BiZdesign

Enterprise Studio

BPMN Getting Started Guide

2017-09-21

Applies to: Enterprise Studio 3.0.0, Team Server 3.0.0

Table of contents

1 About modeling with BPMN

1.1 What is BPMN?

The BPMN modeling method (Business Process Model and Notation) allows the modeler to integrate three domains: process, participants, and information/data. Processes are described using activities, events, gateways, and flows. The participants of the process are represented by using lanes and pools. Information/data can be input, output, or updated during process execution.

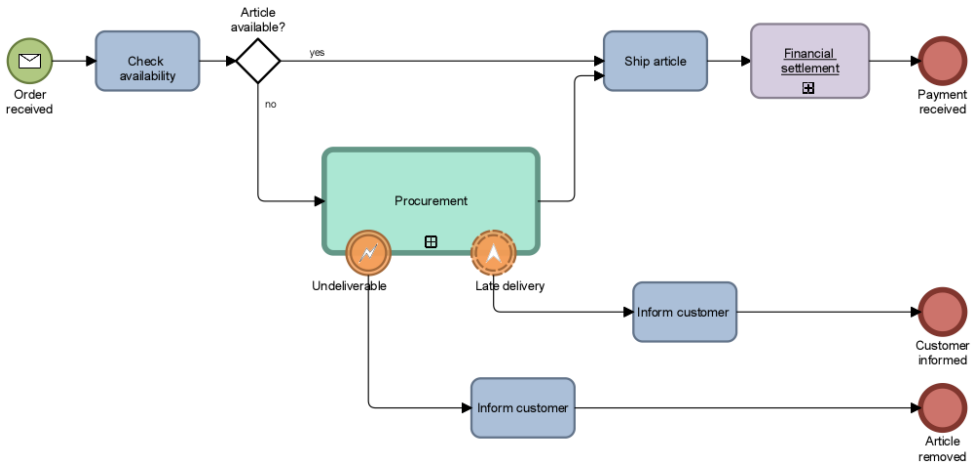


Figure 1.1 Example of a BPMN process

1.2 BPMN modeling

The different domains are modeled in an integrated way. Processes can be modeled from different perspectives, by using a number of different diagram styles: *processes*, *collaborations*, and *choreographies*.

If you have the required tool license, you can even relate your BPMN models to models from other domains, like entity-relationship (ER) models and UML models.

Modeling levels

BPMN models can be created at different levels of abstraction. Enterprise Studio distinguishes three levels for it: *Introductory*, *Descriptive*, and *Analytical*. The levels Descriptive and Analytical correspond to the levels defined within BPMN. The Introductory level is an additional level, defined by BiZZdesign.

1.3 BPMN in Enterprise Studio

A basic understanding of the BPMN modeling language is assumed for modeling with BPMN in the tool.

Required tool license

The BPMN functionality is available to you if your Enterprise Studio license includes one of the following tool packages: Pro BPM, Advanced, or Enterprise.[1]

Used colors

The different concepts in the BPMN functionality in the tool have their own color. These colors are not defined by the BPMN specification, it does not distinguish colors for its concepts.

Templates and example models

Enterprise Studio has example models containing process diagrams, collaboration diagrams and a choreography diagram. These example models can be viewed and used as inspiration for your own models, but you can also edit them yourself. The example models can be opened by clicking **File > New > BPMN**, and then clicking one of the available examples.

Model templates are available if you prefer to start with an empty model with only a few basics for BPMN.

[1]Enterprise Studio tool packages "Pro EA" and "Pro BPM" only apply to software purchases before release 3.0.0 (September 2017). Later purchases only have the "Advanced" or "Enterprise" package.

Contents of the documentation

The BPMN documentation focuses on the basics of modeling with BPMN in Enterprise Studio. It supports the BPMN 2.0 specification. For a more detailed explanation and application of the BPMN language, please refer to the official BPMN specification, third party BPMN documentation, or visit the website of the Object Management Group (OMG).

2 Components of a BPMN model

2.1 Diagram types and data for BPMN modeling

2.1.1 Process

A *process* diagram is used to describe the collection of activities on the path from a request for a product or service from a customer, and the delivery of that product or service to that customer. You model the behavior of a single process. The process diagram shows the order in which activities are organized, when and under what conditions activities should be executed, including exceptions. However, it does not show exactly how an activity is executed, and neither where nor why the activity should be executed.

A process runs from a starting point (*start event*) to and end point (*end event*), with possible occurrence of events between the start and end of the process (*intermediate events* and *boundary events*). Events are defined as something happening during the execution of a process. Events determine the course of the process flow and are usually the result of a cause (*trigger*), or exert some influence (*result*).

In between events, activities (*tasks*) are positioned. An activity can be broken down into underlying, lower level (more detailed) activities and in this way describe a sub-process (*sub-process*, *transaction*, *call activity*).

In the process model, relationships (*sequence flows*) are used to determine the order between the starting point, the activities, the intermediate events, and the end point. Moreover you can add splits, joins and repetitions of (parts of) the process with the use of *gateways*.

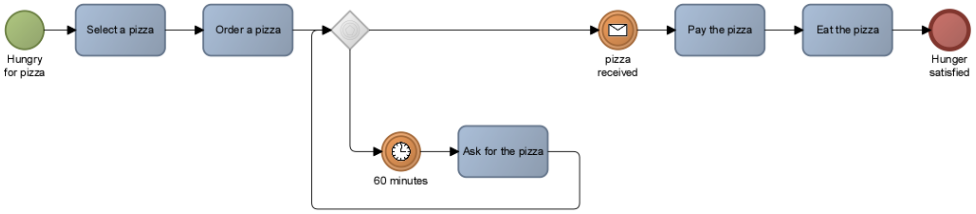


Figure 2.1 Example of a process

In order to describe who is involved in the execution of a process, *lanes* can be used. In this way, a lane can represent roles or organizational units that perform the activities in the process, or in other words, the *participants* of a process. A participant can represent the role of a person, but also systems or departments. The name of a lane is often the name of the entity that executes the process described in that lane, although in some situations it is more useful to use the name of the process.

The use of lanes, as well as the number of lanes to be used is up to you. These decisions depend on whether or not you want to add a responsibility aspect to your process.

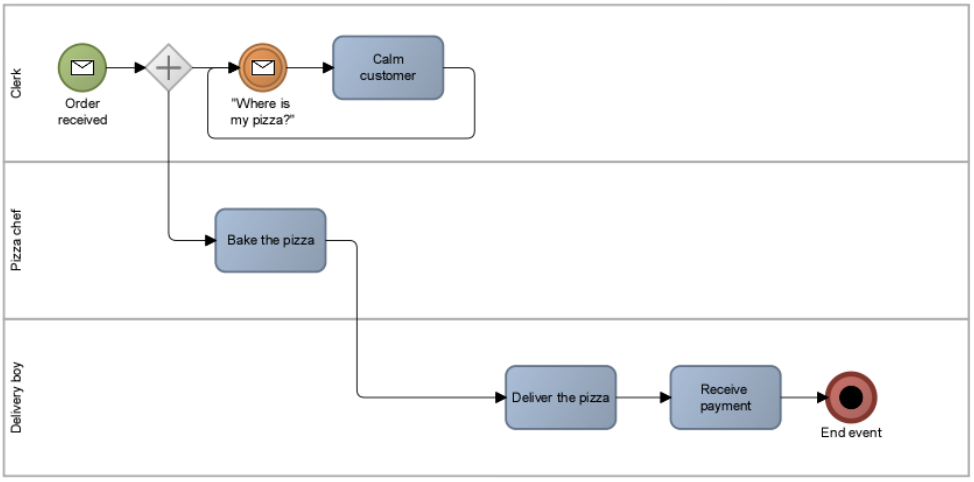


Figure 2.2 Example of a process, with the use of lanes

A process diagram using lanes describes participants of the process from an internal perspective only. In order to also describe external participants, you should use the collaboration.

2.1.2 Collaboration

A *collaboration* diagram can be used to visualize the various participants of the process and their interactions. It describes the cooperation between the participants. A collaboration combines a number of individual processes.

Similar to process diagrams, a collaboration diagram consists of activities, sequence flows, and gateways. Also, participants of a collaboration are represented using lanes. In a collaboration however, they are classified as subdivisions of the rectangle in which the process is (*pool*). Similar to a lane, a pool can represent an organizational unit, a role, or a department.

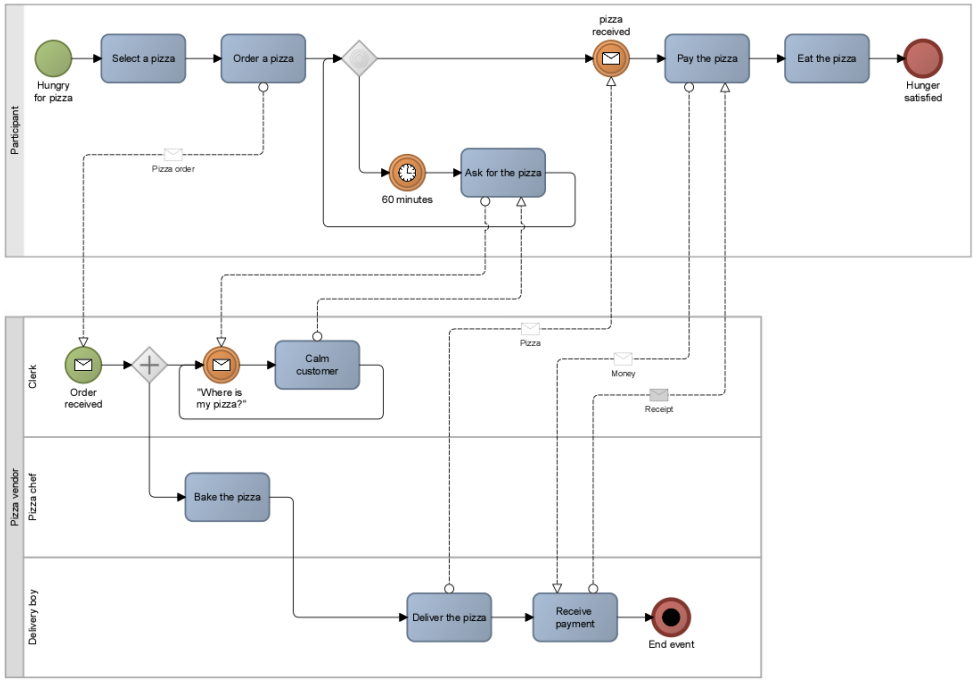


Figure 2.3 Example of a collaboration consisting of two pools (processes)

Any external participants (e.g. the customer for which the process is carried out) can also be included, using a separate, empty pool. Within BPMN these pools are referred to as "black boxes", as the details of the process of external participants are not further specified.

The interaction between participants in a collaboration is modeled using *messages*, the actual communication is visualized using *message flows*.

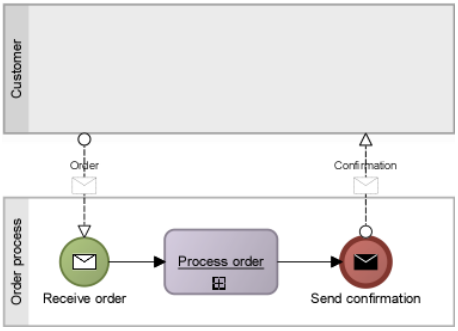


Figure 2.4 Example of a collaboration that includes an external participant as well as message flows

2.1.3 Choreography

A *choreography* is a type of process, but differs from an actual process in terms of goal and behavior. A process defines the ordered structure of activities within an organization or department. A choreography diagram visualizes the way in which participants coordinate their interactions. The focus is not on the orchestration of tasks performed by participants, but rather on the exchange of information among participants. An orchestration can also be considered as a business agreement between two or more organizations. This implies that the exchange of messages is executed in an orderly fashion.

