represent points in a process flow where work is performed. They are the executable (i.e. manual, automatic) elements of a BPMN process.
- The types of activities that are a part of a process are:
  - Task (atomic activity)
  - Sub-process (compound activity)
  - Call activity (allows the inclusion of re-usable Tasks and Processes in the diagram),
- A process is not a specific graphical object. Instead, it is a set of graphical objects.



Types of BPMN Activities

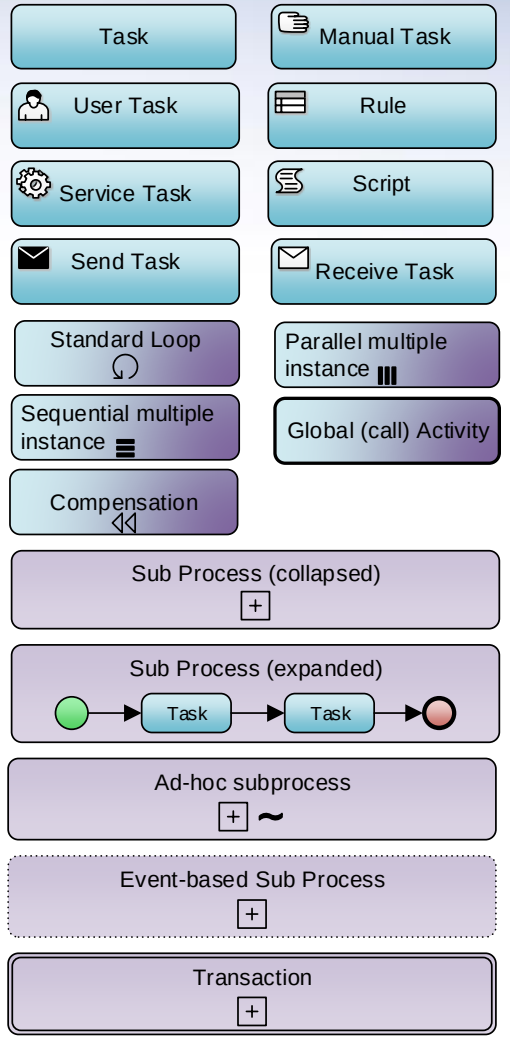
BPMN 2.0
Page 151-152

- Special activities (i.e. sub-processes and tasks)
  - Behavior: loop, multiple instance (sequential, parallel), compensation, call activity.
- Special sub-processes (i.e. compound activities)
  - View: collapsed / expanded
  - Type: basic (i.e. embedded), event-based sub process, transaction, ad-hoc.
- Special tasks (i.e. atomic activities)
  - Types: manual, user, service, send, receive, rule, script.

Special task types.

Special task and sub-process types.

Special sub-process types.



BPMN Task Types

BPMN 2.0
Page 158-164

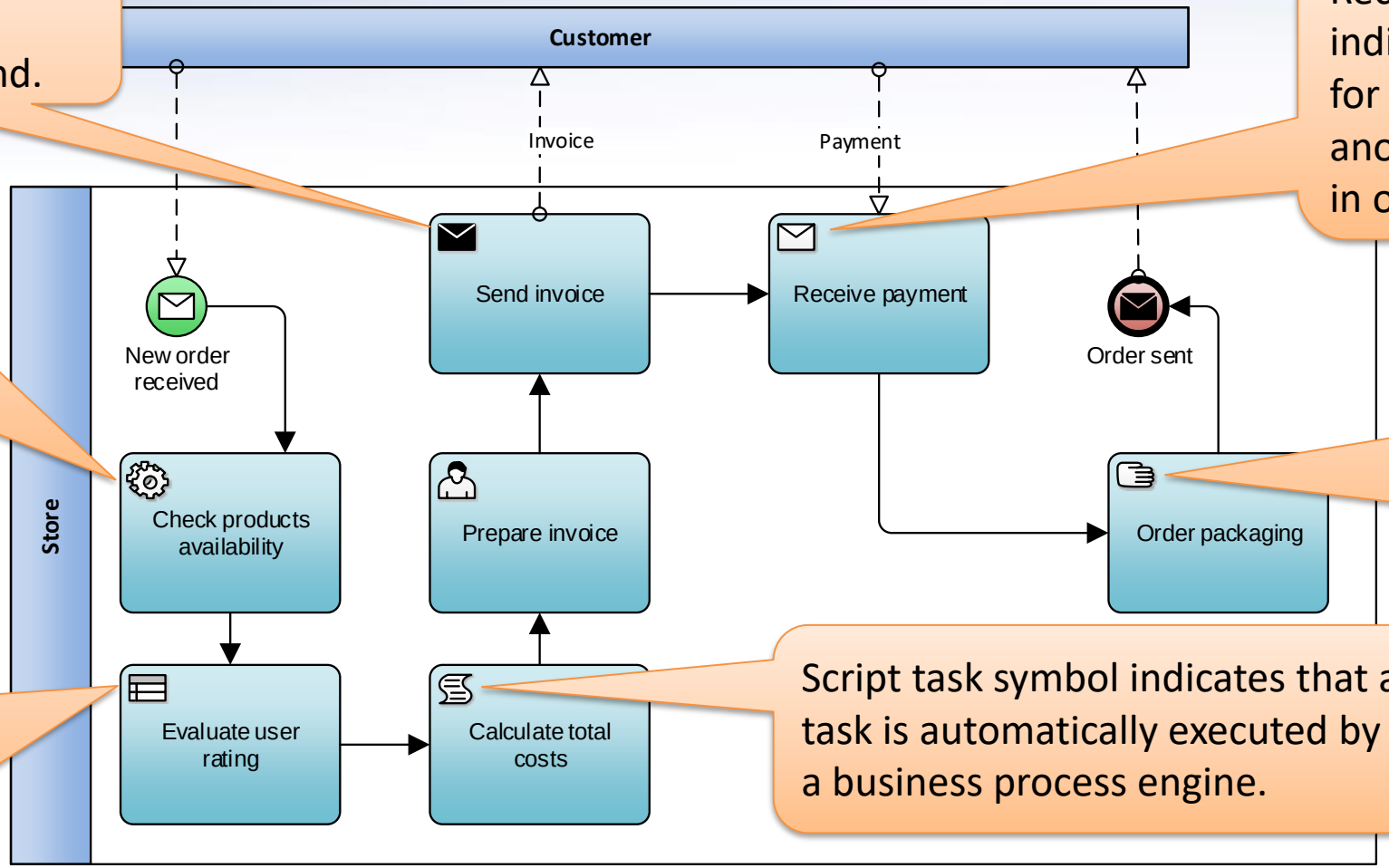
- A **service task** is a task that uses some sort of service, which could be a web service or an automated application.
- A **send task** is a simple task that is designed to send a message to an external participant (relative to the process). Once the message has been sent, the task is completed.
- A **receive task** is a simple task that is designed to wait for a message to arrive from an external participant. Once the message has been received, the task is completed.
- A **user task** is a typical “workflow” task where a human performer performs the task with the assistance of a software application and is scheduled through a task list manager of some sort.
- A **manual task** is a task that is expected to be performed without the aid of any business process execution engine or any application.
- A **business rule task** provides a mechanism for the process to provide input to a business rules engine and to get the output of calculations that the business rules engine might provide.
- A **script task** is executed by a business process engine. The modeler or implementer defines a script in a language that the engine can interpret. When the task is ready to start, the engine will execute the script. When the script is completed, the task will also be completed.

BPMN Task Types Example

Send task symbol indicates that a task sends a message prior to its end.

Service task symbol indicates that an (external) IT solution (service) performs a task.

Business rule symbol indicates that a business rule engine evaluates a task.



Receive task symbol indicates that a task waits for a message from another participant (pool) in order to be performed.

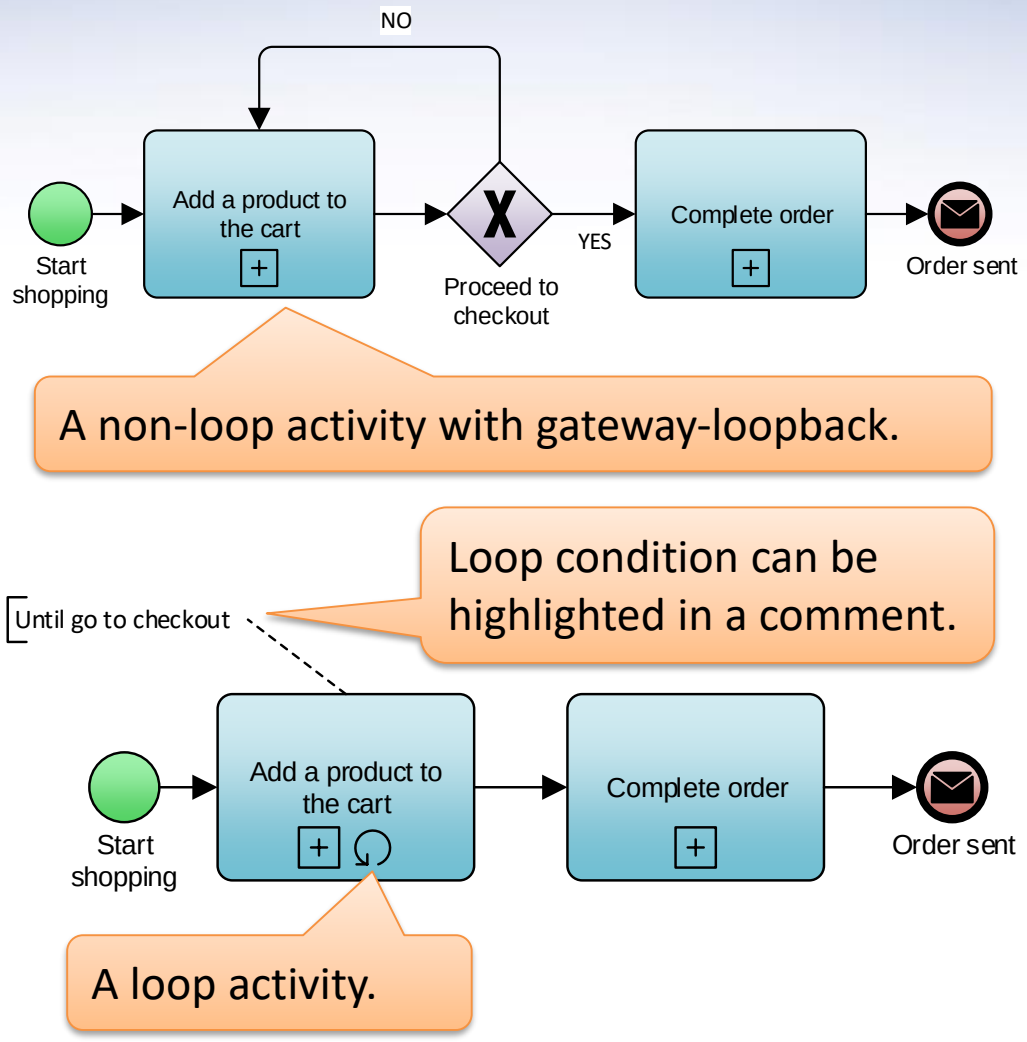
Manual task symbol indicates that a task is performed without IT support.

Script task symbol indicates that a task is automatically executed by a business process engine.

BPMN Activities: Loops

BPMN 2.0
Page 189-190

- Loop activity is a repeating activity with sequential iteration.
  - You cannot start second loop iteration until finishing the first one.
- Loop activity is primary evaluated at the end.
  - 'Do-while' type of loop.
  - When the first iteration starts, the number of iterations is unknown.
- The activity will loop as long as loop condition is true.
- A loop marker can be attached to sub-processes and tasks.

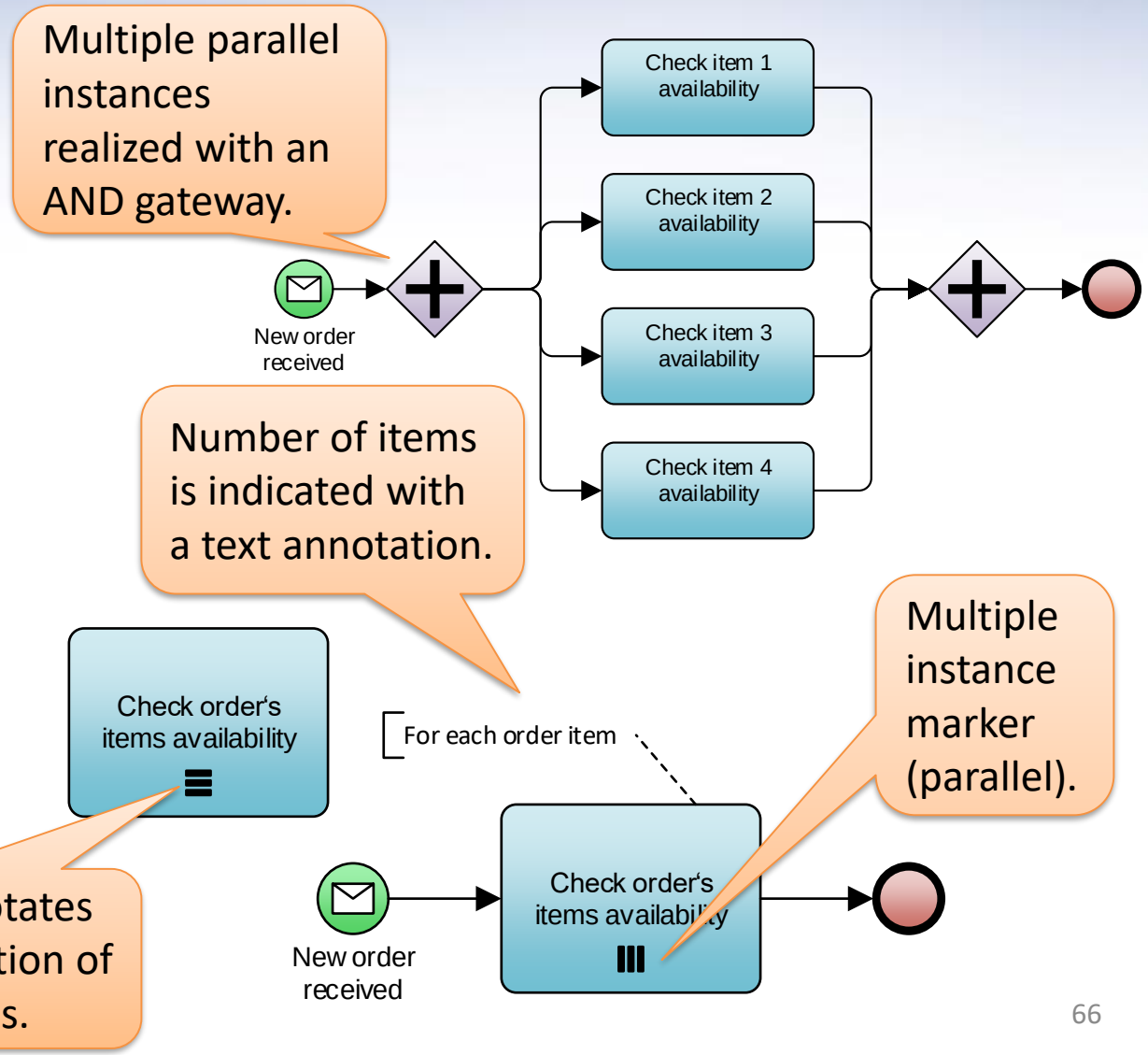


(Silver 2011)

BPMN Activities: Multi Instance

BPMN 2.0
Page 432, 433

- The multi instance activity represents an activity with several activity instances.
- The instances may execute in parallel or may be sequential.
- Multi instance activity makes sense when the process instance data contains some kind of collection (e.g. items in an order).
- The number of iterations is known in advance
  - ‘For each’ type of loop.
- Can be applied to tasks and sub-processes.