The figure below shows an example of a choreography based on the earlier shown collaboration figure with the two pools "Pizza vendor" and "Participant". The following choreography maps out the exchange of information between the two pools.

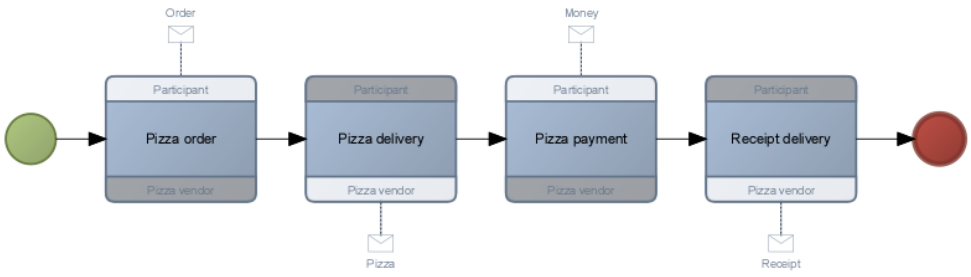


Figure 2.5 Example of a choreography

Similar to a process, a choreography runs from a starting point (start event) to an end point (end event), with in between activities which are here called *choreography tasks* and *call choreographies*, and possible intermediate events. Also, activities can be broken down into underlying, lower level (more detailed) components and in this way describe a sub-process (*sub-choreography*). The starting point, the activities and the end point are connected by relationships (sequence flows) which describe the order of the process flow. Gateways can be used to indicate splits, joins and repetitions in the process flow.

2.1.4 Information

Data objects

The information used during execution of a process, like documents, files and other types of objects, can be represented using *data objects* in a process or collaboration. The data objects describe information that is updated and changed within the process, but does not influence the behavior of that process. Every data object sits within a specific process or sub-process, and its life cycle and visibility are limited to that (sub-)process. A *data input* can be used to specifically call out information objects necessary to start an activity, while a *data output* can be used to describe the (possible) outcomes/results of an activity.

Data stores

A *data store* represents a data structure from/to which a process can read/write, but will remain to exist after termination of the process. So it is a permanent location for data storage. Examples of a data store include a database, file cabinet or document management environments like SharePoint.

Associations

In Enterprise Studio the BPMN *association* and *data association* are considered the same and are called association. There is also only one symbol for it. An association is used to connect a data object with a flow object, or to connect two data objects. An association that is connected to an

activity or event provides a connection to the data input or output of this activity or event. With this you can also indicate the direction.

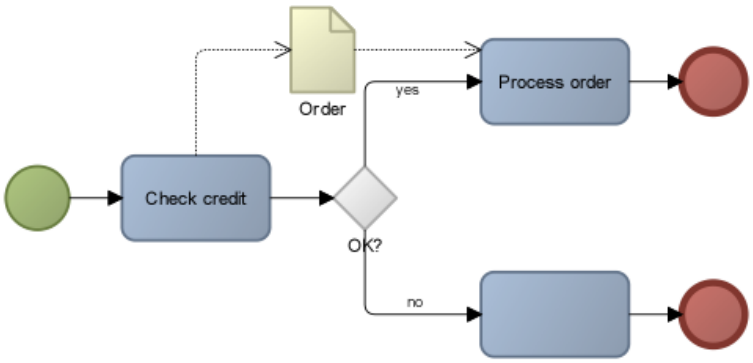


Figure 2.6 Example of a process with a data object

2.1.5 Resources and resource roles

In the BPMN language, *resources* and *resource roles* describe the resources to which activities in a process or collaboration can refer. A resource executes an activity, or carries the responsibility for it. A resource, for example a performer, can be specified in the form of for example a person, or a group of persons, but also an organizational role, a software application. An activity is, for example, performed by means of a work instruction. An example of this may be a work instruction that an installer uses to install a television for a customer.

A resource role is a more detailed specification of a resource. From a process of collaboration references can be made to these resource roles.

BPMN does not visualize resources and resource roles. Enterprise Studio does offer the possibility to visualize them by using a resources diagram. This resource diagram is used to describe various resources and their roles.

Figure 2.7 Example of a resource diagram with a resource and resource roles

As a next step, you can assign resource roles to activities in a process or collaboration.

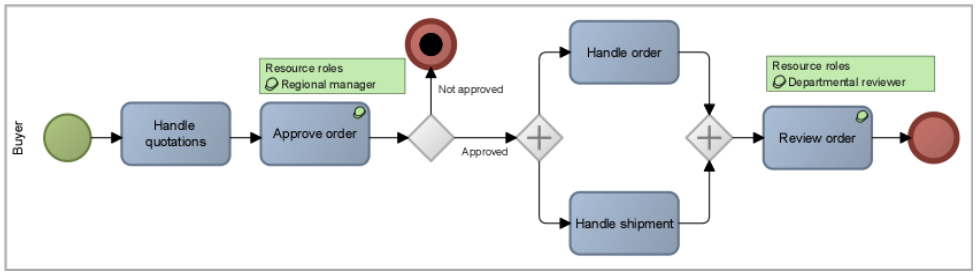


Figure 2.8 Example of a process with resource roles assigned to activities

Difference between resources and participants

A resource is different from a participant. Depending on how the two concepts are used, the difference may be unclear, but in general it is the case that a resource defines the resource that is responsible for the execution of an activity. A participant is defined within the context of a collaboration and only has meaning within this context. A participant is responsible for the execution of a single process (pool) within the collaboration. A process is one of the participants, the other pools in the collaboration form the other participants.

2.1.6 Items

Although items are not defined as part of the BPMN specification, they are included in Enterprise Studio as an extension to it. Items can be used to describe different types of data storage and information flows that are used

in a model. A separate diagram/visualization for items is not available, but they can be added to diagrams that include data stores and message flows. You can create items at the same time while creating a data store or message flow. Alternatively, you can create and add items to the model browser prior, and then use them later when creating a data store or message flow. Once an item is created and available in the model browser, it can be reused indefinitely in process and collaboration diagrams that are part of a BPMN model.

The process in the figure below includes a data store "Order status", as well as two message flows "Order" and "Status update". The items are shown as the names of these objects, while the actual item objects can be found in the model browser.

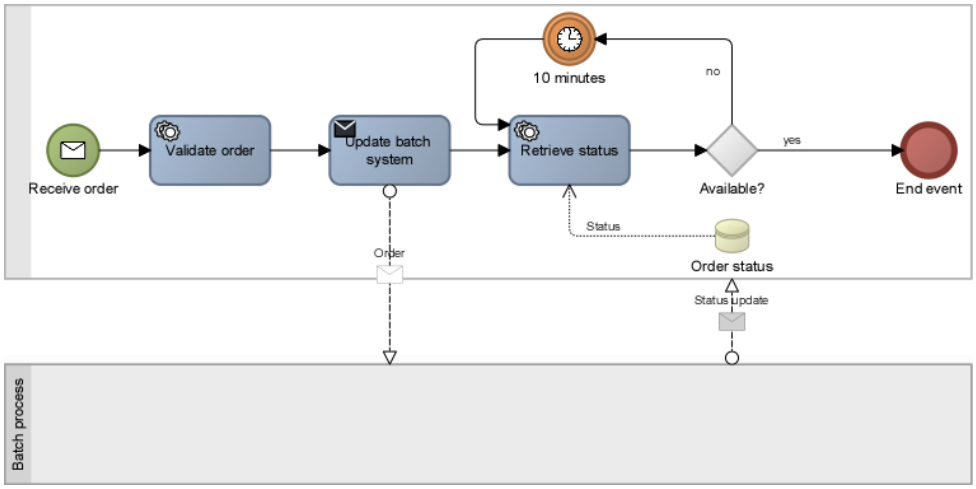


Figure 2.9 Example of the use of a data store and message flows, further detailed by using items

2.1.7 Global tasks, partners and other domain

The following data can also be used in BPMN modeling:

- *Global tasks* to denote generic tasks which can be invoked by a process or collaboration, but do not need to be defined in more detail.

- *Partners* to model partner entities and partner roles. These partners represent the participants in a collaboration.
- *Other domain* for the import of objects from domains not covered by BPMN, which can be related to BPMN objects.

Of the diagrams and data mentioned, the focus is set on the process, collaboration, choreography, resources and items. The others will not be discussed here.

2.2 BPMN modeling levels

BPMN models can be created at different levels of abstraction. Enterprise Studio distinguishes three levels for it: *Introductory*, *Descriptive*, and *Analytical*.

Introductory level

The Introductory level is the lowest of the three levels. At this level you can map out processes greatly simplified. You can only work with tasks and gateways. Models at this level are very suitable to map out the main lines.

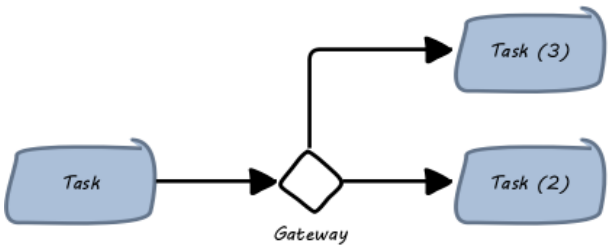


Figure 2.10 Example of a process diagram at Introductory level

Descriptive level

This level is for descriptive modeling and is suitable if you want to ignore event-driven behavior. This level corresponds to BPMN level 1 (Descriptive modeling). A model at descriptive level ensures a good understanding of how business is going. At the descriptive level you work with the fundamental BPMN concepts like swim lanes, lanes, tasks, sub-processes, and sequence

flows. The complexity of managing flows and patterns of events are not addressed here.

Analytical level

The Analytical level as we know it in the tool is a combination of BPMN levels 2 and 3 (Analytical and Executable modeling). It includes level 2 completely and level 3 partially.

At the analytical level you model in detail, all the paths are shown, including the exceptions paths. At this level you also work with events and patterns for decisions, merges, and exception paths. Diagrams modeled at this level must also be valid according to the rules of the BPMN standard.

2.2.1 Choosing a modeling level

Before you start modeling you determine the level at which you want to model. You can set the level of you choice in the open diagram. Do this by clicking the control in the upper left corner of the diagram. Next, you click the stairs symbol and subsequently the desired level:

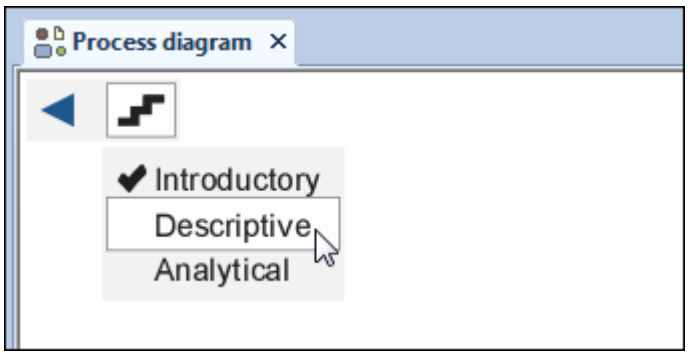


Figure 2.11 Selecting the modeling level in a diagram

2.2.2 Viewing models of other levels

Although not all concepts are available at every modeling level, they all are viewable on each level. If you for example open a process that has been modeled at a higher level than the level you are viewing the process at, then all objects and relationships present in the process will be simply shown. You

just cannot edit these objects and relationships, or add new ones. In the example below you can see the differences in model presentation between the three levels.

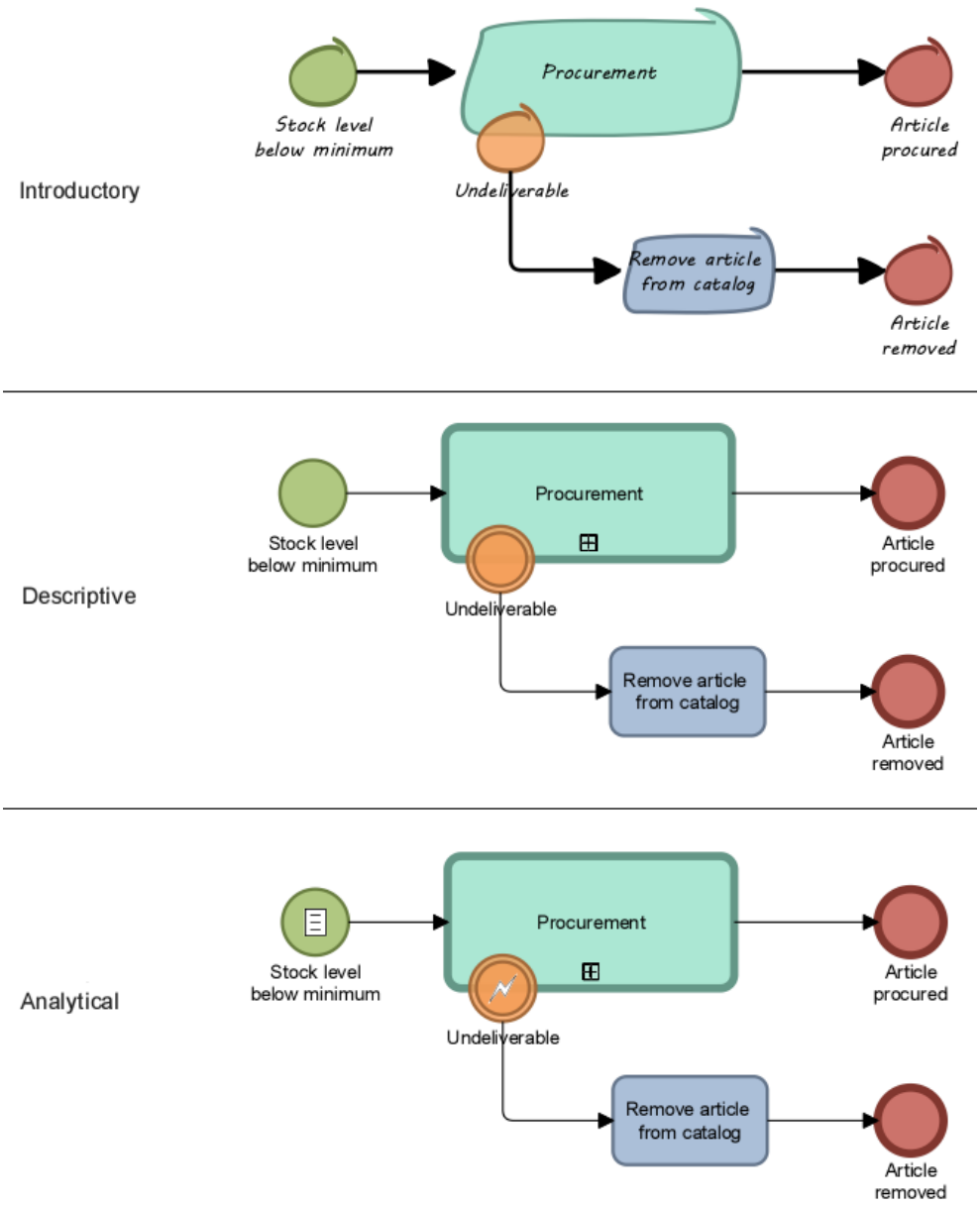


Figure 2.12 Display of a process at different modeling levels

2.3 Modeling a BPMN process

2.3.1 Modeling in a process diagram

When modeling a process in a process diagram, two alternative ways of working can be followed:

- First you create one or more lanes, then you position tasks, events, gateways etc. within them.
- Or you first model the tasks, events, gateways etc. in the process and subsequently drag a lane around them.

By placing the process elements you build the framework of the behavior that is modeled. The exact place and size of the process elements are not of direct importance. You can easily change them later. You can later also add or remove process elements.

By using the controls at the top of a lane you can create sub-lanes, add new lanes above or below an existing lane, or remove a lane.

2.3.2 Modeling in a collaboration diagram

There are two ways to include processes in a collaboration:

- Add existing processes you have created separately at an earlier time, by dragging them from the model browser onto the collaboration diagram. For each process that is added in this way, a pool will automatically be created in the diagram, containing the contents of the process.
- Create new pools in the collaboration diagram and subsequently add objects and relationships in them to form processes. Within a pool, lanes can be created and within them objects and relationships can be added to organize the objects that are part of the process. For every created pool in the collaboration, a separate process diagram will be created automatically, which can be found in the model browser, in the **Processes** folder below the model.

Whichever way of working is applied, processes will exist and can be edited as separate components within the model.

2.3.3 Modeling in a choreography diagram

A choreography uses a collaboration as starting point. In the collaboration diagram you look where the exchanges of information between the participants (pools) are located. Based on these exchanges you add choreography tasks to the choreography diagram. Each choreography task is a set of one or more information exchanges between two participants. The names of the participants are placed in the task: one at the top, the other one at the bottom.

Subsequently, you finalize the choreography diagram to form a process by adding events, gateways, and sequence flows.

2.3.4 Starting point and end point of a process

When modeling a process you start with drawing a start event (starting point) and an end event (endpoint). They delimit the process, they indicate the beginning of the process and the end or result. Between these events, you are modeling the process. Start events do not have incoming relationships, end events have no outgoing relationships. Intervening events can be set with intermediate events and boundary events. Boundary events are not available in a choreography.

2.3.5 Activities

A process consists of different activities. If all activities are executed in the correct order, they together form the process, leading to the result or product. Activities can be drawn by using tasks and choreography tasks. To properly express and consistently formulate actions, you usually name the activities in the form verb-noun, for example "Submit claim" or "Create claim file".

2.3.6 Relationships

Sequence flows, message flows, and associations can be added to a diagram by drawing lines between the objects. There are two ways of working to add

flows and associations.

Method 1: In the **Create** window pane, click the **Sequence flow**, **Association** or **Message flow** relationship. Next, click the first of the two objects in the diagram that should be connected and draw the appearing line to the other object.

Method 2: Click the first of the two objects in the diagram that should be connected. Subsequently click the control in the lower right corner of the object for drawing a relationship and drag the line to the other object. Depending on the type of relationship that can be drawn from the object, the control represents a sequence flow  , message flow  or association  .

3 Creating BPMN models

In most cases, you start with modeling processes. Once you have completed two or more process descriptions, you can integrate them in a collaboration.

You can first create the individual process diagrams separately and then include them in a collaboration diagram, but it is also possible to create the processes directly as part of a collaboration diagram. Once you have finalized a collaboration diagram, you can use it as a starting point for a choreography.

3.1 Creating a new BPMN model

Creating a BPMN model starts with creating a new model. By choosing a specific model, you can already take into account what you want to register. You can always add additional diagrams later to the model if desired. Determine what you want to do and perform the relevant procedure:

Model a process in a process diagram

- On the **File** tab, click **New > BPMN > 2. Empty model with process diagram**.

  The model browser shows a new model package with a BPMN model, and a new empty process diagram is opened in which you can start modeling.

Model a process in a collaboration or choreography diagram

- On the **File** tab, click **New > BPMN > 1. Empty model with collaboration diagram**.

  The model browser shows a new model package with a BPMN model and a new collaboration diagram with a pool is opened in which you can start modeling.

For modeling a choreography you can add a choreography diagram later.

Start with an empty model

- On the **File** tab, click **New > BPMN > 3. Empty model**.

  The model browser now shows a new model package with an empty BPMN model. After that you can add the desired diagrams to the model package.

3.1.1 Adding a new diagram

1. In the model browser, right-click the BPMN model or a folder within the model, point to **New** and click the diagram of your choice: **Collaboration diagram**, **Choreography diagram**, **Resources diagram**, or **Process diagram**.

   The new diagram appears in the model browser, right below the model.

2. Click the newly added diagram in the model browser to open it.

3.2 Adding content to a BPMN diagram

Objects and relationships that can be drawn in a diagram are based on the concepts available to the open diagram.

Objects and relationships can be added in different ways, by using:

- the **Create** pane
- the quick-create pop-up window
- the model browser
- the quick-create object controls
- the smart connector
- the context menu

For detailed information about the possible ways to add objects and relationships, please refer to "Creating objects and relations" in the Enterprise Studio User Guide (PDF).

3.2.1 Concepts for modeling

The concepts for modeling in BPMN can be divided into a number of main categories. These are:

- *Flow objects*. These include events, various activities, and gateways. They are the most important graphical elements that can be used to define the behavior of a process.
- *Connecting objects (relationships)*. These include sequence flows, message flows, and associations. They are used to connect flow objects to one another, and to other information.
- *Swimlanes*. These are the pools and lanes. They can be used to group the primary modeling elements.
- *Data*. These are the data objects, data input, data output, and data stores. They represent the information/data and data structures that can be used within a process.
- *Resources*. These are the resources and resource roles. They represent the sources to which activities in a process can refer.
- *Artifacts*. In the BPMN language also distinguishes the artifacts *group* and *annotation*. Because of the general nature of these elements, there are no specific BPMN elements defined in Enterprise Studio to cover these. Instead, the graphic shapes *group* and *comment* are used, which are present in the **Create** window and are available to every model type.

What concepts are available, depends on the type of diagram you are modeling in, but depends also on the level at which you are modeling. The lower the modeling level, the less concepts will be available.

Appendix A shows a detailed overview of the available concepts.

> **Note:** Some types of objects that are used in BPMN diagrams, like *data stores*, behave like objects in views. This is essential for understanding their behavior in modeling. For more information, please refer to "Difference between diagrams and views" in the Enterprise Studio User Guide (PDF).

In addition to the concepts that are specific for the modeling language or method, there are several graphic shapes that can be included in a diagram or view. These graphic shapes are generic and available in each modeling language and method in Enterprise Studio.

The graphic shapes are discussed in the Enterprise Studio User Guide (PDF).

3.2.2 Available controls

The objects in a BPMN diagram have controls that can be used to characterize and mark them, or provide them with specific information. These controls appear when you click an object. They are usually above and below the selected object. Example:

Figure 3.1 Example of a BPMN object control

When you move the mouse pointer over a control, it will be highlighted. Click the control to add a type or marker, or to provide it with information. Appendix B shows a complete overview of all available controls.

> **Note:** Which controls are available for the objects, depends on the level at which you are modeling. The lower the level, the less controls are available.

TIP: By selecting multiple objects at the same time and then clicking the control on the first selected object (with the light green border), you can typify or mark multiple objects at once.

3.2.3 Object types and markers

Some of the BPMN objects can be further specified when used in a process by assigning types or by characterizing the objects when they are in the diagram. Task and events for example can be assigned by a type, and when an activity is connected to a sub-process, the object icon will be marked with a plus sign.

Assigning object types and marking objects

In order to assign a type to an object, select the object in the diagram and click the control at the top edge of the object. Next, click the type of your choice. After you have assigned a type or property to the object, it is shown with a symbol in the object icon. Example:

Figure 3.2 Display of the type in tasks and events

Marking objects is done using various controls. Appendix C shows a complete overview of all available object types and markers.

Please note that it is not mandatory to assign object types, or add properties to objects. Whether or not you are using it depends on the amount of information you intend to capture in the model.

> For a more detailed explanation and application of the BPMN elements, their characterizations and properties, please refer to the official BPMN 2.0 specification, or visit the website of the Object Management Group (OMG).

Removing object types and markers

To remove the type of an object, select the object, click the control and subsequently click the activated types, which are shown in red. After you have switched off a type, it turns black again.

Removing a marker can be done in various ways. Which way, is dependent on the way the marker has been set or has appeared on the object.