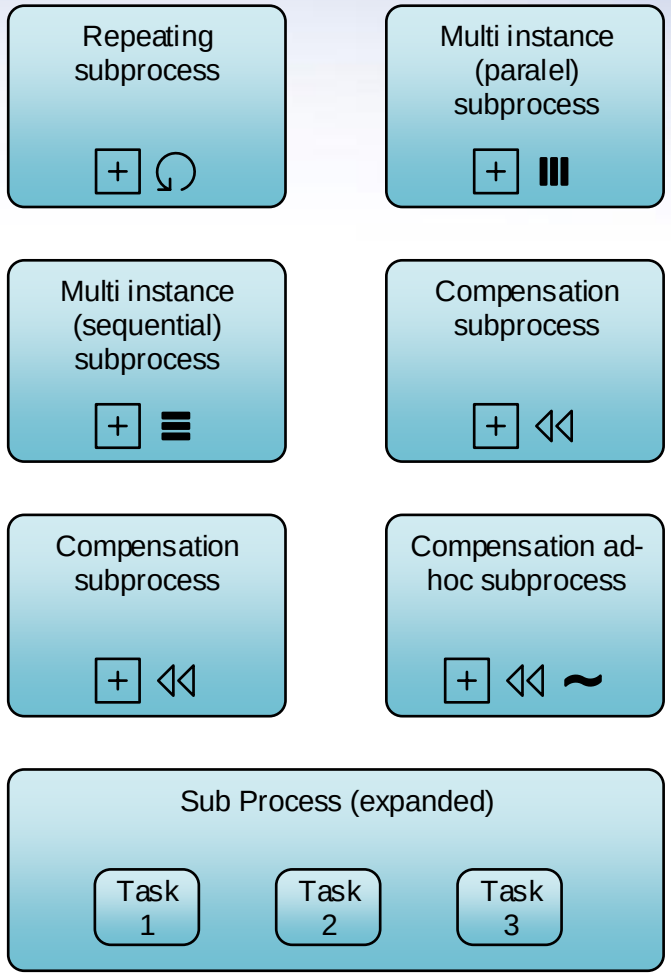


BPMN Sub-processes

BPMN 2.0
Page 174-175

- A sub-process is a graphical object within a process, which can be “opened up” to show a lower-level process.
- Sub-processes define a contextual scope that can be used for :
  - attribute visibility,
  - transactional scope,
  - for the handling of exceptions or events,
  - for compensation.
- A sub process can be in collapsed (i.e. hidden details) or expanded (i.e. visible details) view.

Collapsed sub-process markers.

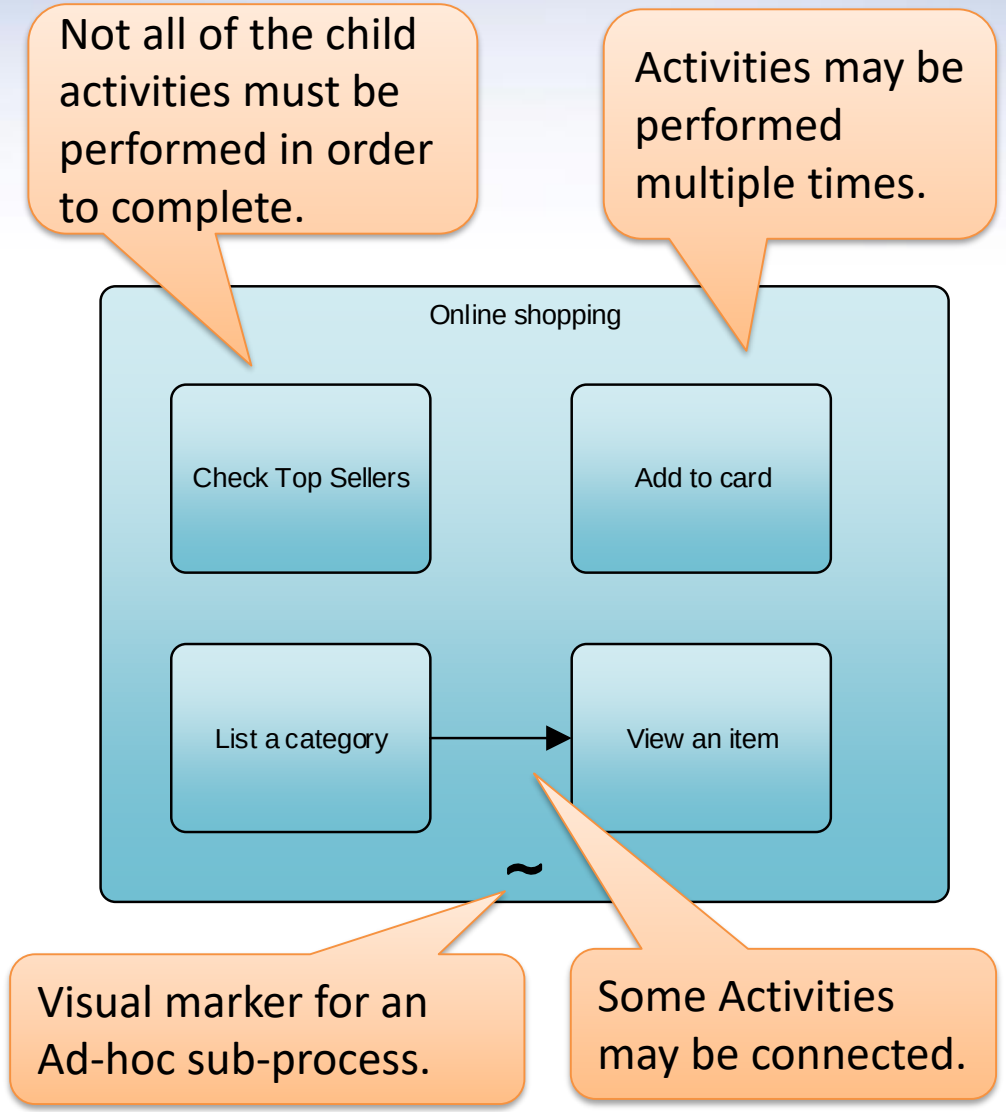


An expanded sub-process with three parallel tasks.

BPMN Sub-Processes: Ad-hoc

BPMN 2.0
Page 180-183

- A specialized type of sub-process in which activities have no required sequence relationships.
  - The sequence and number of performances for the activities is determined by their performers.
  - Some sequence and data dependencies can be defined.
- During the execution of the ad-hoc sub-process, any one or more of the activities may be active.
- An ad-hoc sub process is complete when a performed declares it to be complete.

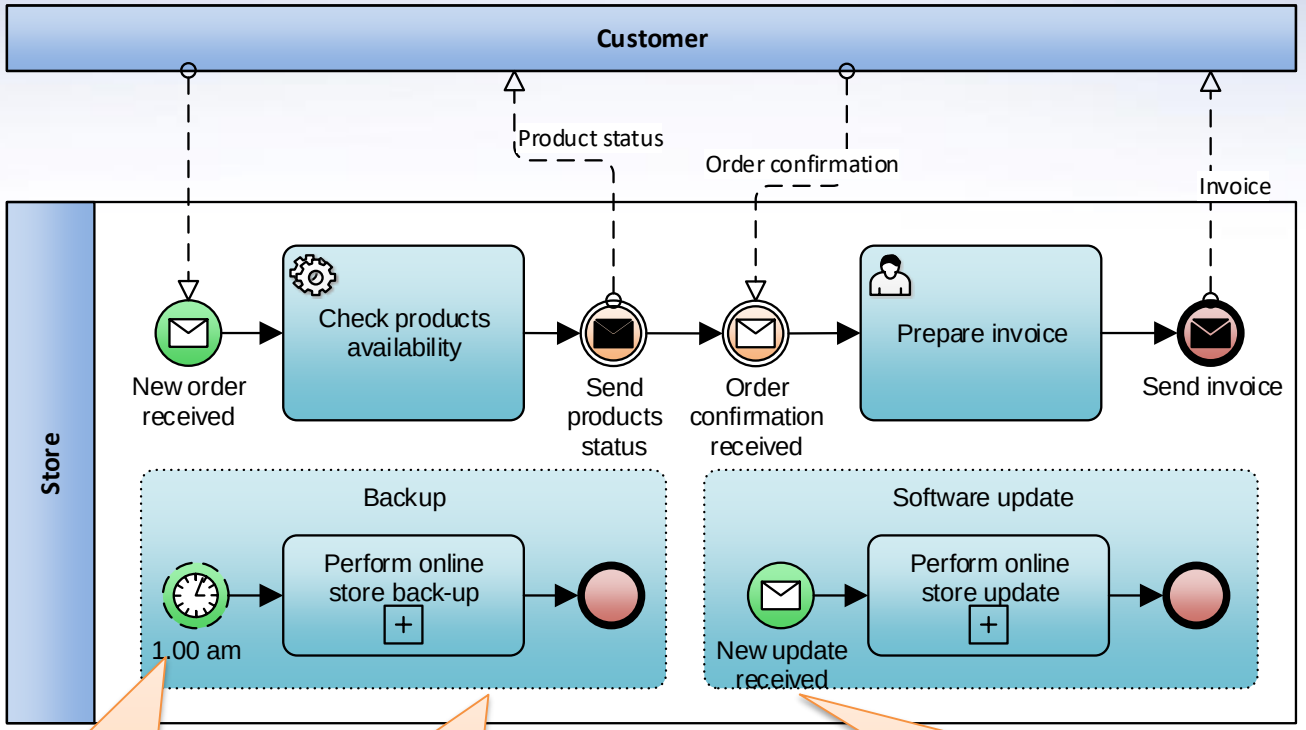


(Silver 2011)

BPMN Sub-processes: Event-based

BPMN 2.0
Page 176-178

- An event sub-process is not part of the normal process flow.
  - It has no incoming or outgoing sequence flows.
- When an event sub-process is triggered the parent process:
  - Can be interrupted,
  - Can continue its work.
- An event sub-process may occur zero or many times.



Event sub-process non-interrupting timer event – the main process flow continues.

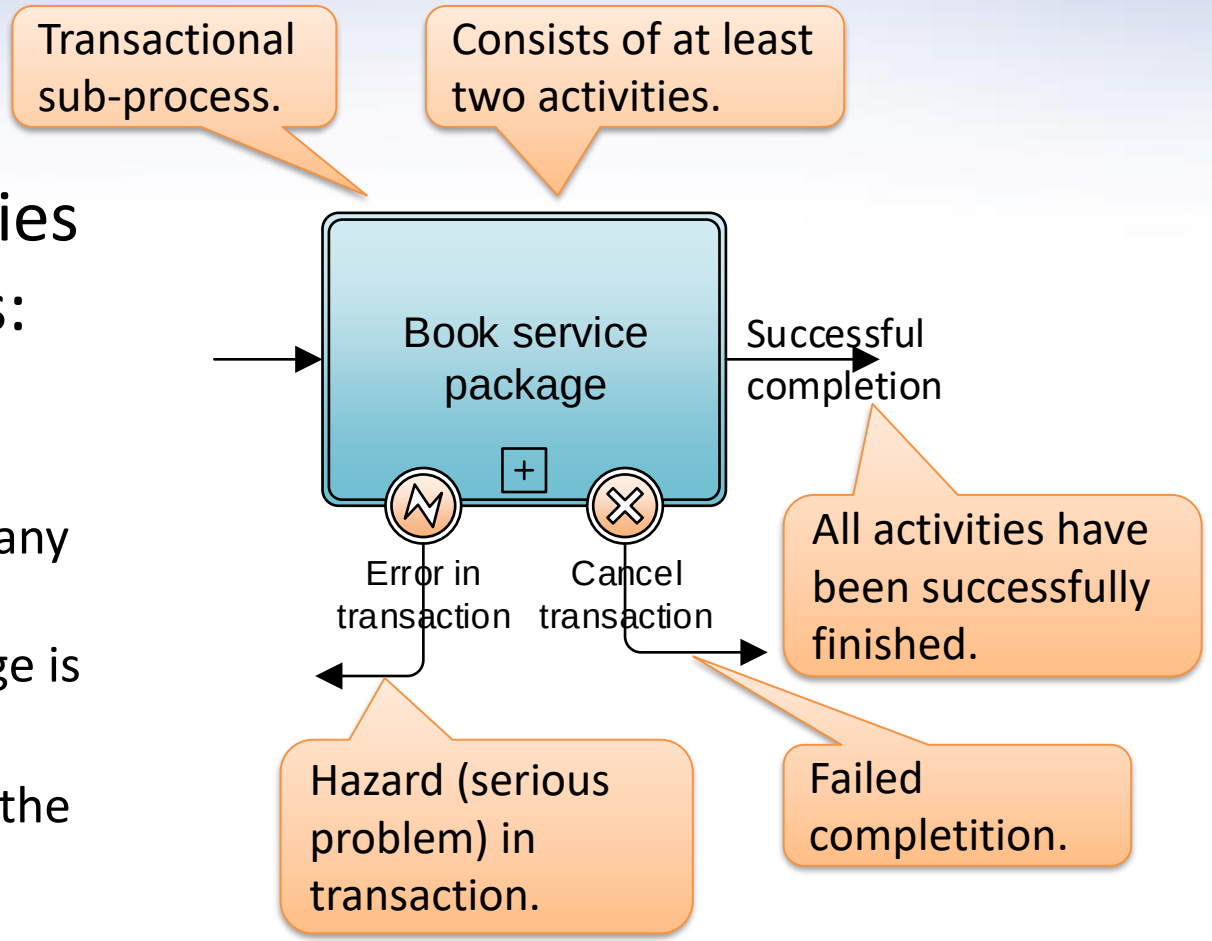
Event sub-process is bordered with a dotted line.

Event sub-process interrupting message event – it interrupts main process flow.