3.3 Relationships

3.3.1 Defining a sequence flow as default or conditional

An added sequence flow shows only the behavior of the activity, it is not affected by any conditions. When a sequence flow is used in a split or join the flow can get a characteristic assigned. This characteristic determines when a flow is ended.

A flow should be defined as *default* when it is used to create a default branch that is only used if the result of all other outgoing conditional flows is negative. A default flow has a diagonal slash to the beginning of the connector: .

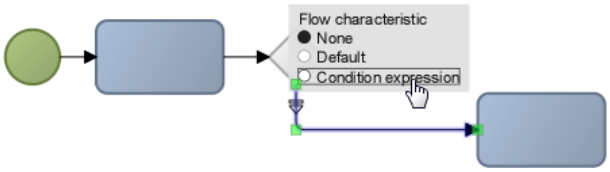
A flow should be defined as *conditional* when it is used in a split to indicate when it should be followed or not. In that case you add a condition to the flow. The flow will then only be followed if the result of the condition is positive. If the conditional flow is outgoing from an activity, then the flow will have a mini-diamond at the beginning of the connector: . However, there must be another conditional flow present outgoing from this

activity in order to show the mini-diamond. If the conditional flow is outgoing from a gateway, then the flow will not have a mini-diamond.

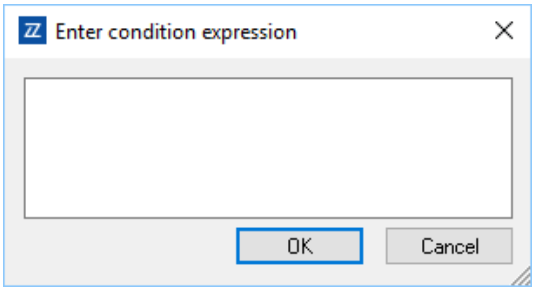
To define a sequence flow as default or conditional, do the following:

1. Click the sequence flow in the diagram.

2. Click the control that appears on the connector and click **Default** or **Condition expression** in the **Flow characteristic** pop-up window.



3. If you have chosen **Condition expression**, then the **Enter condition expression** window will appear. Enter the condition for the sequence flow in this window and click **OK**.

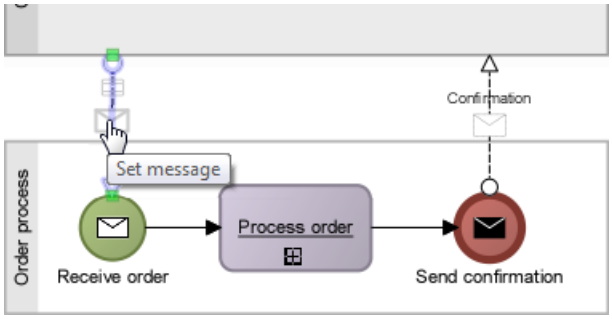


If you at any time want to change the characteristic of a sequence flow, then click again and select the characteristic of your choice in the pop-up window. Click **None** if you want to remove the condition again. If you want to change a condition that has been set at an earlier time, then click the already selected **Condition expression** option button in the pop-up window. The window for entering the condition will appear.

3.3.2 Adding content description to a message flow

Adding message flows between pools, activities, or events alone in order to show the information flows, does not make clear *what* information is actually exchanged. The contents of a message flow, for example a confirmation, assignment, money or an invoice, can be explicitly described using items. The way of working is as follows:

1. Select a message flow and click the transparent envelope that appears on the flow. The envelope represents the message (information) that is exchanged.



2. Next, a window pops up that displays the content of the active model package. Navigate to the items folder, select the message flow item of your choice and click **OK**.

   If the desired item is not (yet) available, click **Cancel**. Next, a new window pops up in which a name for the new item can be entered. Enter the name and click **OK**.

The item is now connected to the message flow. An envelope with the name of the connected item is depicted on the message flow.

If you want to, you can place the name of the connected item in a different position next to the envelope, or hide the name or even the entire envelope.

3.3.3 Configuring the display and nature of a connected message flow item

Configuring the label position of a connected message flow item

By default, the label with the name of the connected item is displayed on the right side of the envelope. However, the label can also be displayed at another location next to the envelope. In order to change the label's position, click the envelope. You will then see little blocks and arrows appearing on the different sides of the envelope. Click a block or arrow at one side of the envelope to place the label on this side.

Figure 3.3 Configuring the label position

Configuring the display of the name of a connected message flow item

By default, the name of the connected message flow item is displayed next to the envelope. If you want to hide the name, then click the envelope and subsequently click the block or arrow at the side where the label is positioned. The label will disappear. Click one of the blocks or arrows once again and the label will reappear on the same spot.

Configuring the nature and visibility of a connected message flow item

The connected message flow item is displayed with an envelope on the message flow. This envelope can be used for displaying the nature of the message flow. You can use it to define a message as initiating or non-initiating ("responsive").

In most cases collaboration diagrams contain multiple information exchanges. The nature of these information exchanges can be different in terms of initiative, information can be initiating or non-initiating (information

returning as response). Message flows representing these exchanges of information can be defined as initiating or non-initiating. Initiating message flows are usually represented by a white envelope, non-initiating message flows by a gray envelope.

By default, the envelope is shown in white. To turn the envelope gray, select the envelope and click it once again. That way you can determine whether the message flow is initiating or non-initiating.

However, using colors for the envelope is not mandatory. If you do not want to visualize the nature of message flows, all envelopes should be of the same color. The common color to use then is white. You can also choose not to show the envelopes at all. In that case click the envelope twice after selecting it. It will disappear. The envelope acts as a sort of toggle for its display. By clicking it repeatedly you can change the envelope's state: white, gray or invisible.

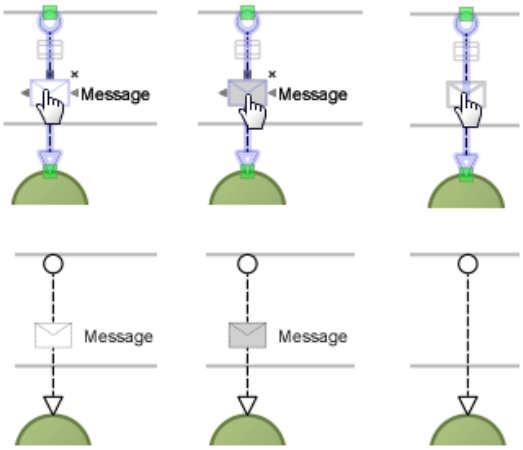


Figure 3.4 Nature and visibility of a message flow item

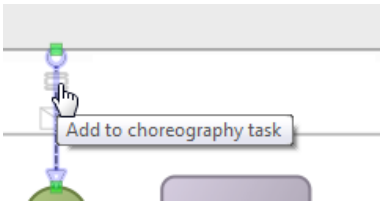
3.3.4 Removing a connected message flow item

To remove the connection between a message flow and item, first click the envelope in the flow and subsequently click the small cross in the upper right corner of the envelope. The connection with the item is now removed. The item itself is *not* removed, it remains to exist and can be found with the items in the model browser.

3.3.5 Referring to choreography tasks from message flows in a collaboration

After creating a choreography based on a collaboration, references between the collaboration and (sub)choreography tasks can be defined to indicate that the exchange of information between pools is further detailed in the choreography. The way of working is as follows:

1. In the collaboration diagram, select the message flow and click the choreography task icon that appears near the starting point of the message flow line.



2. Next, a window pops up that displays the content of the active model package. Select the desired choreography and click **OK**.

The collaboration diagram does not visualize the connection between the collaboration and choreography task. But it is possible to visualize the connection in the choreography diagram.

3.3.6 Visualizing message flow references from a collaboration in a choreography

Once the choreography of a collaboration has been mapped out, it is possible to create references from the message flows in the collaboration, to the (sub)choreography tasks in the choreography (see also Message flows in Collaboration). Subsequently, these references can be shown in the choreography. This is done by selecting a (sub)choreography task and then clicking the control that appears on the object. (Please note that this control is only available if a reference from a message flow to the

choreography task exists.) Next, the name of the linked message flow from the collaboration will be shown with an envelope:

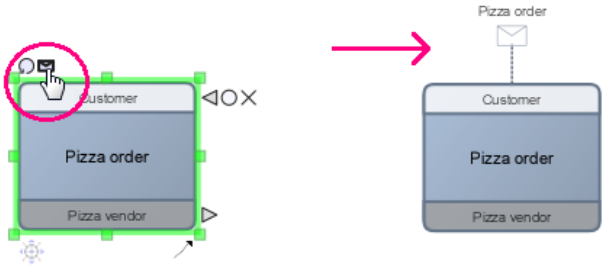


Figure 3.5 Show message flow reference

In case you want to align the name of the referenced message flow at an alternative side of the envelope, click the envelope and subsequently click one of the blocks and arrows that appear at all sides of the envelope, corresponding with the desired location of the text.

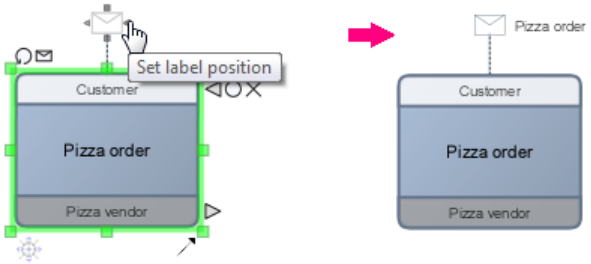


Figure 3.6 Configuring the position of the message flow name

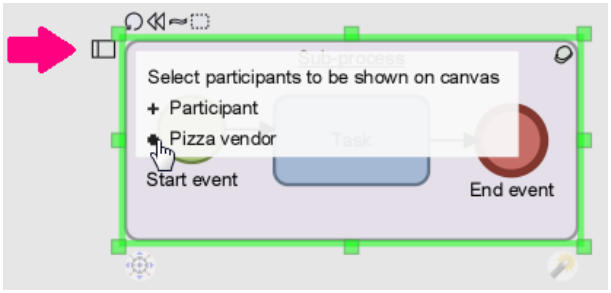
3.3.7 Drawing message flows between sub-process elements and process elements of other pools

If there is a collapsed sub-process present in your collaboration, it is possible to draw message flows between this sub-process and process elements of the other pools in the collaboration. However, these relationships remain on collaboration level; it is not clear to which element of the sub-process the relationship belongs. To make this clear, then you can draw message flows at sub-process level. In that case you draw the desired message flows between the elements in the sub-process and the elements in

the other pools. To do this, first make the pools you want to relate to visible in the sub-process. Next, draw the relationships. To do this, do the following:

1. Open the collapsed sub-process in the collaboration by clicking the plus sign ⊞ in the object. The diagram containing the opened sub-process will now appear.

2. In the opened diagram, click on the sub-process and subsequently click the ▭ control on the left side of the object.



3. In the pop-up window, indicate which pools (participants) of the collaboration must be displayed in the sub-process diagram. To do this, click the plus sign in front of a participant. The selected participant will be displayed directly in the diagram. If desired, you can rearrange it in the diagram to keep a good overview.

4. Draw the desired message flows between elements in the sub-process

and elements in the present pools.