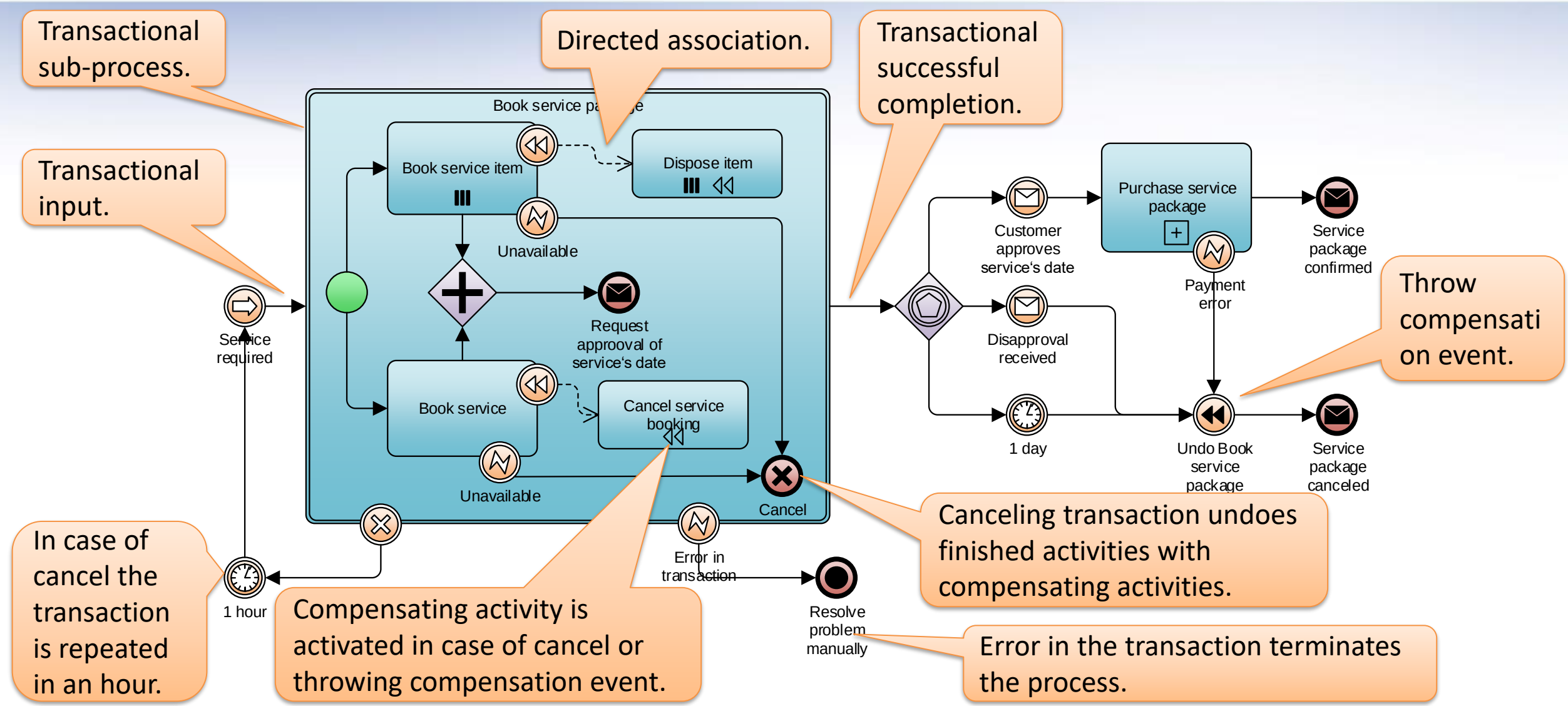
BPMN Sub-Processes: Transactions

BPMN 2.0
Page 178-180

- BPMN provides built-in support for business transactions.
- A transaction consists of several activities and might result in following outcomes:
  - Successful completion. In this case, all tasks in a transaction are completed successfully.
  - Failed completion (cancel). This scenario occurs if any of the pre-determined criteria of failure of the transaction is satisfied or in case an ‘abort’ message is received from outside of the transaction.
  - Hazard (exception). In this case any of the tasks in the transaction end up not being executed or compensated.



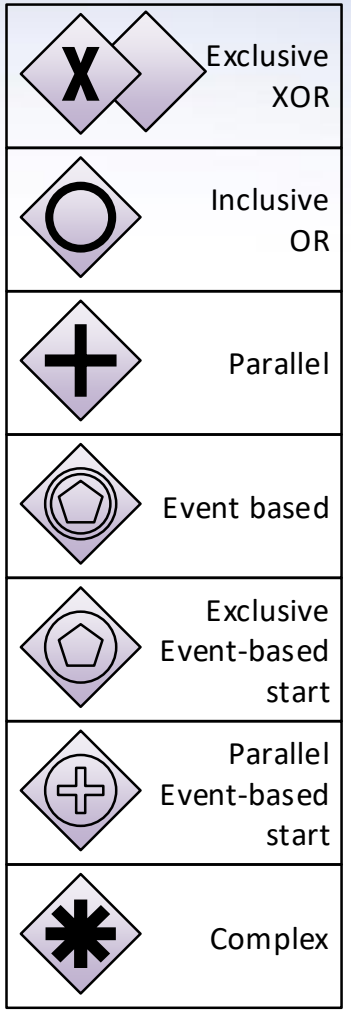
BPMN Sub-processes: Transactions Example



BPMN Gateways

BPMN 2.0
Page 287,288

- Gateways are used to control how sequence flows interact as they converge and diverge within a process.
  - Decisions/branching (exclusive, inclusive, and complex), merging, forking, and joining.
- Gateways are capable of consuming or generating additional tokens (i.e. AND gateway).
- Gateways have zero effect on the operational measures of the process being executed
  - E.g. process duration, and costs.
- A single gateway could have multiple input and multiple output flows.

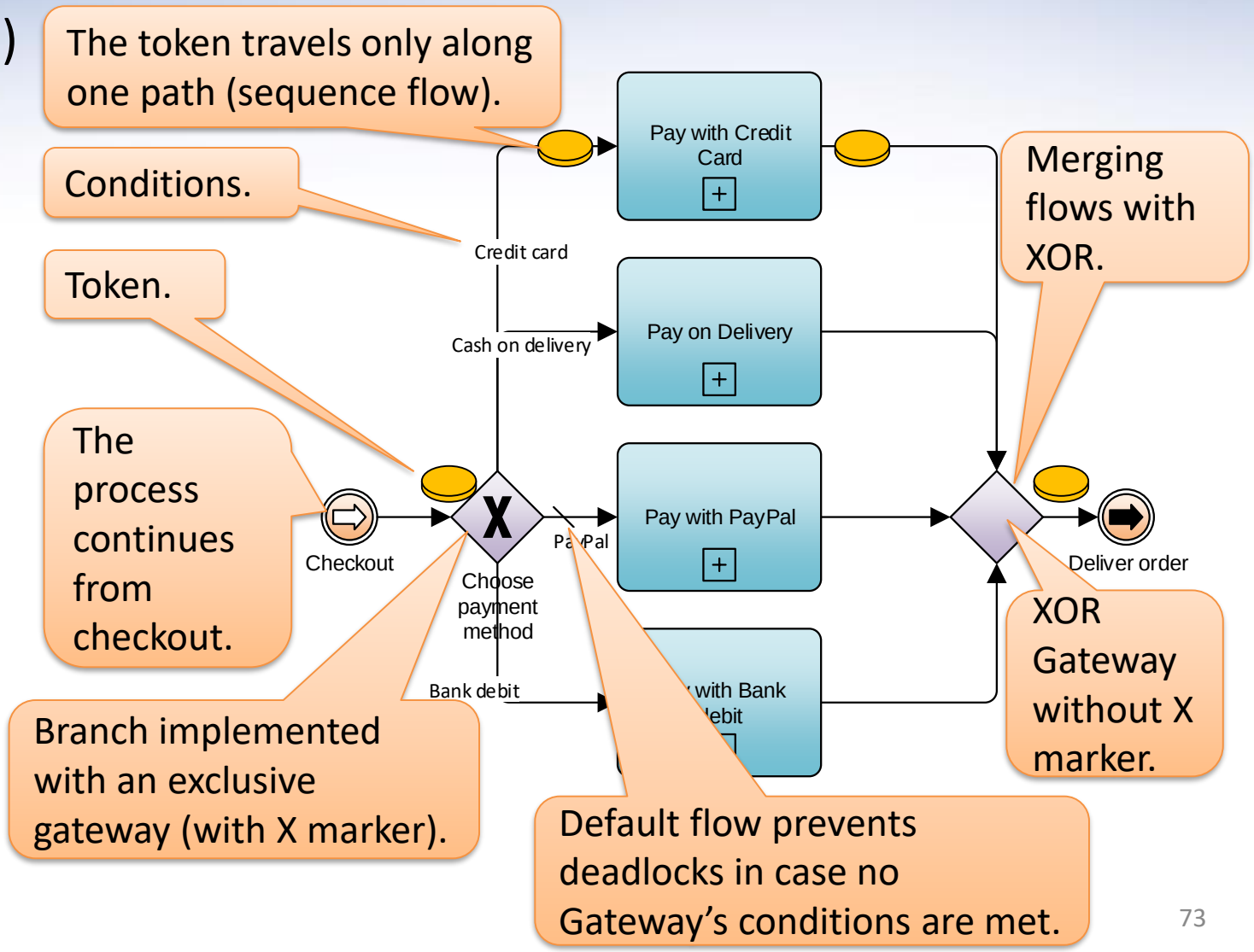


BPMN 2.0 Gateways

BPMN Gateways: Exclusive (XOR)

BPMN 2.0
Page 290-292

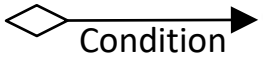
- A diverging exclusive gateway (XOR) is used to create alternative paths (i.e. decisions) within a process flow.
  - A decision can be thought of as a question that is asked at a particular point in the process.
  - The question has a defined set of alternative answers.
- The exclusive gateway may be represented with or without an x marker.



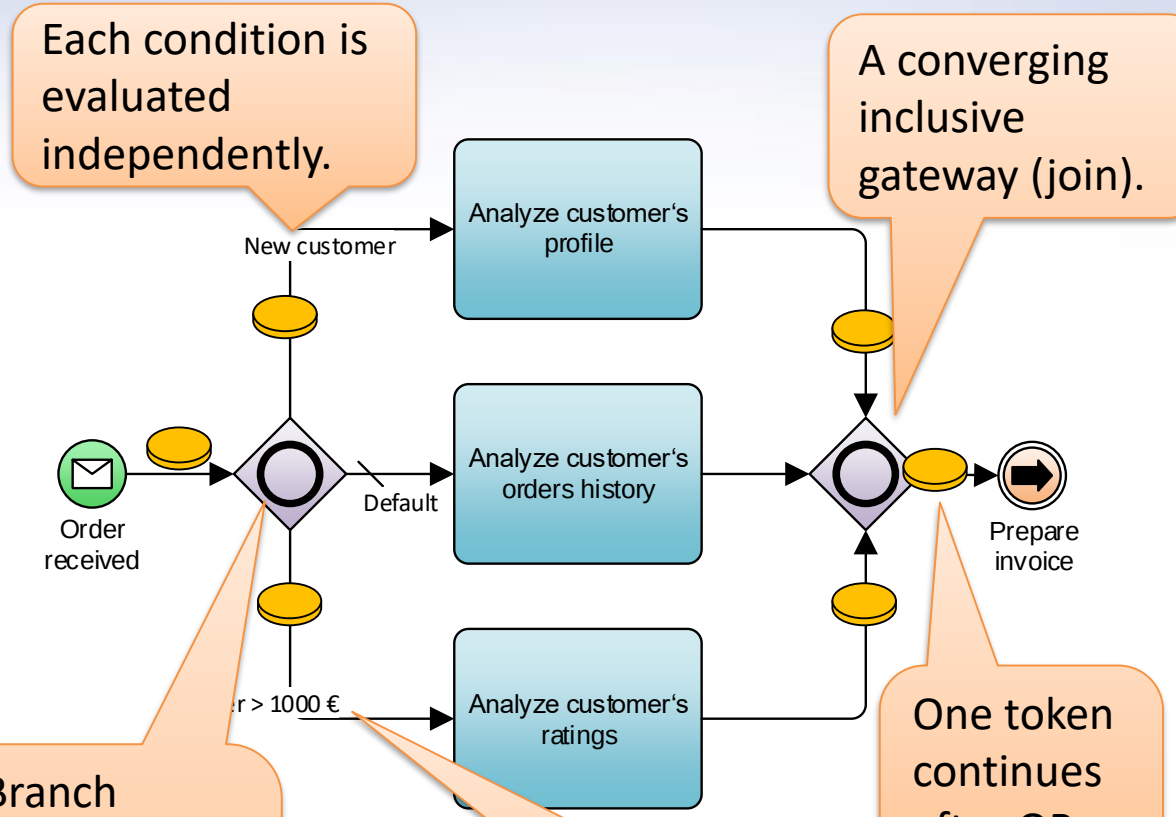
BPMN Gateways: Inclusive (OR)

BPMN 2.0
Page 292-293

- A diverging inclusive gateway (inclusive decision) can be used to create alternative but also parallel paths within a process flow.
- All combinations of the outgoing paths may be taken, from zero to all.
  - A default flow assures at least one active path.
- A converging inclusive gateway is used to merge a combination of alternative and parallel paths.
  - An or gateway join is like an and gateway join except that it ignores incoming sequence flows that are not enabled.



Inclusive gateway is semantically equal to conditional sequence flow.

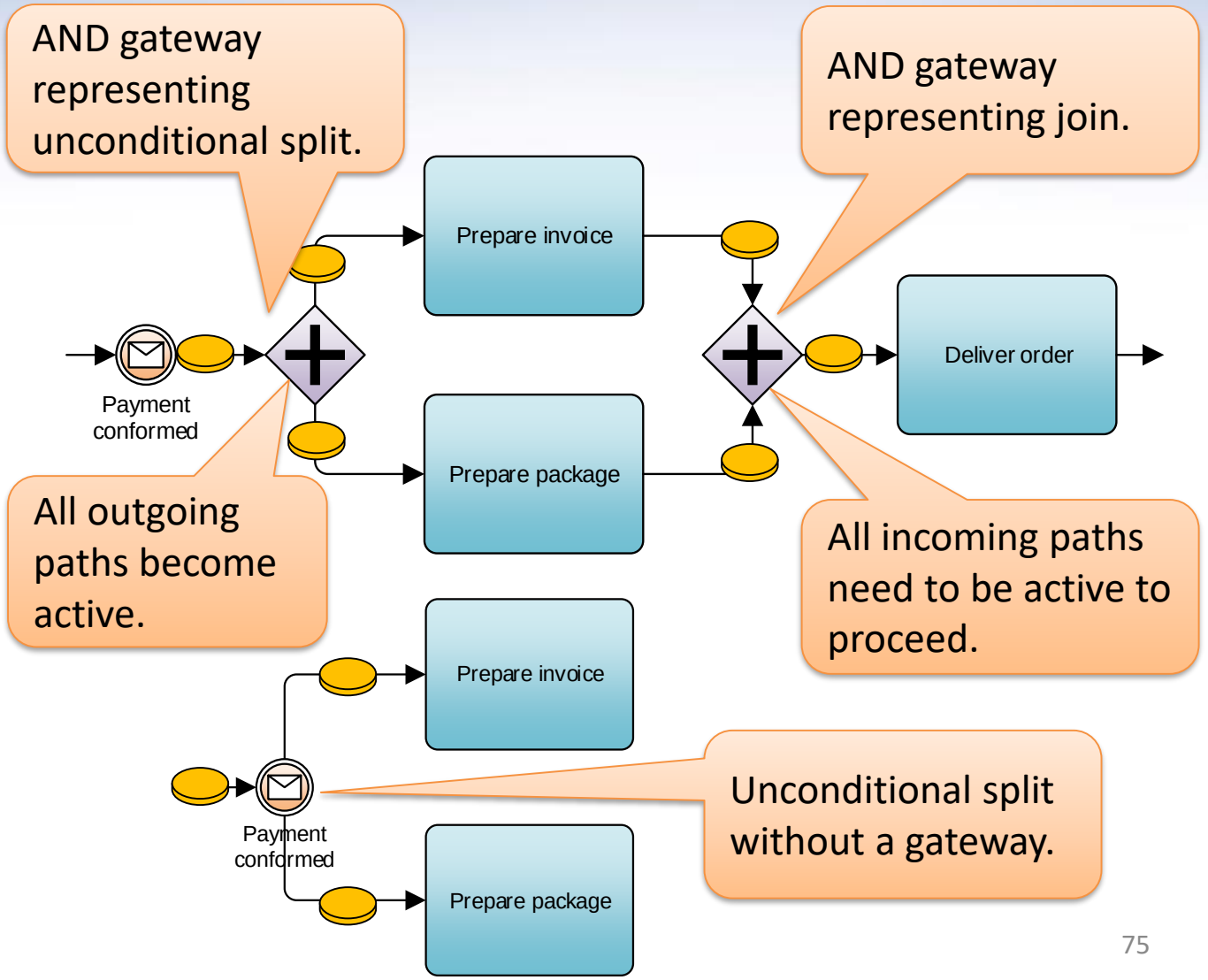


BPMN Gateways: Parallel (AND)

BPMN 2.0
Page 293-295

- A parallel gateway is used to create parallel flows and synchronize (combine) parallel flows.
  - A parallel gateway creates parallel paths without checking any conditions.
  - Each outgoing sequence flow receives a token upon execution of and gateway.