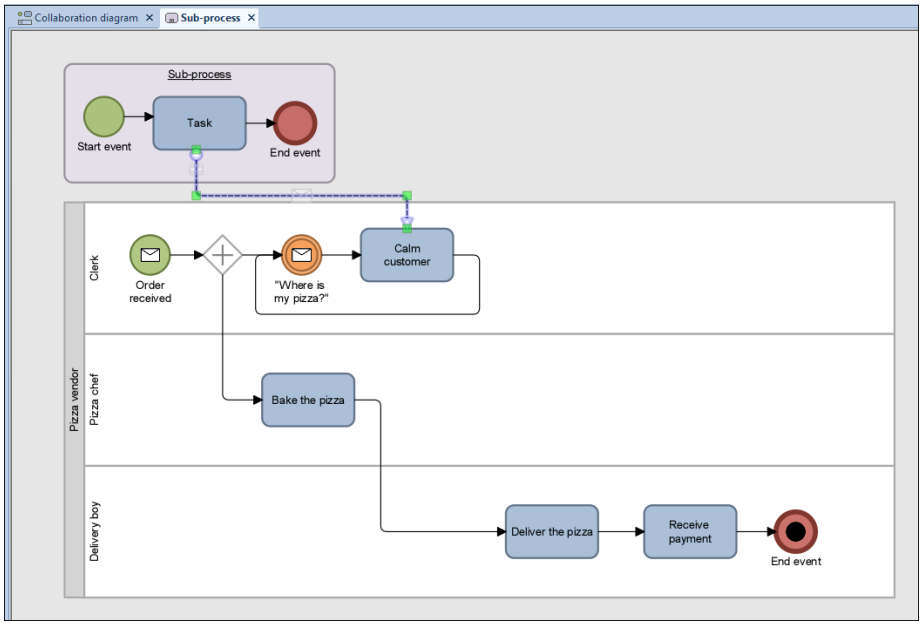


Figure 3.7 Drawing message flow between element in sub-process and element in other pool.

When you, after drawing the message flows, return to the collaboration diagram, you will see the newly created message flows too.

To remove a selected pool from the diagram, click the sub-process object once again and click the control. In the pop-up window, click the cross sign in front of the pool. Any message flows that were drawn between the sub-process and the pool will continue to exist. You can see this in the collaboration. However, you cannot see which elements of the sub-process the message flows belong to. In order to be able to see this you will need to reactivate the pool in the opened sub-process.

3.3.8 Setting the direction of an association

It is possible to set the direction of an association in order to further specify the association. A drawn association does not have a direction by default. To set the direction of an association, first click the line and subsequently click the control that appears on the line. Click it repeatedly until the desired direction is displayed.

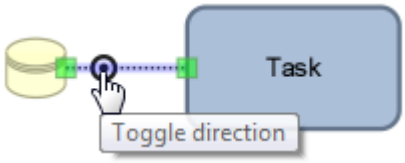


Figure 3.8 Setting the direction of an association

3.4 Events

3.4.1 Linking a boundary event to an event from a sub-process

Boundary events are attached to the boundary of an activity or sub-process. As a result, they may already be activated while the activity or sub-process is still being executed.

If the boundary event is an *error* boundary event and attached to the boundary of a sub-process or call activity with a sub-process connected to it, you can generally assume that the source of the error event is indicated by an error end event in the sub-process. We call this an *error throw-catch*: the error signal *thrown* by the error end event is *caught* by the boundary event, after which the boundary event is executed.

If the boundary event is an *escalation* boundary event, the boundary event can be linked to an escalation end or intermediate event in the sub-process in order to set an *escalation throw-catch* between the sub-process and the boundary event.

To link a boundary event to an error or escalation event from the sub-process, click the boundary event that is attached to the boundary of the sub-process or call activity. Next, click the ➡ control and select the desired event in the window that pops up. Example:

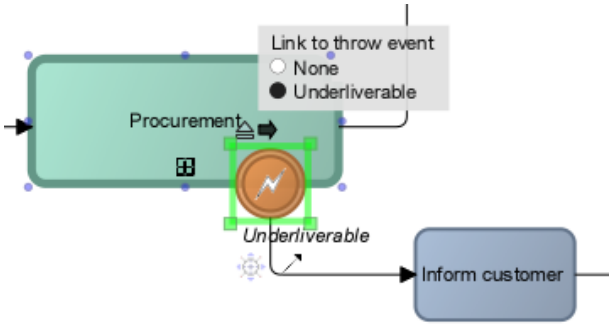


Figure 3.9 Connecting a boundary event to an error end event

After selection, the name of the connected event is displayed in italics below the boundary event. In order to remove the connection with the event of the sub-process, click **None** in the pop-up window.

3.4.2 Defining events as multiple or parallel multiple

One of the characterizations you can assign to an event is that it is *multiple* or *parallel multiple*. If an event is (parallel) multiple, there may be multiple events (triggers) that can activate the event, or multiple results an event can deliver.

In order to define an event as multiple you do not assign just one but several types to the event. Once you have selected more than one type, a pentagon is displayed in the event icon to indicate the event is multiple.

Figure 3.10 Events with multiple types

To define an event as *parallel* multiple, open the properties window of the event, and click the red cross next to the **parallel multiple** property on the **Properties** tab. A green check mark ✓ will appear next to the property.

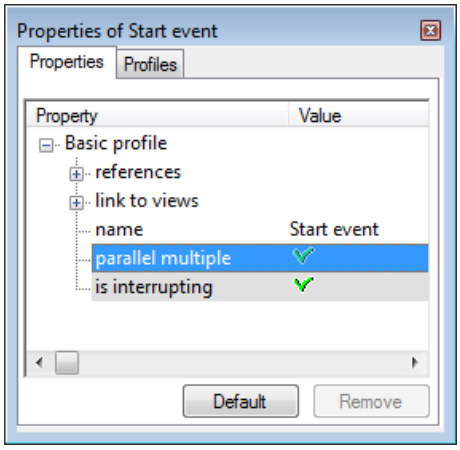


Figure 3.11 Defining an event as parallel multiple at the event properties.

A plus sign is now displayed in the event icon to indicate that the event is parallel multiple.

Figure 3.12 Parallel multiple events

3.5 Activities

3.5.1 Linking participants to a choreography task

The two participants of a choreography task are placed in the upper and lower band of the task. The bands are the narrow gray edges on the top and bottom of the task.

In order to add a participant, select the choreography task and click the ▷ control at the right side of a band. Then click the ○ control. Select the participant in the window that pops up. A pool in a collaboration diagram should be selected here, since the choreography maps out the information exchanges between pools in a collaboration. The second participant is documented following the same steps.

Since there is always an initiating party to a choreography task, one of the two participants should be defined as initiating. To do this, click the ☆ control to the right of the band, which will appear after you have clicked the ▷ control.

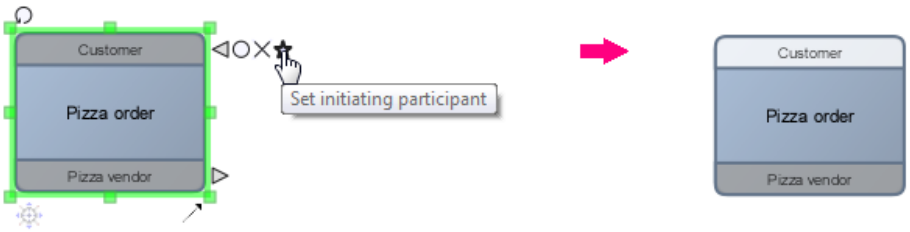


Figure 3.13 Defining the initiating participant

The band that represents the initiating participant has a lighter color shade, while the opposing band has a more dark color shade. There can be only one initiating participant; the moment you define the second participant as the initiating party, the first participant will automatically be set back to non-initiating.

3.6 Resources

3.6.1 Defining resources and resource roles

The resources diagram is used to describe the resources and their roles that are relevant for (referred to from) processes and collaborations.

In order to connect a resource role to a resource, select the resource role and click the control at the top of the object. Next, select the resource in the appearing pop-up window. As a result the name of the linked resource is displayed in the resource role.

Figure 3.14 The name of the linked resource displayed in the resource role

3.7 Items

There is no separate diagram available for items in Enterprise Studio; items can be used in diagrams that include message flows and *data stores*. There are two alternative ways of working with items:

1. First you create the items in the model browser and subsequently add them to the message flows and data stores you have included in the process or collaboration diagram.
2. You add message flows and data stores to a diagram and at the same time create the items.

Both ways of working can be used interchangeably. Once an item is created, it can be reused indefinitely in process and collaboration diagrams within a model.

Adding items

To create items in the model browser, do the following:

1. In the model browser, right-click the model, point to **New** and click **Items**. A new element called **Items** is added to the model browser, one level below the model.

2. Right-click the newly created **Items** element, point to **New** and click **Data store** or **Message**, depending on what you want to add.

3. Give the newly created a name and press **Enter**.

4. Repeat steps 2 and 3 for each item to be added to the model.

How to use the created items as you add message flows and data stores is discussed in the relevant sections.

4 Relations between BPMN and other modeling domains

BPMN models can be integrated with models from other domains, like entity-relationship (ER) models and UML models by linking data objects in processes and collaborations to ERD entities or UML classes. Not only the general data objects, but also data inputs and data outputs can be linked to ERD entities or UML classes.

To be able to link data objects to ERD entities and UML classes, the model where the ERD entity or UML class is located, must be part of the model package where the BPMN data object is located.

A data object can only be linked to one ERD entity or UML class, not to both.

4.1 Linking a data object to an ERD entity

To create a link between a BPMN model and a ER model, ERD entities must be present in the ER model.

1. Open the process diagram or collaboration diagram containing the data object you want to link to an ERD entity.
2. Click on the data object and click on the control on the left side of the object.
3. In the **Select ERD entity** window, select the desired ERD entity from the ER model and click **OK**.

The data object now refers to the chosen ERD entity.

4.2 Linking a data object to a UML class

To create a link between a BPMN model and a UML model, UML classes must be present in the UML model.

1. Open the process diagram or collaboration diagram containing the data object you want to link to a UML class.

2. Click on the data object and click on the control on the left side of the object.

3. In the **Select UML class** window, select the desired UML class from the UML model and click **OK**.

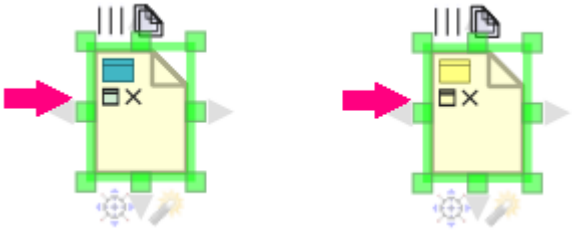
The data object now refers to the chosen UML class.

4.3 Navigating to a linked ERD entity or UML class

Once a link between a data object and an ERD entity or UML class has been created, it is possible to navigate directly from the data object to the linked entity or class. To do this, click on the data object in the diagram and then click on the or control in the object. Next, the view (ERD) or diagram (UML) containing the object is opened.

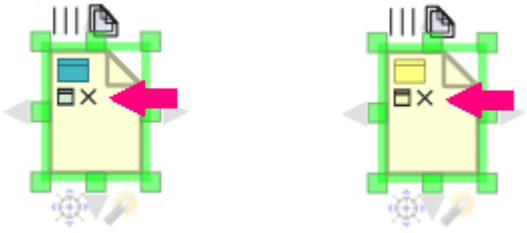
4.4 Changing the link to an ERD entity or UML class

To change an existing link to an ERD entity or UML class, click on the data object in the diagram and then click on the or control in the object. Next, you can select another entity or class.



4.5 Removing the link to an ERD entity or UML class

If you want to remove the link to an ERD entity or UML class, click on the data object in the diagram and then click on the cross sign in the object.



5 Import and export of BPMN model data

5.1 Importing BPMN model data

It is possible to import BPMN model data into Enterprise Studio.

5.1.1 Supported file formats

The following file formats are supported:

BPMN

The Business Process Modeling Notation file format is a standard file format that can be used to exchange BPMN models between tools that create or interpret BPMN models.

When importing BPMN files, the most practical way to do this is to first export one or more existing BPMN models. After that the necessary changes are made to the export file. Ensure that the structure of the file is not changed! The modified export file can be imported again.

Visio

Visio XML export files can be imported in BPMN models. You can import Visio XML files with the extension .VDX or .VSDX (Visio 2013).

To be able to use Visio import, you first need to save your Visio model as XML file. Visio will create a file with extension .VDX or .VSDX. Also, within Visio you need to use a (the standard) Visio scheme. This Visio scheme is like a metamodel specification.

If you save the file, this specification is added to the file. Enterprise Studio uses this specification to determine which types of objects must be created while importing. This transformation is defined in the file **VisioToBPMNTypeMapping.xml** in the folder MetaModels\BPMN\ImportTypeMappings.

If you use a non-standard Visio scheme, you may need to update this transformation file. For this, please contact your application manager or BiZZdesign.

5.1.2 Importing files

To import a BPMN or Visio file, follow these steps:

1. In the model browser, select the model package or model in which you want to import data.

2. On the **BPMN** tab, in the **Import** group, click the import type of your choice: **BPMN** or **Visio**.

3. In the **Import objects** window, select the file you want to import and click **Open**.

4. Visio import only:

    a. In the **Choose Diagrams** window, select the diagrams you want to import and select the diagrams/views where they must be imported in.

    b. Optional: **Transformation to use during import**shows the transformation file that is used for the Visio import. If the correct file is located elsewhere, select the correct location and file.

    c. Optional: By default all imported objects are added as new objects to the model the are imported in, even if objects (of the type) already exist in the model. New objects will be created with the same name as the existing ones, including a sequence number between brackets. If you select the **Match names with existing objects** check box, the imported objects will be matched with the existing objects. If an object (of the type) already exists in the model, it

remains the same and the new object will not be imported.

d. Click **OK**.

Including referenced models in the import

The models in a BPMN file may contain references to models that exported in other BPMN files. When importing, you therefore have the possibility to not only import the selected file, but also any other files with models the selected file refers to. If you do not want to, then only the selected file will be imported.

Condition for the import of these other BPMN files is that the files must be placed in the same folder as the selected BPMN file. If they are not there, they will not be imported.

Import log file

When importing BPMN files, the tool automatically generates a log file and stores it. The log file can be opened after import. Th file contains a list of the imported data.

5.2 Exporting BPMN model data

It is possible to export your BPMN models from Enterprise Studio. You can export a single model, but also multiple BPMN models at the same time to one file.

The Business Process Modeling Notation file format (BPMN) is a standard file format that can be used to exchange BPMN models between tools that create or interpret BPMN models.

To export BPMN model data, follow these steps:

1. In the model browser, select the model(s) you want to export.
2. On the **BPMN** tab, in the **Export** group, click **BPMN**.

3. In the **Export to BPMN file** window, specify the location and folder where the export file must be stored and click **Save**.

6 Analysis functions for BPMN models

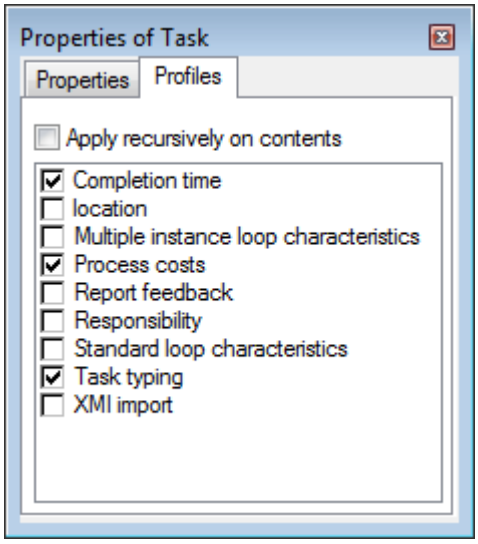
The BPMN analysis functions allow you to get more information on your BPMN diagrams, ranging from the cost of a process path to detecting loops in the process. Analyzing a BPMN model has several advantages, ranging from the detection of potential problems in the design of these models to offering support in making decisions because of the possible results.

6.1 Required profiles

For some model analyses to perform correctly you must first establish certain properties of the elements in the model using a profile that is associated with the element. The elements that have the required profiles are mainly tasks, sub-processes and relationships. The properties must be filled in properly for a correct and accurate output of the analysis.

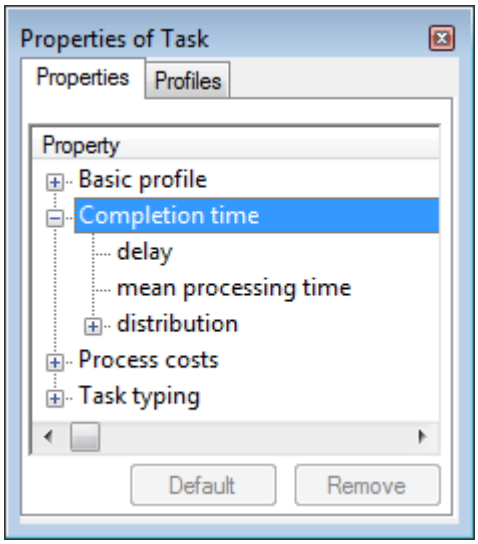
The properties that must be defined are associated with a profile. Each analysis indicates which profiles are required and which can additionally be used. To define the properties of a profile:

1. In the open diagram, select the element you want to link to a profile and subsequently click the **Profiles** tab in the properties window.



2. Activate the desired profile by checking the box in front of this profile.

3. Click the **Properties** tab and define the required properties.



4. Repeat steps 1 to 3 for each element in the process that must be included in the analysis.

6.2 Messages

When performing an analysis, messages may appear shown in the message window. This may include the following types of messages:

- Error message/warning. In these messages the user is warned about properties that are not defined properly.
- Navigable messages. In some analyses the output contains navigable messages. When a message is related to multiple objects, a message line contains more information than is visible. To view the complete information, use the light blue arrow buttons  at the bottom of the message window. Clicking them will navigate you through the elements in the diagram.

6.3 Available analyses

6.3.1 Transfer of work

Required profiles: None.

Transfer of work exists when resource roles change in the transition from one task to another, but also when two consecutive tasks belong to different lanes.

This analysis shows an overview of consecutive tasks that have different resource roles, and consecutive tasks in different lanes.

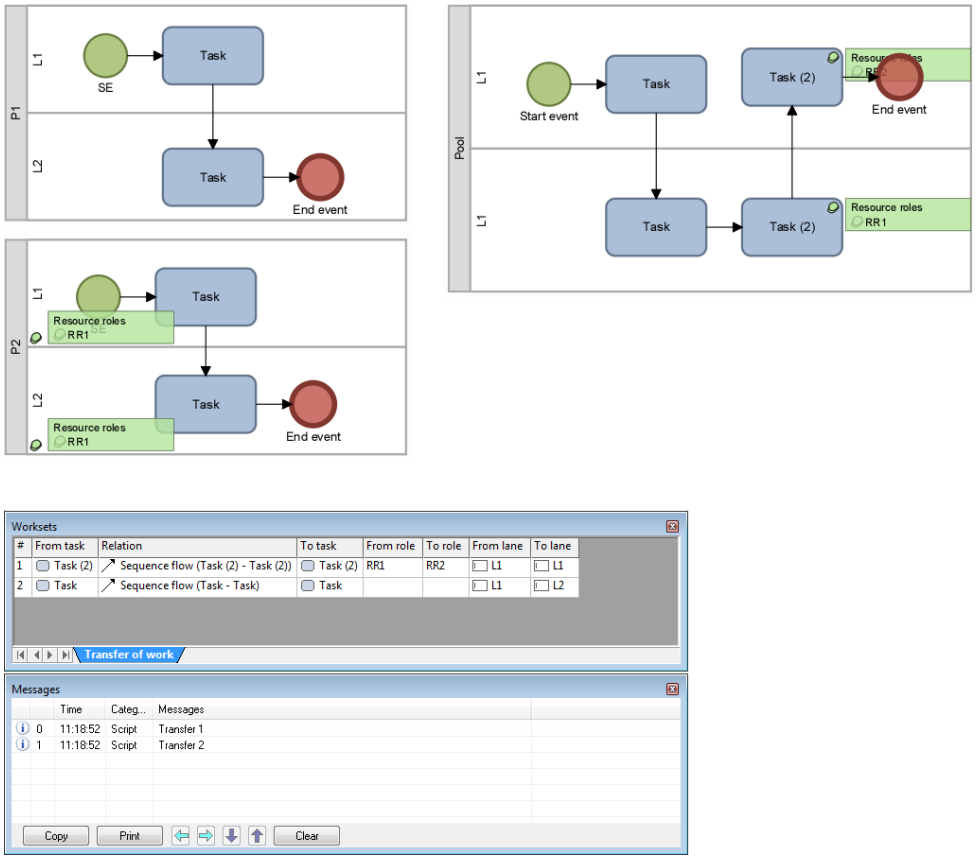


Figure 6.1 BPMN analysis "Transfer of work"

6.3.2 Critical path

Required profiles: *Completion time* and *Probability and delay*.

Additional profiles: *Process costs*.

This analysis shows the paths with the maximum completion time, including and excluding consideration of the probabilities. This analysis does not take into account the costs of the paths. Indicated costs however will be displayed in the output window when they are entered.

The elements present in each critical path can be navigated in the message window.

Maximum critical path

The path of which the maximum completion time has been calculated, is returned without taking into account the probabilities of the specified path followed.

Weighed critical path

In contrast to the maximum critical path, here the chances of any decision (gateways, intermediate events, etc.) are calculated with the completion time of each path. As a result, the maximum path is given that meets the criteria.