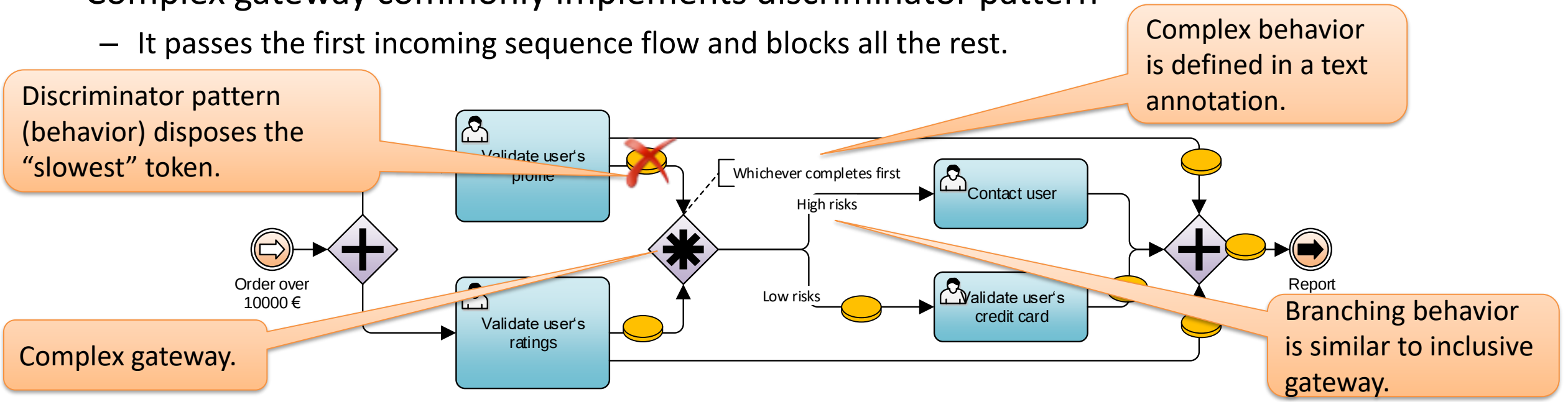
- For incoming flows, the parallel gateway will wait for all incoming flows before triggering the flow through its outgoing sequence flows.



BPMN Gateways: Complex

BPMN 2.0
Page 295-296

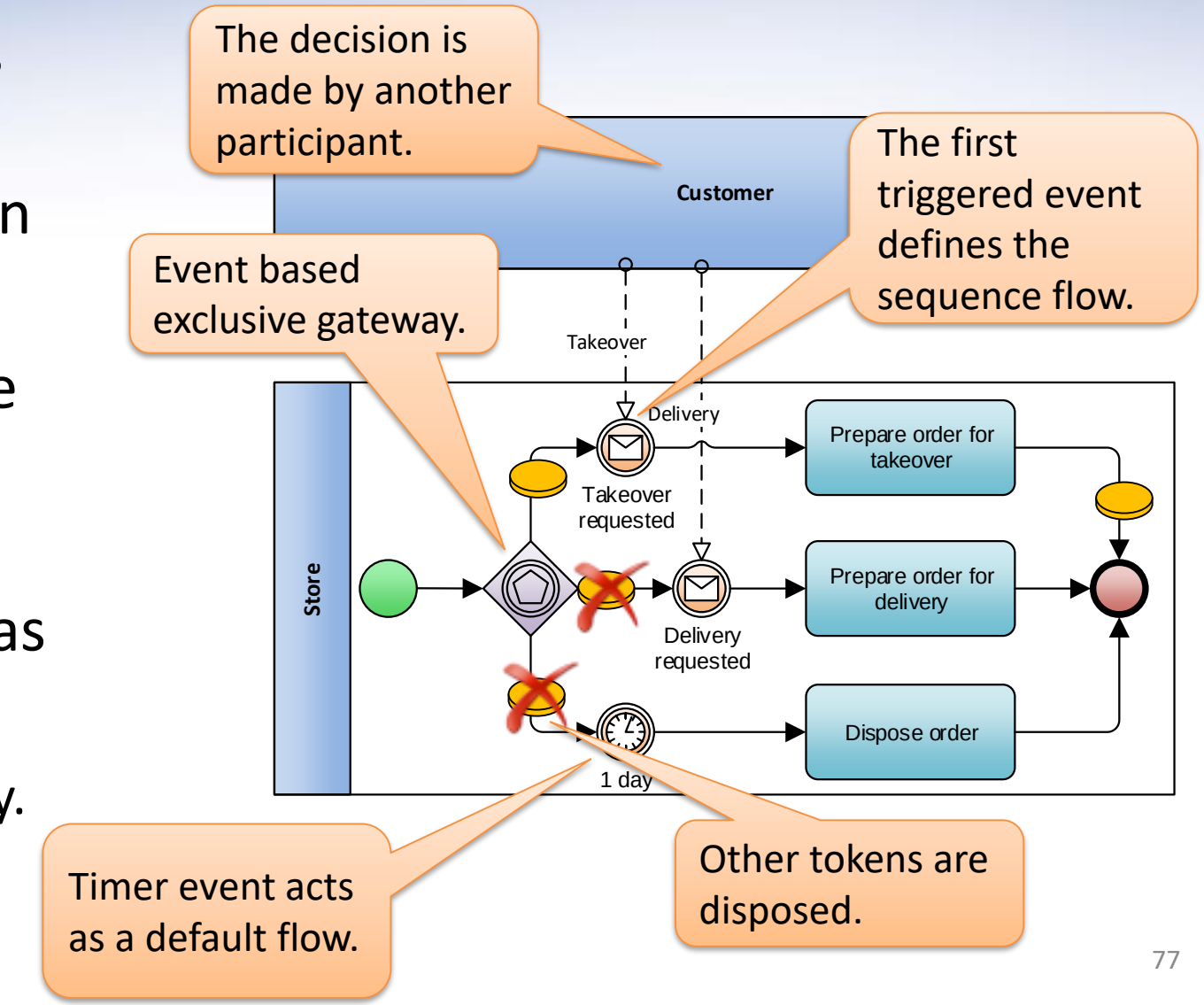
- The complex gateway is mainly used to model complex synchronization behavior.
  - Text annotation may be used to explain intended behavior.
  - Example behavior: three out of five incoming sequence flows are needed to activate the gateway.
- What tokens are produced by the gateway is determined by conditions on the outgoing sequence flows (similar to inclusive gateway).
- Complex gateway commonly implements discriminator pattern
  - It passes the first incoming sequence flow and blocks all the rest.



BPMN Gateways: Event Based XOR

BPMN 2.0
Page 297-298

- Event-based exclusive gateway is similar to XOR gateway but the trigger of the gateway is based on event instead of data condition.
- Each alterative path must include a ‘catching flow element’.
  - Receive activity may be also used.
- The timer event commonly acts as a ‘default event’.
  - E.g. wait for any messages for a day.

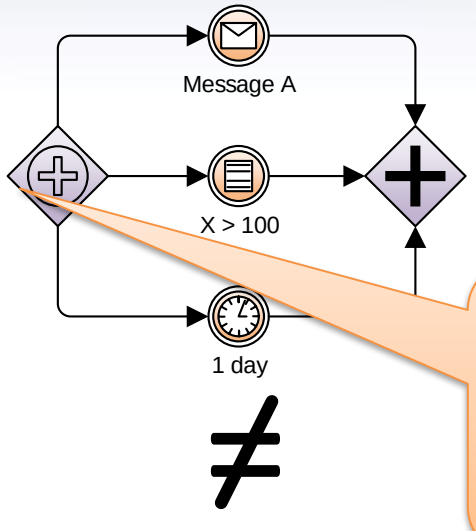


BPMN Gateways: Event Based Start

Parallel event-based gateway to start a process.

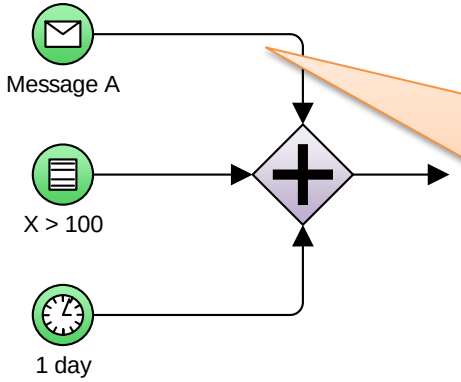
- Event-based start gateways are used to instantiate a process in case of several start events.

Exclusive event-based gateway to start a process

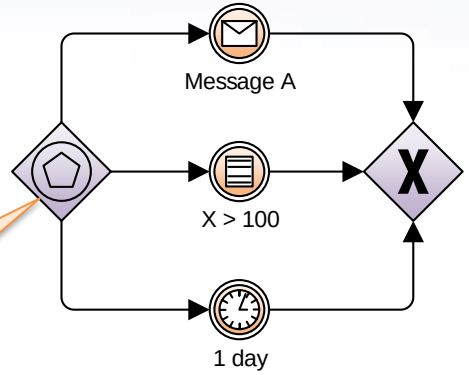


≠

The process is instantiated with the first triggered event. However other events are required for the same process instance.

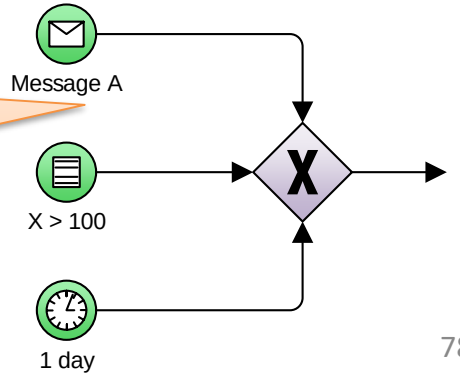


In this case triggering of each new start event creates a new process instance.



=

The process is instantiated if one of the corresponding events is activated.



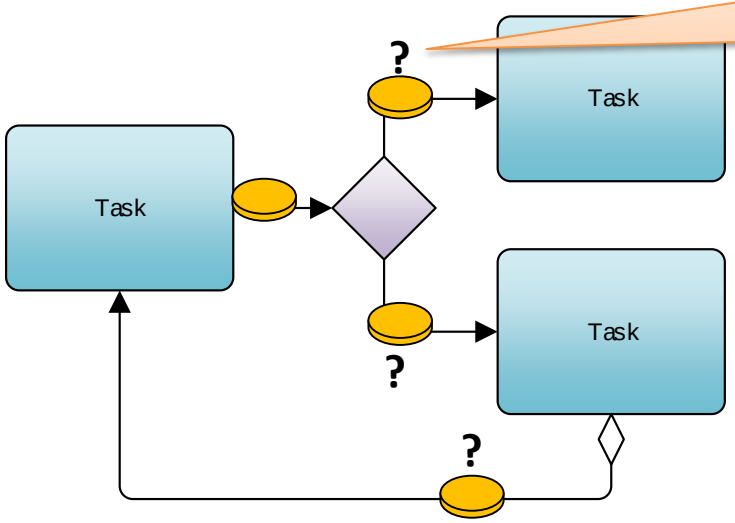
The behavior is similar to an XOR gateway, which merges multiple star events.

Controlled and Uncontrolled Flows

BPMN 2.0
Page 500-503

Controlled flows

- Control of sequence flows from one flow object (i.e. gateway, activity, event) to another with:
  - Gateways
  - Conditional sequence flows

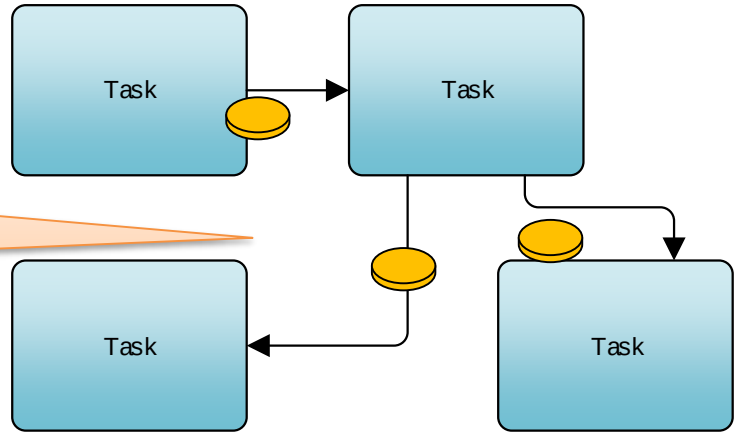


The execution of these flows (?) depends on a condition.

These flows perform always.

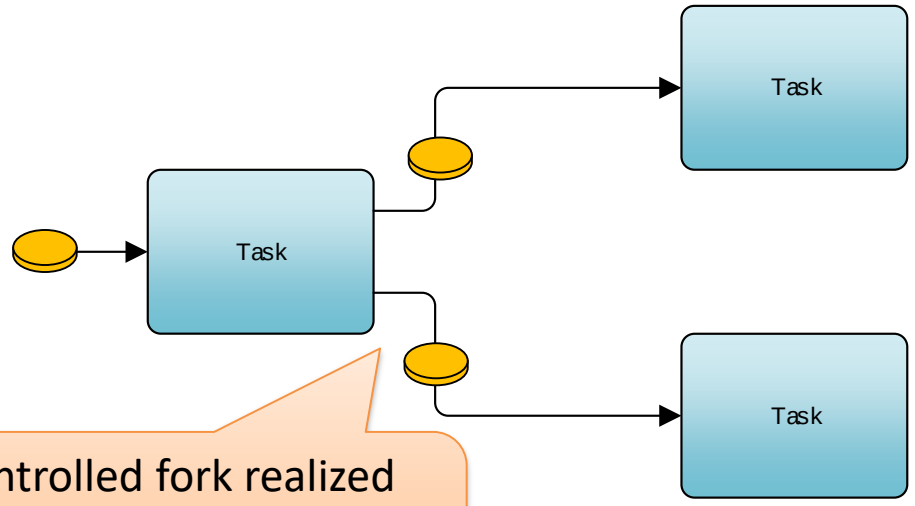
Uncontrolled flows

- No control of sequence flows from one flow object (i.e. gateway, activity, event) to another.
  - Use of (normal) sequence flows.



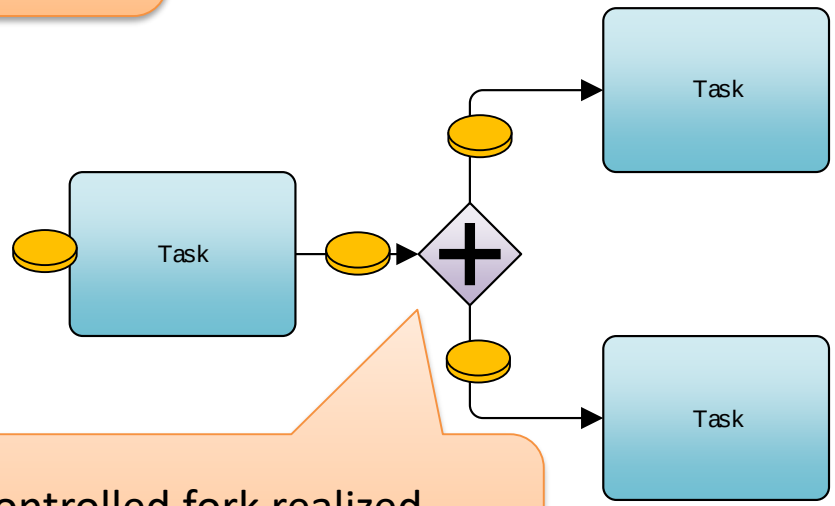
Forking Uncontrolled Flows

- BPMN uses the term “fork” to refer to the dividing of a path into two or more parallel paths (also known as an AND-Split). It is a place in the process where activities can be performed concurrently, rather than sequentially.



Uncontrolled fork realized with multiple outgoing flows.

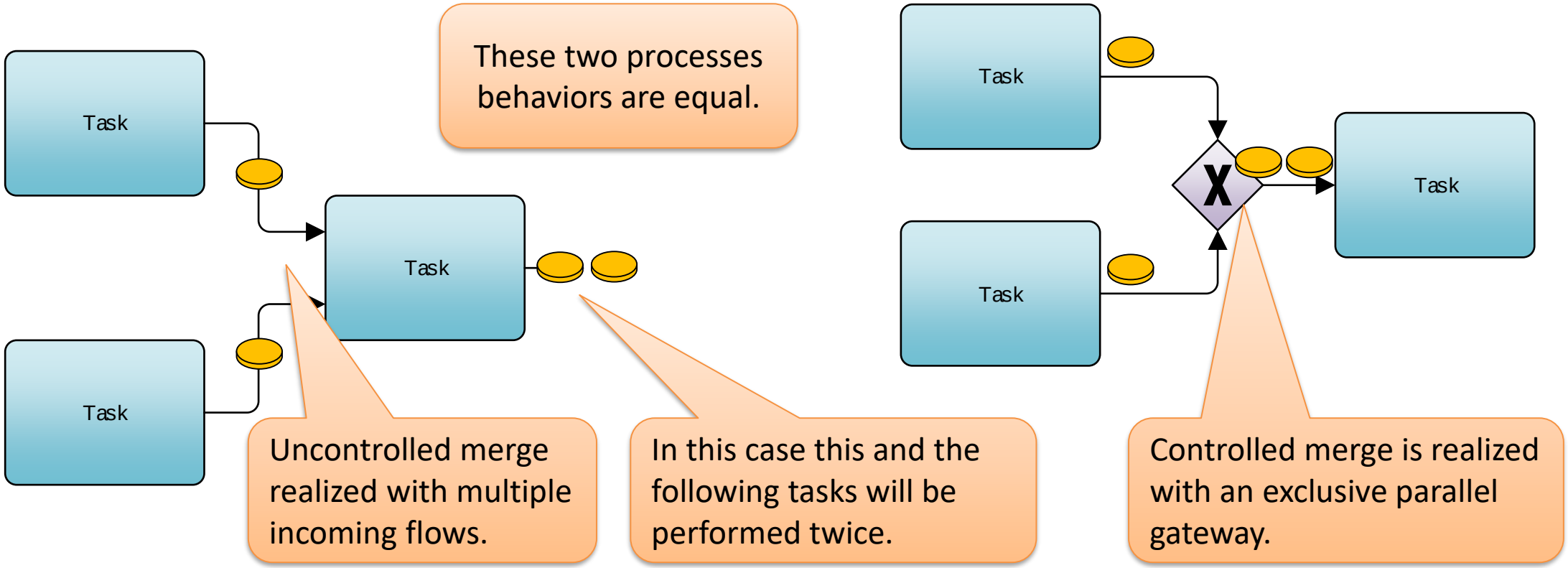
These two processes behaviors are equal.



Controlled fork realized with a parallel gateway.

Merging Uncontrolled Flows

- BPMN uses the term “merge” to refer to the exclusive combining of two or more paths into one path.
  - Uncontrolled merge should be used only if all incoming sequence flows are alternative (exclusive).



Items and Data

BPMN 2.0
Page 203

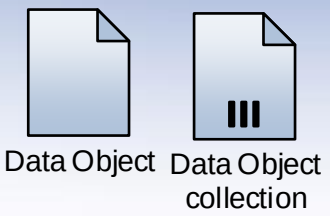
- Process modeling commonly requires modeling of items, that are managed (e.g. created, manipulated, stored, sent, received) during process execution.
- Items may represent:
  - Information (e.g. order and invoice) which moves via information flows.
  - Physical items (e.g. online store products) which move via material flows.
- BPMN elements, which are capable to manipulate with items, are called "item aware elements" (i.e. data objects, data object references, data stores, properties, data inputs and data outputs)
  - Properties are not visual BPMN elements, but are part of following flow elements: processes, activities and events.
  - Data object references are a way to reuse data objects in the same diagram. They can specify different states of the same data object at different points in a process.

An invoice (data) is an information item

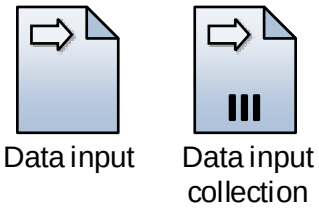
The products of an order might be physical items.

Data Objects

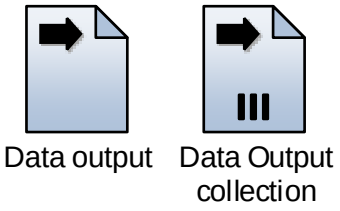
BPMN 2.0
Page 203-209



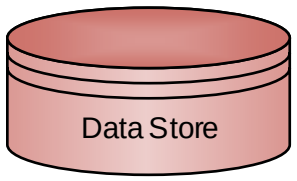
- Data object represents a local instance variable. It is visible only within the process level in which it is defined and its child levels and the variable is disposed when the process level instance is complete.
- Data object collection represents an array of data object elements.
  - Data object has an optional [state] attribute (e.g. draft, reviewed, released).



- Data inputs (collections) represent the inputs to a top-level process or a called process.



- Data output (collections) are visually displayed on a top-level process diagram to show the outputs of the process.



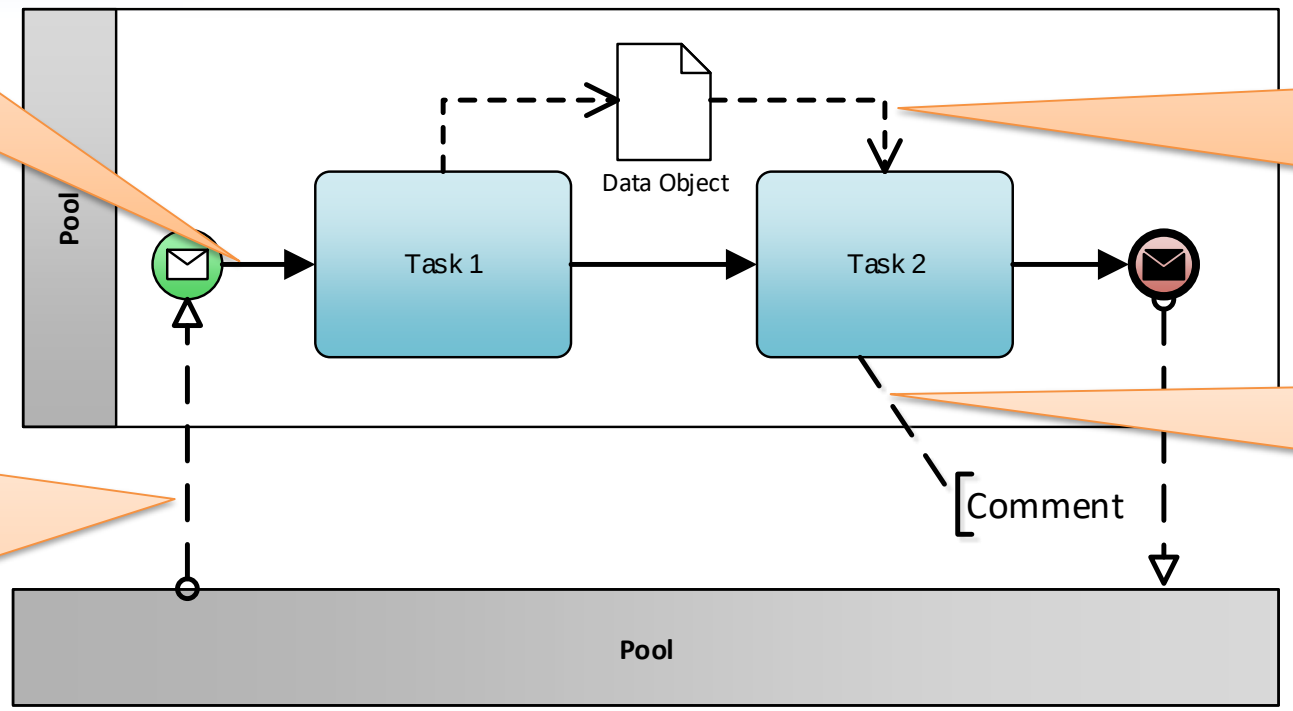
- A data store provides a mechanism for activities to retrieve or update stored information that will persist beyond the scope of the process.

Connecting Objects

- In BPMN, different types of connecting objects are necessary to represent different types of flows in a process diagram.

Sequence Flows are used to represent the order of flow elements in a process.

Data Associations are used to show the flow of information between activities in a business process.



Message Flows are used to show the flow of messages between two participants (i.e. pools) that are able to send and receive them.

Associations are used to link artefacts with other BPMN (graphical) elements.

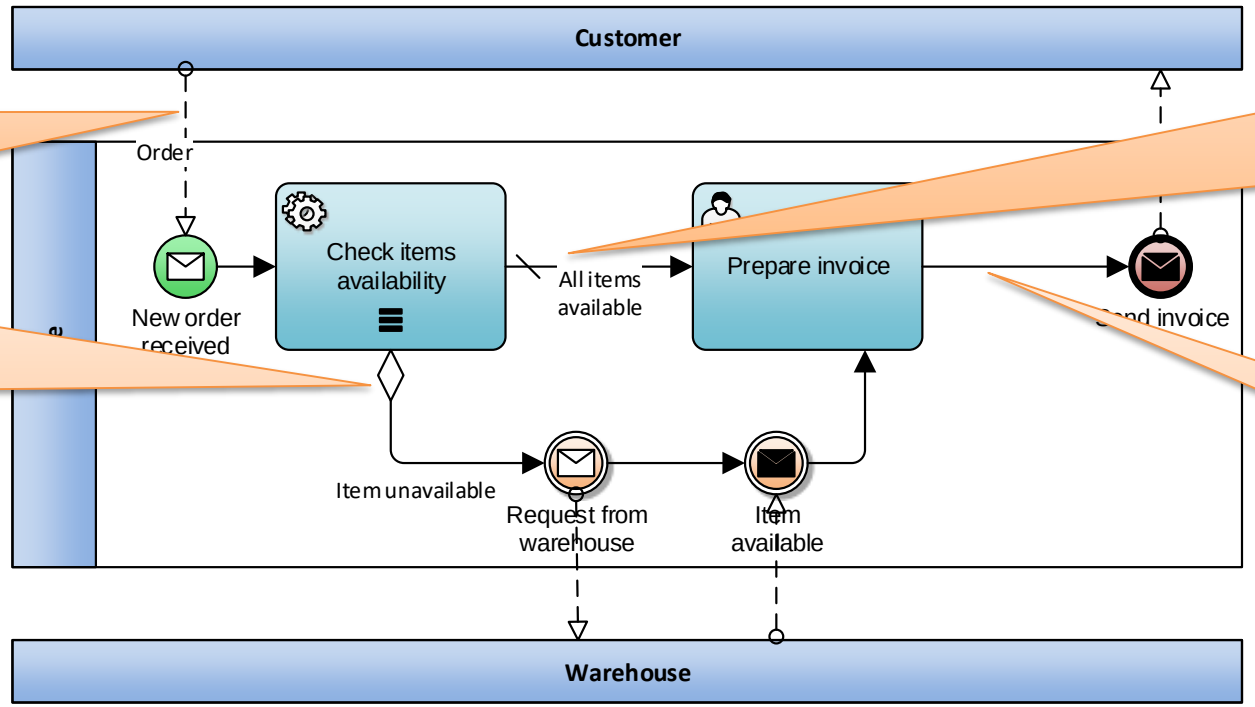
Connecting Objects: Sequence Flows

BPMN 2.0
Page 97-98

- A sequence flow is used to represent the order of flow elements in a process.
- Each sequence flow has only one source and only one target.
- Sub-types of sequence flows: default sequence flow, conditional sequence flow.

Sequence flows may NOT be used between Pools (processes).

Diamond represents conditional sequence flow (similar to inclusive gateway).



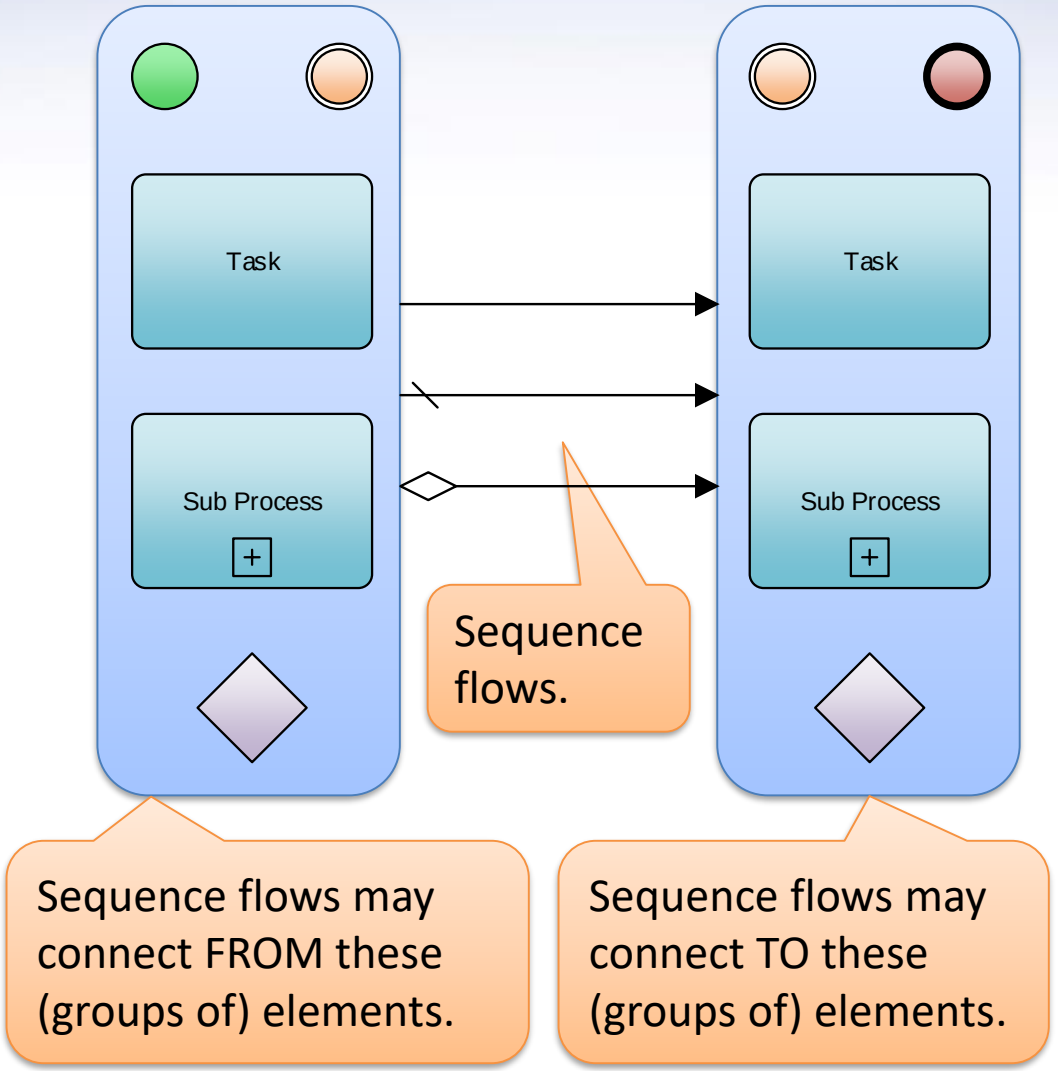
Default (sequence) flow is used to prevent deadlocks in case of conditional flows or gateways.