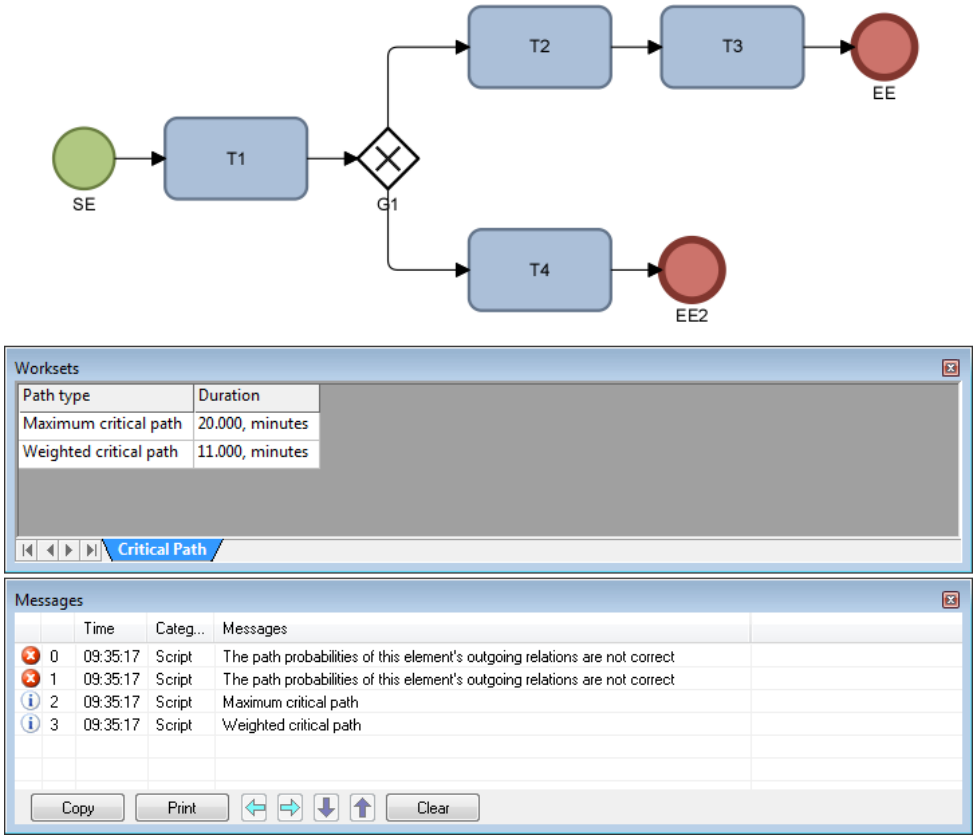


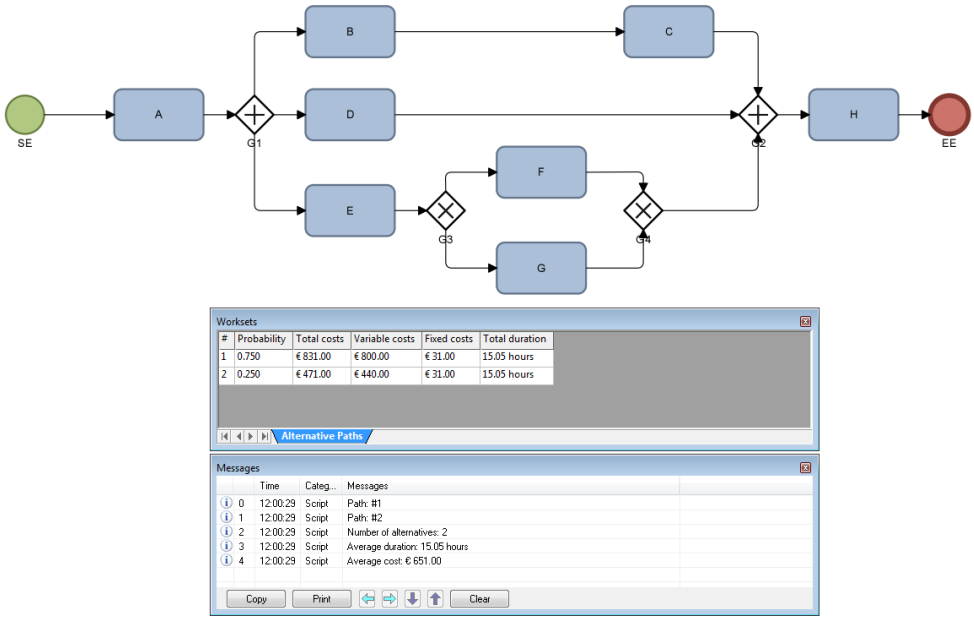
Figure 6.2 BPMN analysis "Critical path"

6.3.3 Alternative paths

Required profiles: *Completion time* and *Probability and delay*.

Additional profiles: *Process costs*.

This analysis follows the same criteria and output as in the critical path analysis, but this analysis shows an overview of all the possible paths in the model with their processing time and costs.



6.3.4 Task types

Required profiles: *Task typing*.

This analysis shows for each task type the number of tasks and their total processing time. All properties for the typing of a task are analyzed and for each property the results are shown in its own pie chart. The available typing properties are:

- adding value (true/false)
- managing (true/false)
- checking (true/false)
- supporting (true/false)
- informing (true/false)

- legally determined (true/false)
- process type (primary, secondary, tertiary)
- execution (manual, supported, automated)
- automatable (true/false)

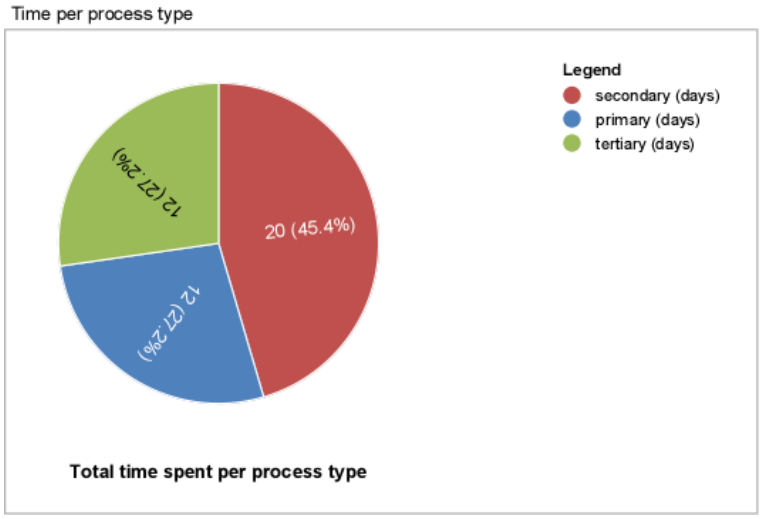


Figure 6.3 BPMN analysis "Task typing"

6.3.5 Rework

Required profiles: None.

This analysis can identify loops in the BPMN model and elements that occur in the model more than once. The results of the analysis may be an indication for necessary revisions in the model.

The result of the analysis is a pie chart showing the objects that appear most often in all possible paths. The indicated loops can be further examined in the message window.

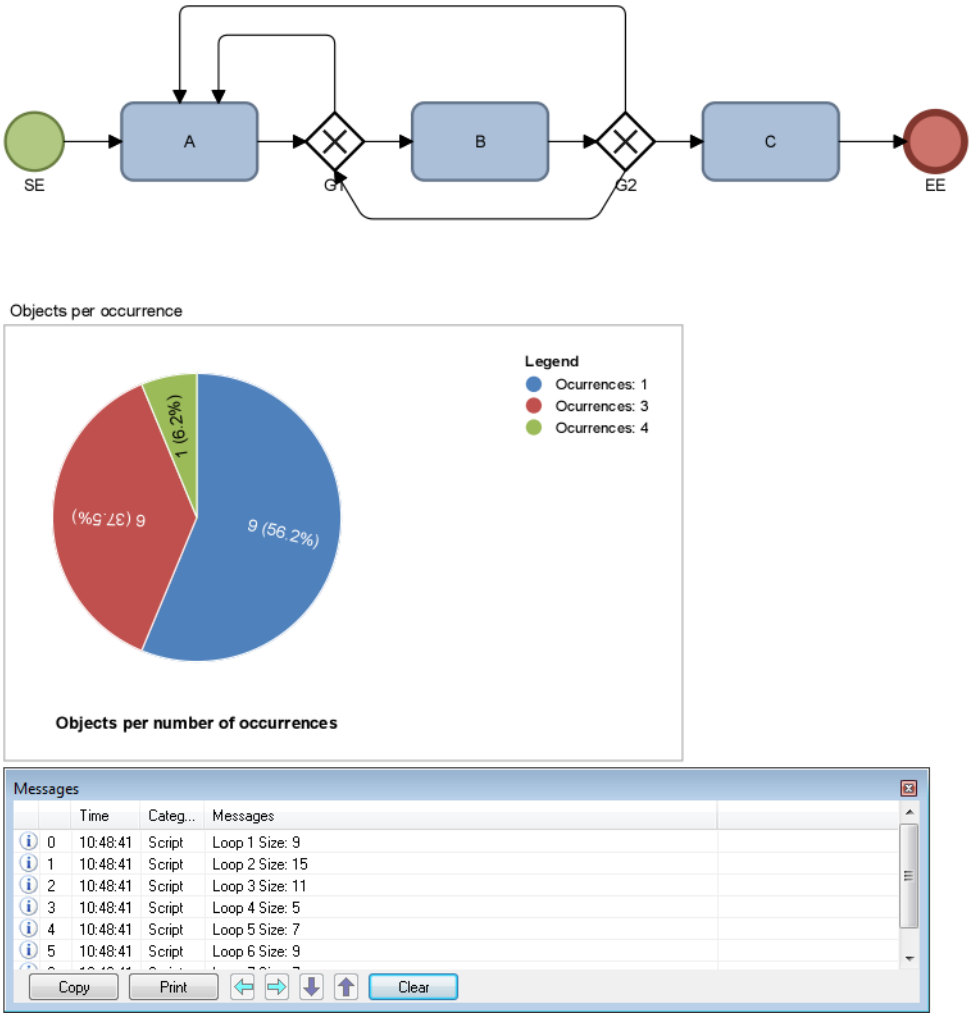


Figure 6.4 BPMN analysis "Rework"

6.4 Performing an analysis

To perform an analysis on a BPMN model, follow these steps:

1. Open the diagram containing the process you want to analyze.

2. On the **BPMN** tab, in the **Analyze** group, click the analysis you want to

perform.

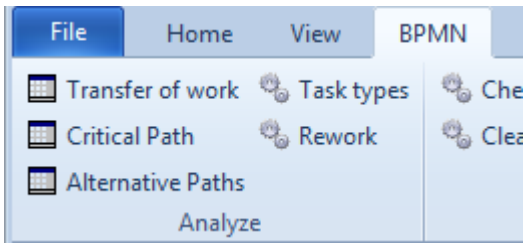


Figure 6.5 BPMN analyses

7 BPMN process check

To check whether the BPMN processes you have build are correct according to the BPMN syntax, you can perform a check on your models. This check can be found on the **BPMN** ribbon tab, in the **Check** group.

To perform the check, select the model package containing the models you want to check in the model browser. Next, on the **BPMN** tab, in the **Check** group, click **Check processes**. The check is executed and the results are presented in the message window that automatically appears. Any errors and warnings in a model are indicated as follows:

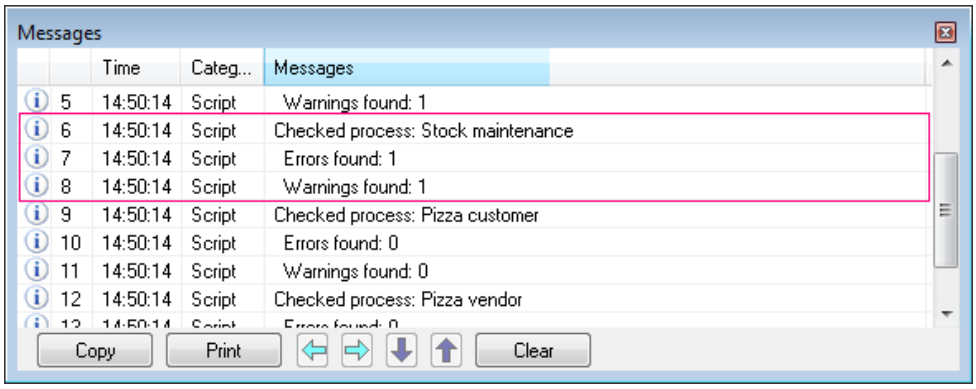


Figure 7.1 Message window with the results of a process check

When you open the appropriate model, signs mark the elements in the model to which the errors and warnings relate. Warnings are identified with a ⚠, errors with a ⛔. The errors and warnings are not further specified.

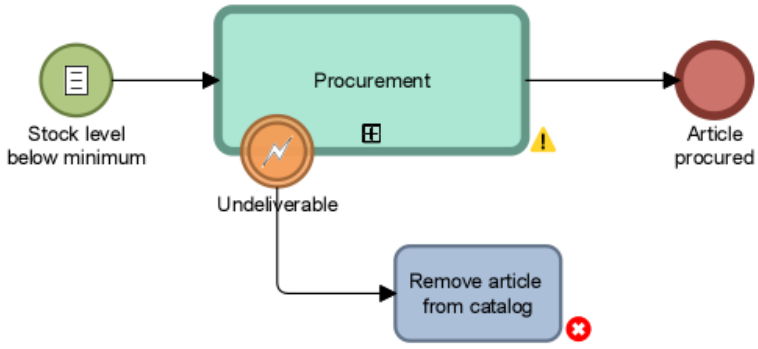


Figure 7.2 Model containing an error and a warning

To remove the error and warning signs from the model, on the **BPMN** tab, in the **Check** group, click **Clear errors and warnings**.

8 Modeling level and count functions for BPMN models

The BPMN functionality in Enterprise Studio offers functions that relate to the modeling levels present in the model, and functions for counting objects on model level and model package level.