Solid arrows represent sequence flows.

Connecting Objects: Sequence Flow Rules

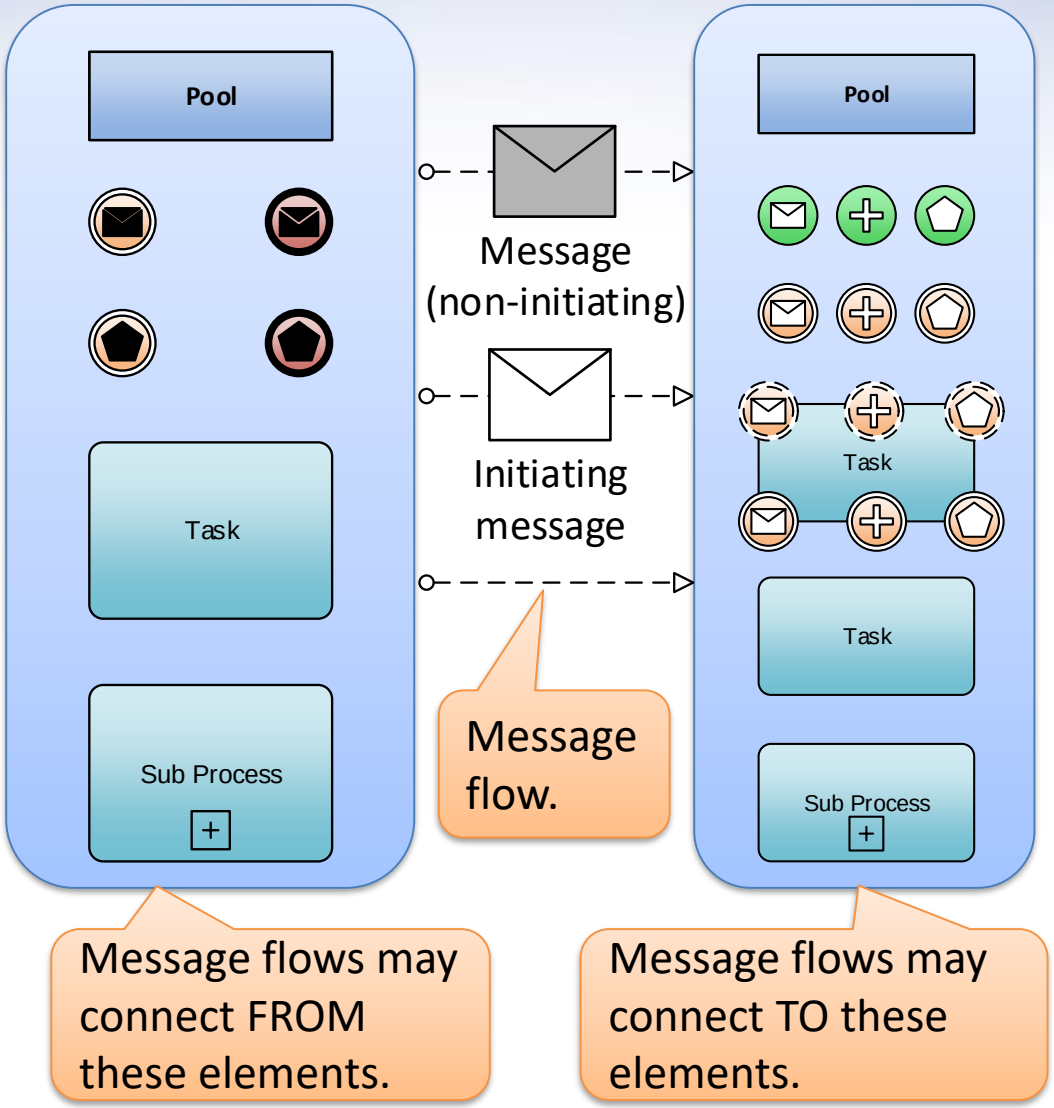
BPMN 2.0
Page 42-43

- Sequence flow may NOT connect to
  - A start event, catching link event.
  - Artefacts and data objects.
- Sequence flow may not result from
  - An end event, throwing link event.
  - Artefacts and data objects.
- Sequence flow may not cross:
  - Pool's boundary.
  - Process – sub-process boundary.



Connecting Objects: Message Flows

- A message flow is used to show the flow of messages between two participants that are 'capable' to send and receive them.
  - The source of a message flow must be either a message or multiple end event, or throwing intermediate event, an activity or black-box pool.
  - The target of a message flow must be either a message or multiple start event, catching (message, multiple) intermediate event, boundary event, an activity or a black-box pool.
- A message can be optionally depicted as a graphical decorator on a message flow.
- To associate a message to a particular process instance, bpmn uses correlations (non-visible concept).

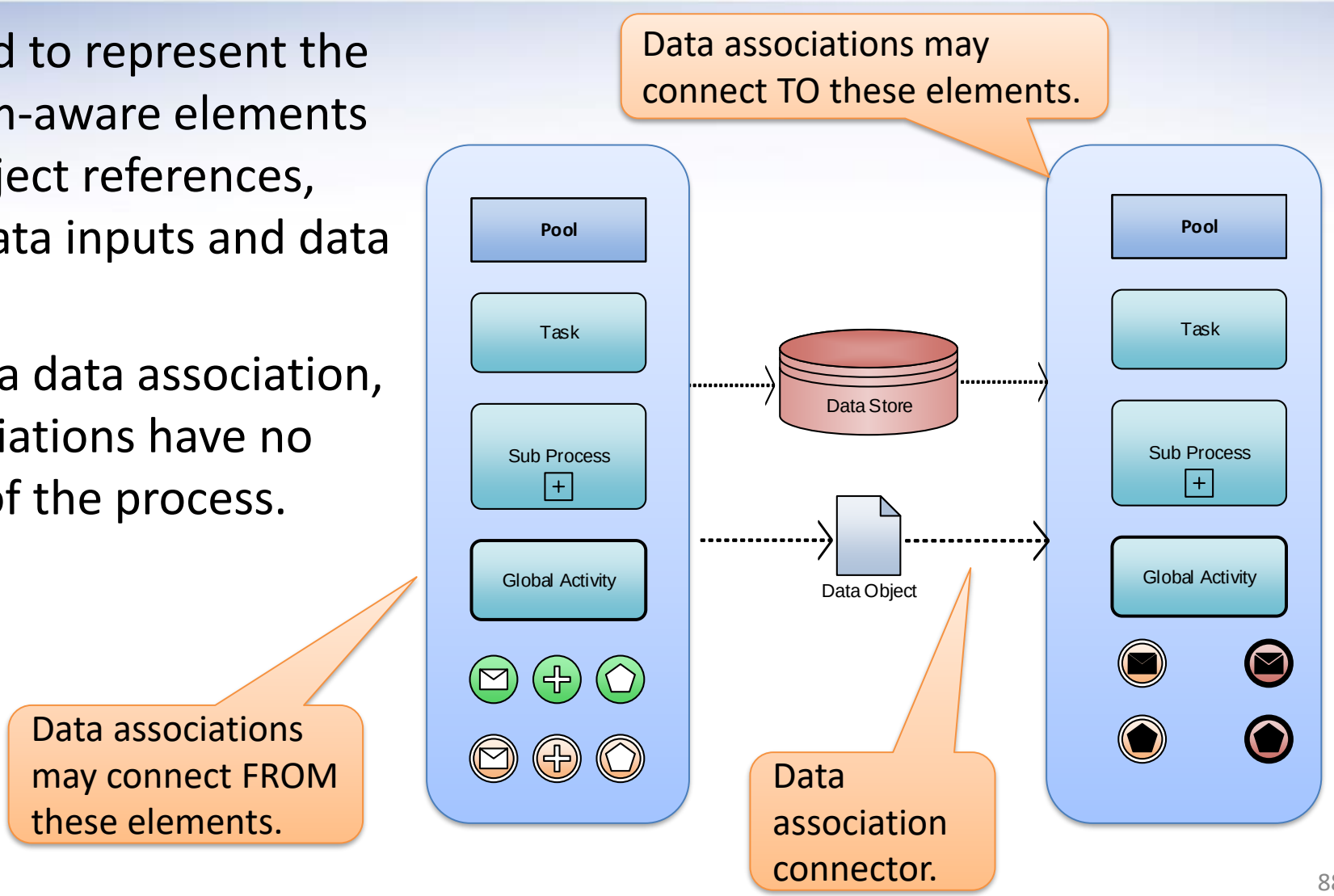


(Silver 2011)

Data Associations

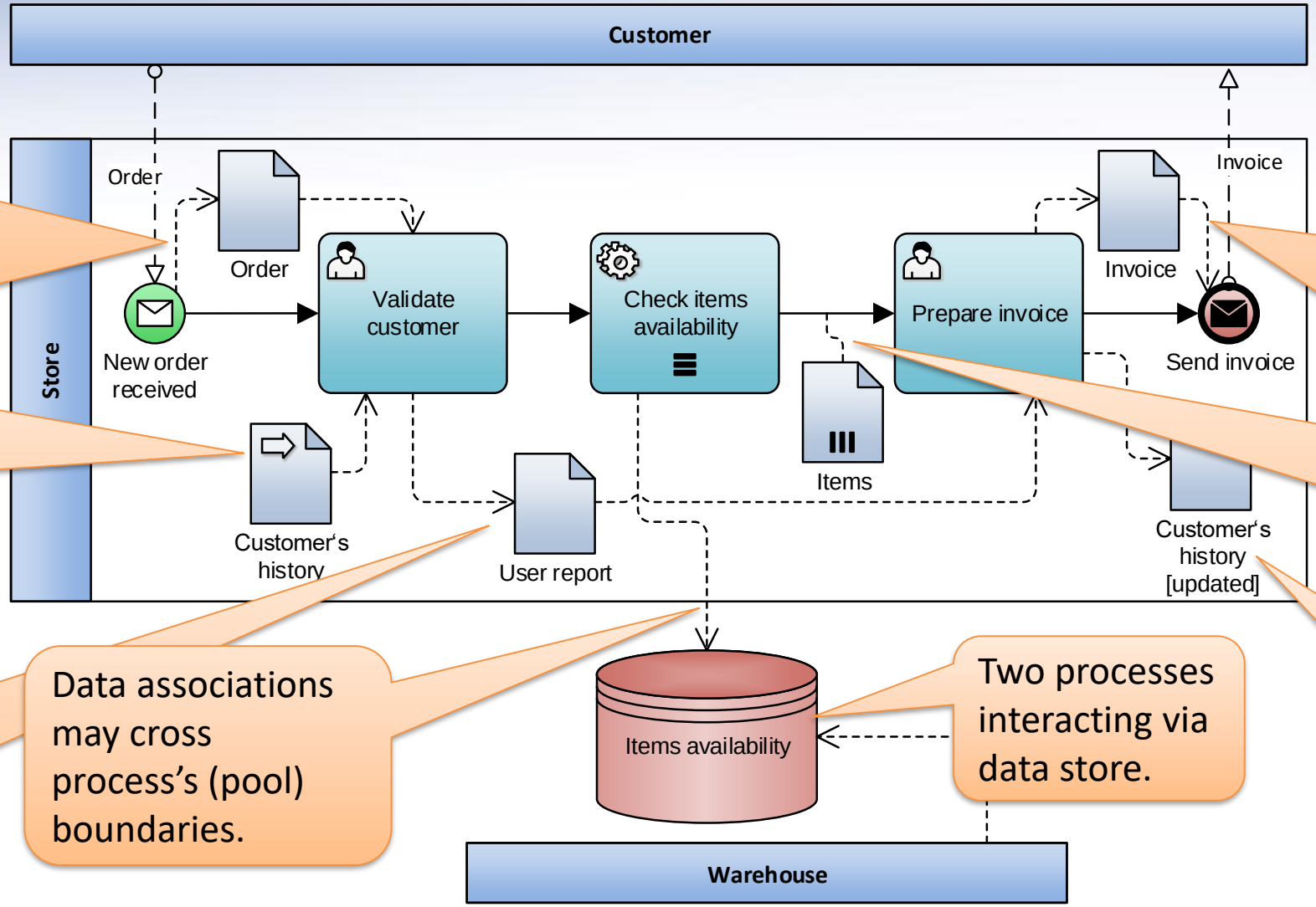
BPMN 2.0
Page 222-225

- Data associations are used to represent the flow of data between item-aware elements (i.e. data objects, data object references, data stores, properties, data inputs and data outputs).
- Tokens do not flow along a data association, and as a result data associations have no direct effect on the flow of the process.



Data and Data Associations Example

For a catch event, data associations are used to push data from the message received into data objects and properties.



For a throw event, data associations are used to fill the message that is being thrown.

Data object inputs and outputs provide information for processes.

Data objects MAY be directly associated with a sequence flow connector.

Data associations are used to represent inputs and outputs of activities.

Data associations may cross process's (pool) boundaries.

Two processes interacting via data store.

This data output object has a state defined [updated].

Full Set of Process Modeling Elements

BPMN 2.0
Page 3-4

	Start			Intermediate				End
		Event sub-pr.			Boundary			
		Inter.	Non-inter.	Catching	Inter.	Non. Inter.	Throwing	
None								
Message								
Timer								
Error								
Escalation								
Cancel								
Compen-sation								
Conditional								
Link								
Signal								
Terminate								
Multiple								
Multiple paralel								

- Manual Task
- User Task
- Service Task
- Send Task
- Receive Task
- Rule
- Script
- Task
- Ad-hoc
- Standard Loop
- Parallel multiple instance
- Sequential multiple instance
- Compensation
- Global (call) Activity
- Sub Process (collapsed)
- Sub Process (expanded) — Task → Task
- Event-based Sub Process
- Transaction

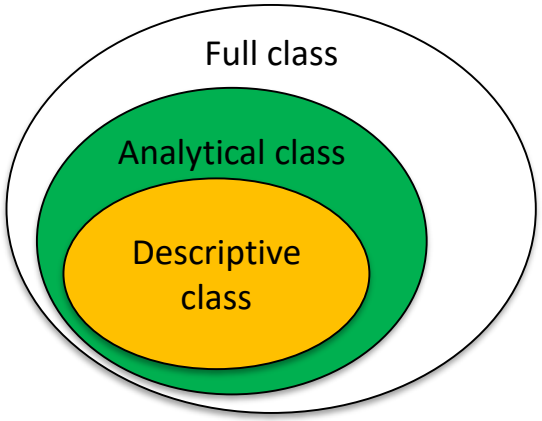
- Exclusive XOR
- Inclusive OR
- Parallel
- Event based
- Exclusive Event-based start
- Parallel Event-based start
- Complex

- Annotation
- Group
- Documentation

- Sequence flow
- Default flow
- Conditional flow
- Message flow
- Association
- Directed/data association

- Data object
- Collection
- Data input
- Data output
- Data store
- Message
- Initiating message

Pool
Lane | Lane | Lane



BPMN Modeling Rules (i.e. BPMN Syntax)

- Address the syntactic quality of BPMN diagrams.
- Formally defined in BPMN meta-model
  - UML class diagrams.
  - Include precise definition of BPMN elements, their attributes and the relationships between elements.
- BPMN modeling rules are scattered over the BPMN specification.

Excerpts of BPMN 2.0 rules, collected by Bruce Silver:

- All flow objects other than start events, boundary events, and compensating activities must have an incoming sequence flow, if the process level includes any start or end events.
- All flow objects other than end events and compensating activities must have an outgoing sequence flow, if the process level includes any start or end events.
- A start event cannot have an incoming sequence flow.
- A start event cannot have an outgoing message flow.
- A start event with incoming message flow must have a message trigger.
- A start event cannot have an error trigger.
- A start event in a sub-process must have a none trigger.
- A boundary event must have an outgoing sequence flow.
- A boundary event trigger must be either message, timer, signal, error, escalation, conditional, cancel, or compensation.
- A boundary event cannot have incoming sequence flow.
- An error boundary event on a sub-process requires a matching error throw event.

(Silver 2011)

BPMN Modeling Style

- Addresses the pragmatic quality of BPMN diagrams.
- Non-normative conventions used in BPMN diagrams.
- Out of scope of BPMN 2.0 Specification.
- Useful to create consistent and unambiguous BPMN diagrams.
- Can be acquired or evolved within an organization.

Examples of Bruce Silver's Style Rules:

1. Use icons and labels to make process logic clear from the printed diagram.
2. Make models hierarchical, fitting each process level on one page.
3. Use a black-box pool to represent the Customer of other external requester or service provider.
4. A child-level diagram should not be enclosed in an expanded sub-process shape.
5. The label of a child-level page should match the name of the sub-process.
6. Activities should be labeled.
7. Two activities in the same process should not have the same name. (Use global activity to reuse a single activity in a process.)
8. A Send task should have an outgoing message flow.
9. A Receive task should have an incoming message flow.
10. A Timer start event should be labeled to indicate the process schedule.
11. A start event in a top-level process should be labeled. If a top-level process contains more than one start event, all should be labeled to identify the alternative start conditions.

(Silver 2011)

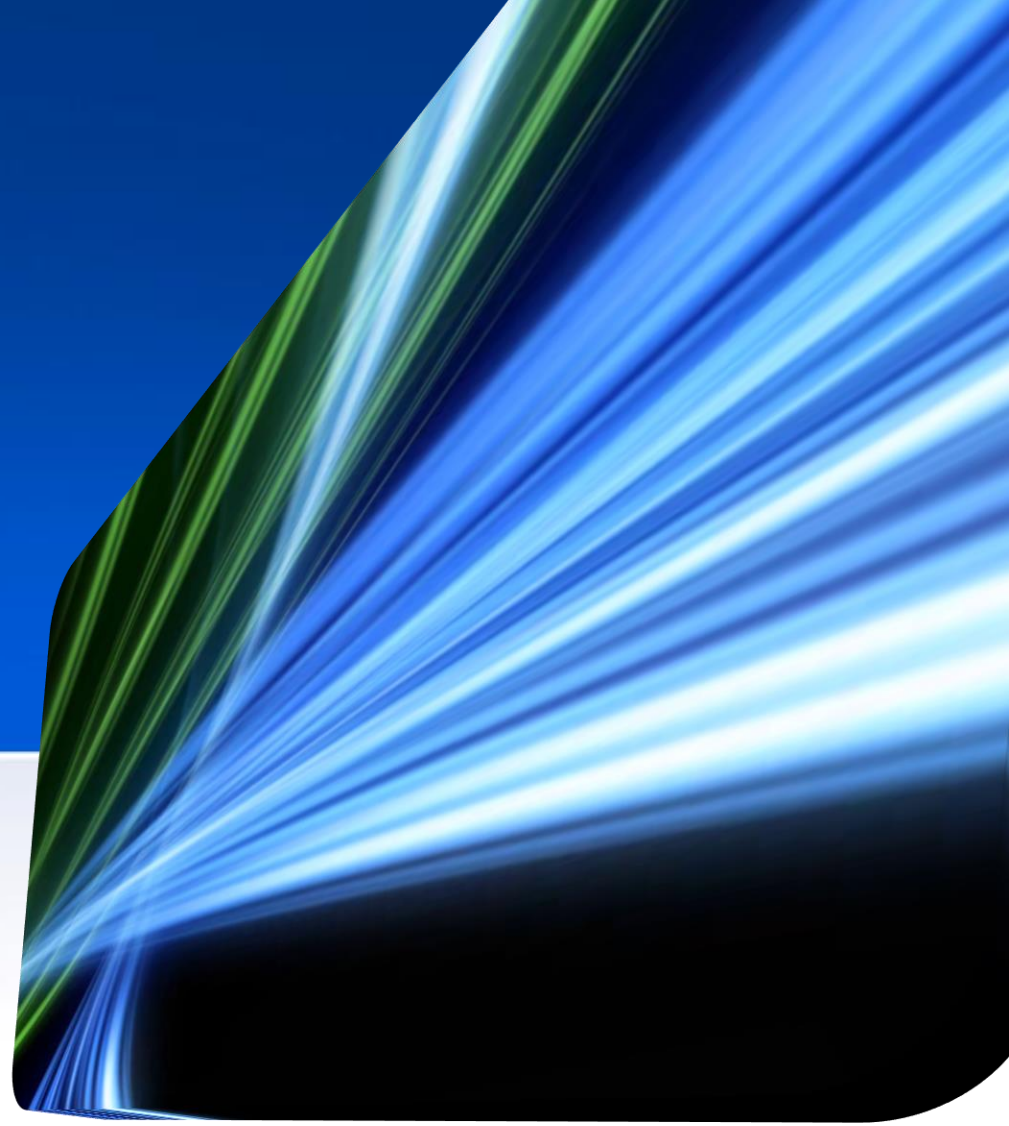
BPMN Modeling Method

- BPMN Method – A 'recipe' (also approach or process) about how to create business process diagrams.
- BPMN Method is out of scope of BPMN 2.0 Specification.
- Useful to create consistent business process diagrams across an organization.
- Can be acquired or evolved within an organization.

Bruce silver's method of hierarchical top-down modeling:

1. Agree on process scope, when it starts and ends, what the instance represents, and possible end states.
2. Enumerate major activities in a high-level map, ten or fewer, each aligned with the process instance. Think about possible end states of each activity.
3. Create top-level bpmn diagram. Arrange high-level map activities as sub processes in a BPMN process diagram, with one top-level end event per process end state. Use gateways to show conditional and concurrent paths.
4. Expand each top-level sub process in a child-level diagram. If sub process at parent level is followed by a gateway, match sub process end states to the gateway (or gate) labels.
5. Add business context by drawing message flows between the process and external requester, service providers, and other internal processes, drawn as black-box pools. Message flows connecting to collapsed sub process at parent level should be replicated with same name in the child-level diagram.
6. Repeat steps 4 and 5 with additional nested levels, if any.

(Silver 2011)

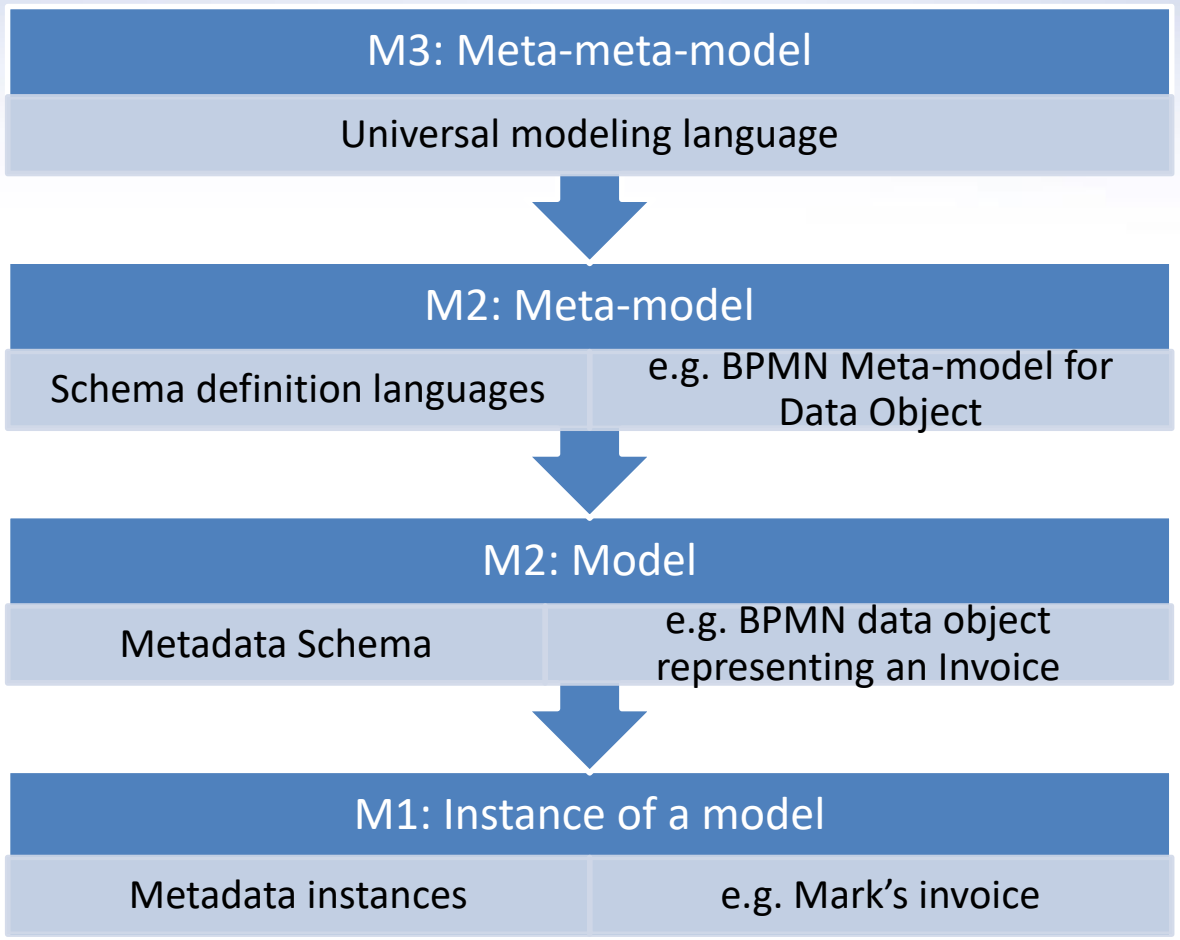


Business process modeling with BPMN 2.0

BPMN BEYOND PROCESS MODELING

BPMN Meta-model

- Meta-model is a formal specification of:
  - semantic BPMN elements (most of them have visual representations) and
  - relationships between semantic BPMN elements.
- Meta-model is represented in UML's class diagrams.
  - (semantic) BPMN elements are represented as object classes with defined required and optional attributes.
- All valid BPMN models must conform to the specification of the meta-model.



(Silver 2011)

MOF Meta-Levels Hierarchy

BPMN Meta-model Example: Data Object

BaseElement is the abstract super class for most BPMN elements. It provides the attributes id and documentation, which are inherited other elements (classes).

Classes are organized in packages (e.g. foundation).

Meta-model defined in UML's class diagram notation.

Each element might have documentation attached to it.

Inheritance relationship.

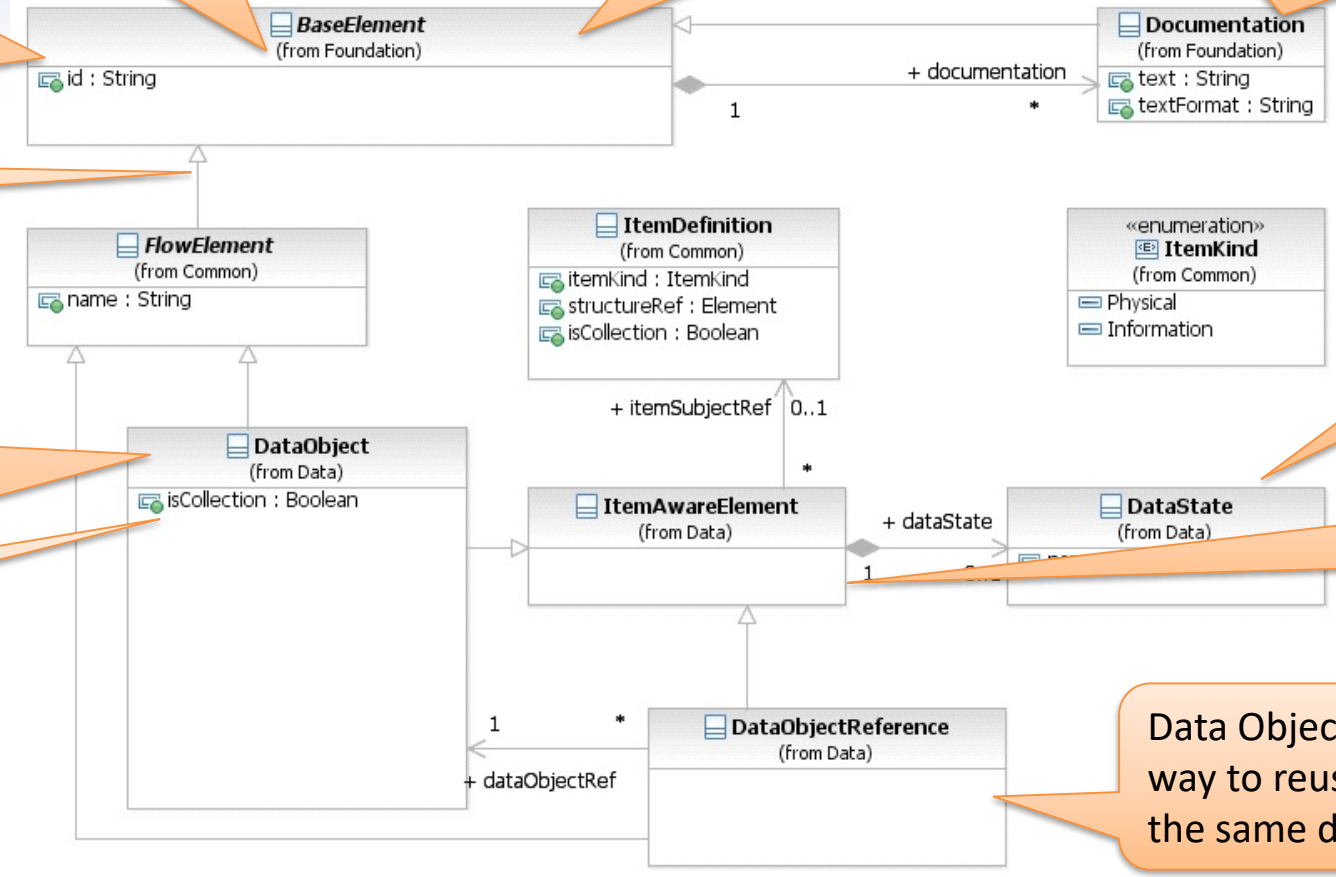
The DataObject element inherits the attributes and model associations of FlowElement and ItemAwareElement.

isCollection attribute defines if the Data Object represents a collection of elements.

DataObject might have Data states (e.g. draft, in review, final).

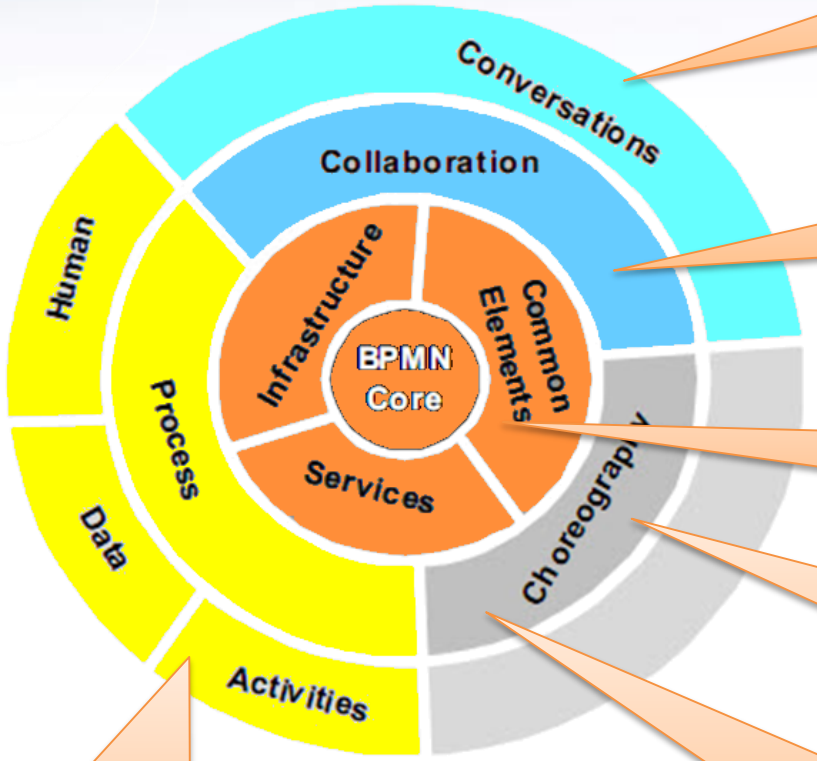
DataObject is an Item Aware Element.

Data Object References are a way to reuse Data Objects in the same diagram.



BPMN Layered Structure

- The **BPMN** specification is structured in layers (i.e. packages of classes, where a class represents a semantic BPMN element).
  - A semantic element might have a visual representation or not.
- Each layer defines a subset of BPMN semantic elements and models.
  - Each layer builds on top of and extends lower layers.
- Process modeling with BPMN includes elements and diagrams, which belong to all layers except choreography.



This layer defines elements, specific for Conversation diagrams.

This layer defines elements, specific for Collaboration diagrams.

Basic BPMN elements, which are common to all BPMN models.

Layer of elements, which are specific for choreography models.

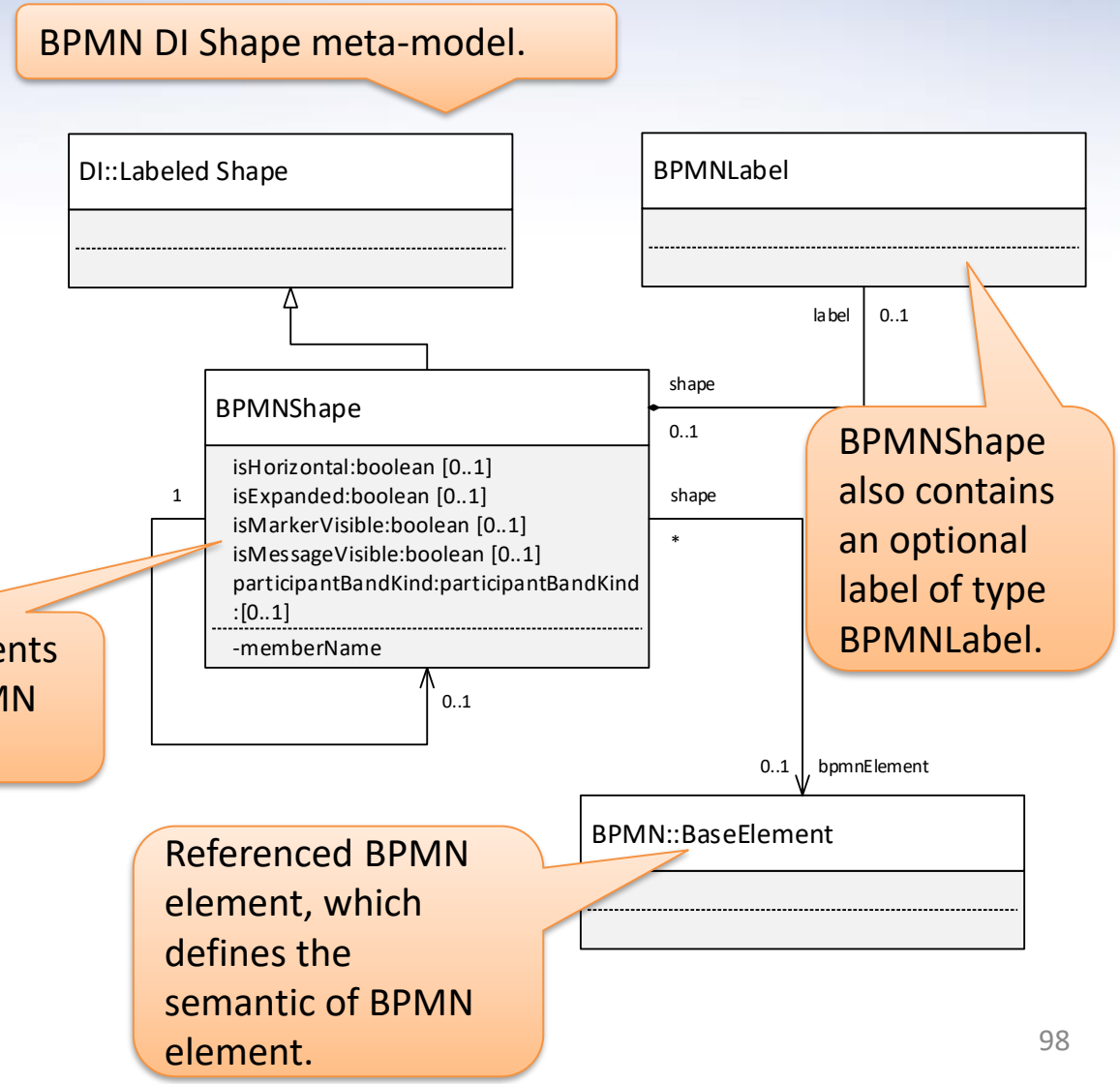
Choreography layer IS NOT part of process modeling conformance.

Elements, which belong to yellow layers are specific for process diagrams.

BPMN Diagram Interchange (DI)

BPMN 2.0
Page 372

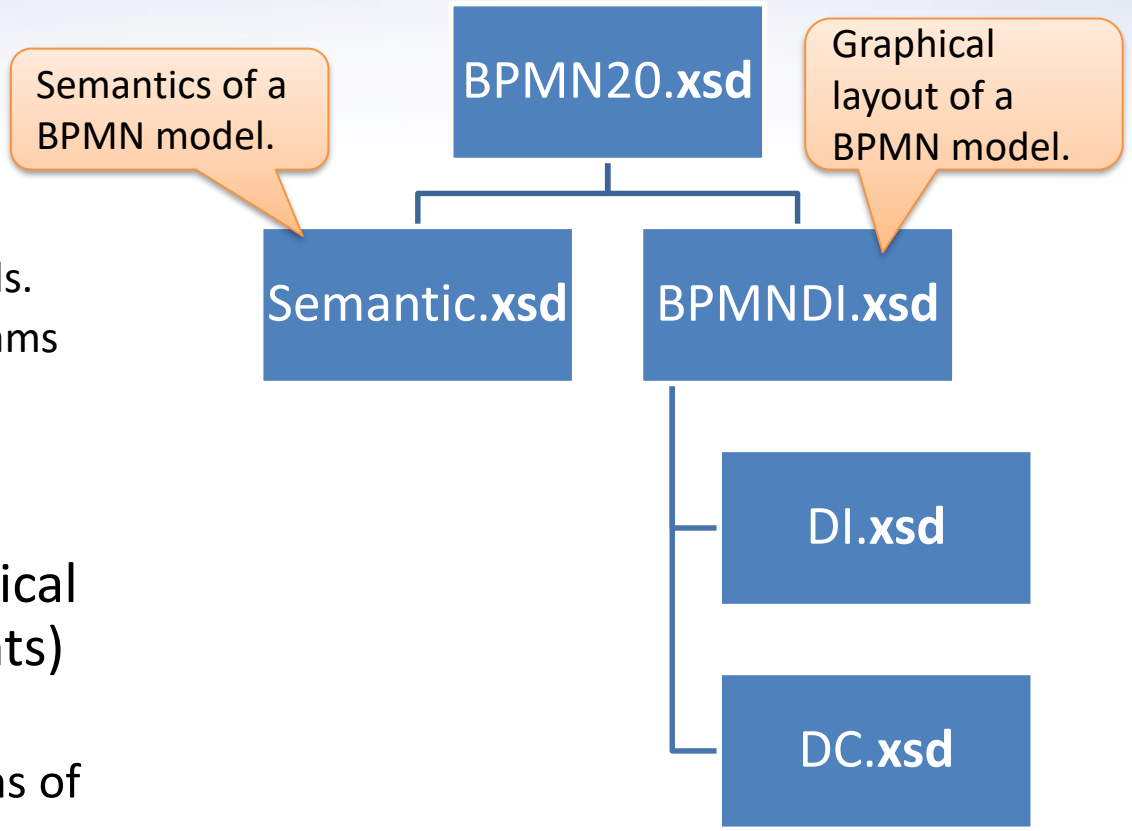
- BPMN DI describes the location and size of shapes and connectors as well as the linked page structure of BPMN diagrams.
  - BPMN DI Meta-model is similar to BPMN's semantic meta-model.
- The BPMN DI is meant to facilitate interchange of BPMN diagrams between tools rather than being used for internal diagram representation by the tools.
- BPMN diagram consists of
  - BPMN DI meta-model information,
  - BPMN semantic meta-model information.



(Silver 2011)

BPMN Serialization

- BPMN Meta-models (semantic and DI), which are represented in class diagrams, are published (i.e. serialized) in two alternative XML formats:
  - OMG's Metadata interchange (XMI) and
  - W3C's XML Schema Definition (XSD).
    - Most BPMN tool vendors use XSD for interchanging models.
    - Cannot represent certain relationships of UML class diagrams (e.g. multiple inheritance).
  - Transformations between XSD and XMI exist.
    - Defined in XSLT.
- In BPMN XSD, the information concerning the graphical layout of shapes (e.g. position, size, connection points) is separated from the semantic model.
  - A valid BPMN model may omit the graphical informations of a BPMN model completely.



BPMN 2.0 schema file structure

(Silver 2011)

BPMN Execution Semantics

BPMN 2.0
Page 425

- Part of BPMN Process Execution Conformance.
  - Not required for BPMN Process Modeling Conformance.
- Describes a clear and precise understanding of the operation of BPMN ‘executable’ elements.
  - Common executable subclass of BPMN elements defines basic ‘executable’ BPMN elements.
  - Those BPMN elements, capable of being executed on a process engine.
  - Non-operational elements examples: manual task, ad-hoc process and abstract task.
- BPMN execution semantics defines:
  - Process instantiation and termination,
  - Flow elements (activities, gateways, events) behavior,
- Execution semantics includes:
  - A description of the operational semantics of the element.
  - Exception issues for the element where relevant.
  - List of workflow patterns supported by the element where relevant.

BPMN Execution Semantics: Activity Lifecycle

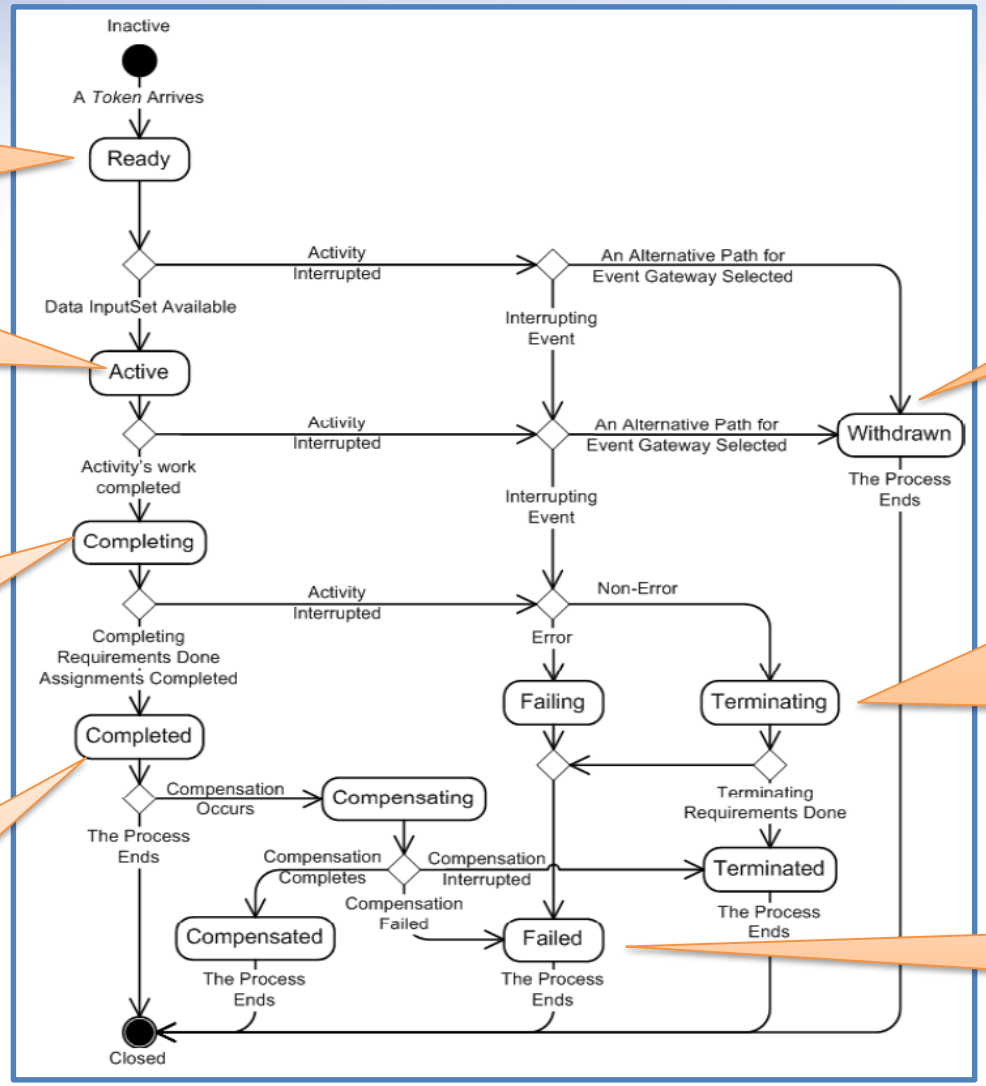
BPMN 2.0
Page 428-429

An activity is ready for execution if the required number of tokens is available to activate the activity /…/.

When some data InputSet becomes available, the Activity changes from Ready to the Active state /…/.

If an Activity's execution ends without anomalies, the Activity's state changes to Completing /…/.

After all completion dependencies have been fulfilled, the state of the Activity changes to Completed /…/.



An Activity, if Ready or Active, can be Withdrawn from being able to complete in the context of a race condition /…/.

An Activity's execution is interrupted if an interrupting Event is raised or if an interrupting Event Sub-Process is initiated. In this case, the Activity's state changes to Failing (in case of an error) or Terminating (in case any other interrupting Event).

If an Activity fails during execution, it changes from the state Active to Failed.

References

- OMG, 2011. Business Process Model and Notation version 2.0. Available at: http://www.omg.org/spec/BPMN/2.0/ [Accessed January 15, 2014].
- Weske, M., 2012. *Business process management concepts, languages, architectures*, Berlin; New York: Springer. Available at: http://site.ebrary.com/id/10650313 [Accessed January 14, 2014].
- Orand, B., 2011. *Foundations of IT Service Management: The ITIL Foundations Course in a Book* 3rd ed. J. Villarreal, ed., CreateSpace Independent Publishing Platform.
- Silver, B., 2011. *BPMN method and style: a structured approach for business process modeling and implementation using BPMN 2.0*, Aptos: Cody-Cassidy Press.