The following functions are available:

- **Set level (model)**: Set the modeling level for all views in the model.
- **Dim higher level elements**: Dim higher level elements for all views in the model.
- **Objects (model)**: Count the objects in the model by domain and by type.
- **Objects (model package)**: Count the objects in all BPMN models within the model package by domain and by type.

To perform a function, do as follows:

- On the **BPMN** tab, in the **Modeling Level** or **Count** group, click the function of your choice.

Depending on the function you have chosen, you are requested to select a modeling level and then will see the result in the active diagram, or you will immediately see the result in the active diagram, in the Messages window, or in another results window.

9 Tips and tricks for BPMN modeling

Below you can find tips and tricks for modeling with BPMN models.

Turn a task into a sub-process

You can turn a task into a sub-process by collapsing a task in a process diagram. Right-click the task and click on **Collapse**. The task changes into a sub-process.

Turn a task into a call activity

You can turn a task into a call activity by dragging a process diagram onto a task. Select a process diagram in the model browser, drag it onto a task in another open process diagram and drop it. The task changes into a call activity.

Turn a global task into a call activity

You can turn a global task into a call activity by dragging a global task onto an open process diagram. Select a global task in the model browser, drag it onto an open process diagram and drop it. In the process diagram a call activity is created based on the dragged global task. The original global task is preserved.

Turn a process diagram into a call activity

You can create a call activity by dragging a process diagram onto another process diagram. Select a process diagram in the model browser, drag it onto another, open process diagram and release it. In the process diagram a call activity is created based on the dragged process diagram. The original process diagram is preserved.

Turn a resource role into a lane

You can create a lane by dragging a resource role onto a process diagram. Select a resource role in the model browser, drag it onto an open process diagram and drop it. In the process diagram a lane is created based on the dragged resource role. The original resource role is preserved.

Turn a collaboration into a pool

You can create a pool by dragging a collaboration diagram onto another collaboration diagram. Select a collaboration in the model browser, drag it onto another open collaboration diagram and drop it. In the collaboration diagram a pool is created that is linked to that collaboration. The original collaboration is preserved. The collaboration's contents is not shown inside the pool, it remains a black box pool.

Appendix A - Concepts for BPMN modeling

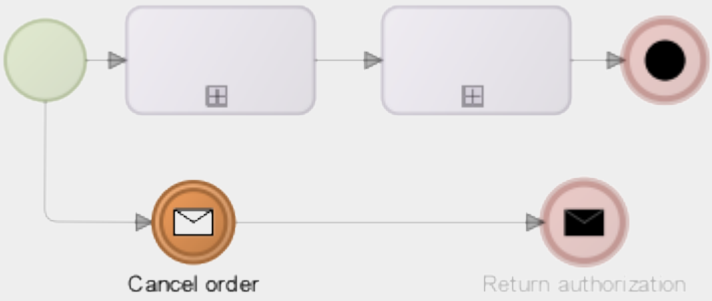
The table below shows the various concepts that can be used to create BPMN models. A number of main categories can be identified for these concepts. In some of these categories, concepts can be specified in more detail by discerning certain variations and adding specific information to them.

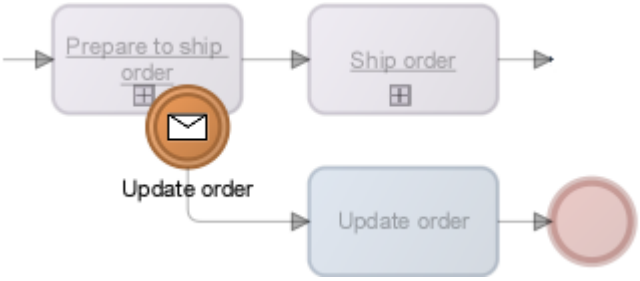
For a more detailed description of BPMN concepts and their use, please refer to the official BPMN specification, or visit the website of the Object Management Group (OMG).

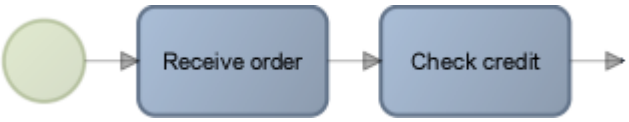
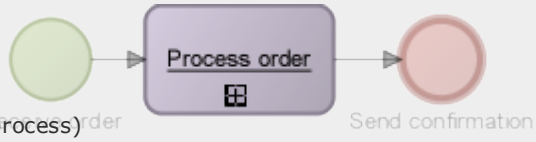
Each concept is visualized in an example, using the default color of the object. Other objects and relationships in the example are faded in order to place them in the background.

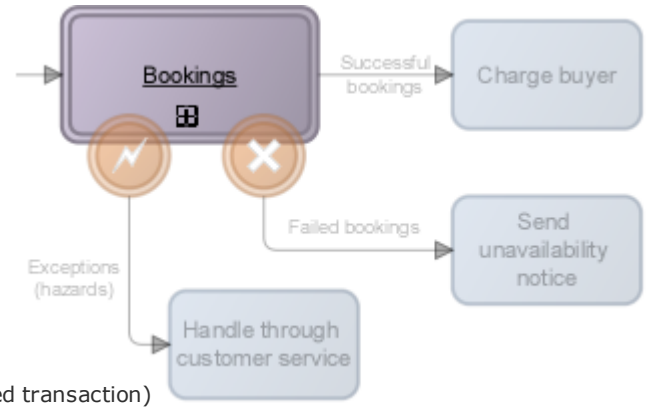
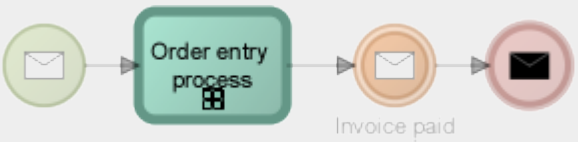
FLOW OBJECTS

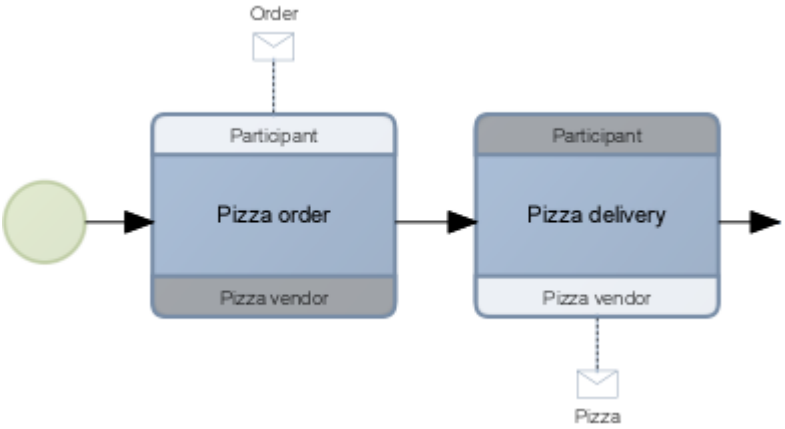
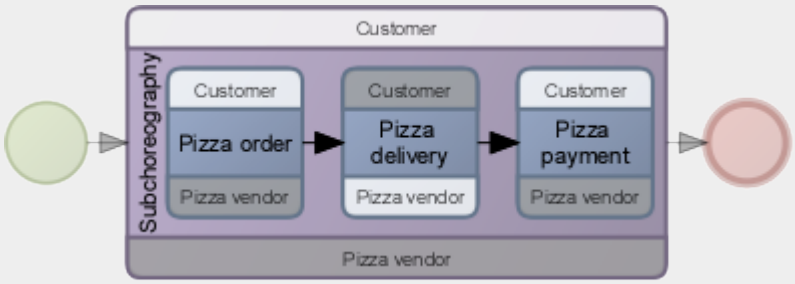
Symbol	Name + description
	**Start event** A start event represents the start of a process or choreography. A process or choreography can only have one start event. A start event can have a *trigger* which describes the cause of the event. A trigger can be modeled by adding a type to the event. 
	**Intermediate event** An intermediate event represents an event (something that happens) during the execution of a process or choreography. An intermediate event can influence the course of a process flow, and is positioned somewhere between the start event and the end event of a process or collaboration. The event is modeled in between sequence flows. The intermediate event can be activated as soon as an activity is

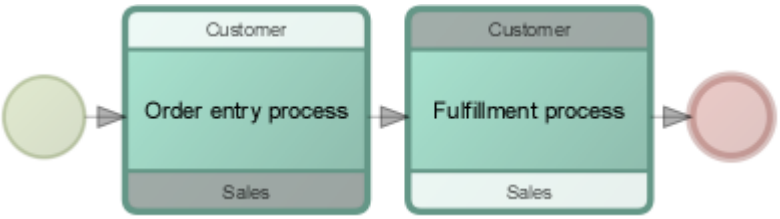
Symbol	Name + description
	completed. An intermediate event can have a *trigger* which describes the cause of the event. A trigger can be modeled by adding a type to the event. The most notable types are message, timer, and error. An intermediate event can be either *catching* or *throwing*, meaning that it can take up ("catch"), or send ("throw") a signal, respectively. Multiple intermediate events can be used within a single process or choreography. 
	**Boundary event** A boundary event is an intermediate event that is attached to the boundary of an activity or sub-process. The meaning of this is that the boundary event can already become active during execution of the activity or sub-process. The activity or sub-process does not need to be completed yet. A boundary event can only be *catching*; while the execution of the activity is underway, the event awaits a trigger signal. As soon as this signal is received by the event, new behavior will be started. A boundary event can be *interrupting* and *non-interrupting.*

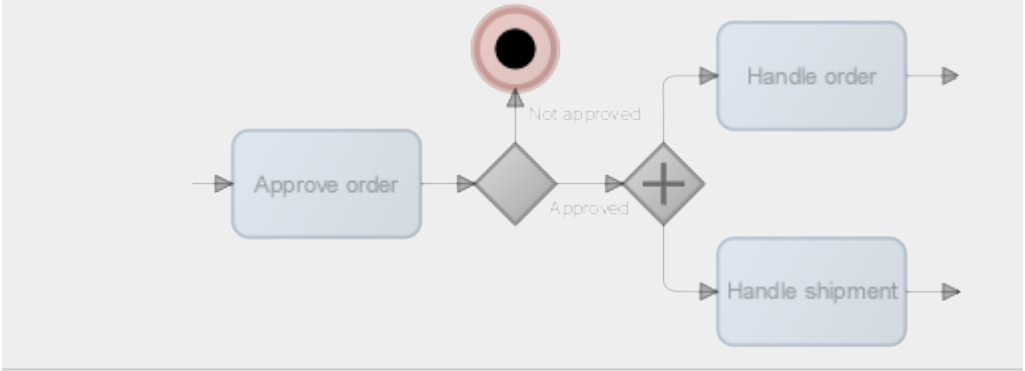
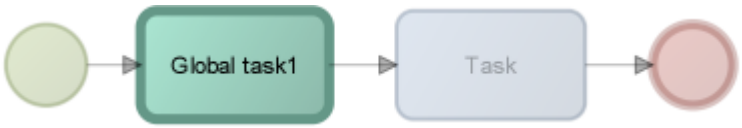
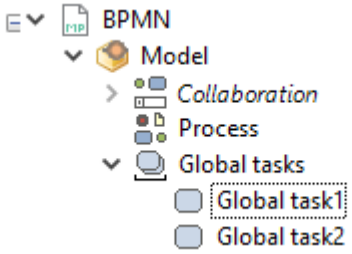
Symbol	Name + description
	
	**End event** An end event represents the point at which the process or choreography finishes. An end event can have a *result*. A result is a signal that will be passed on by the event as soon as the process flow reaches the event. The nature of the result can be defined by adding a type to the end event. An end event can only be *throwing*; it only sends out the result of the event. A process or choreography can include more than one end event, in order to model the different alternative end states as a result of the execution of a process or choreography. 
	**Task** A task is a single activity at the lowest level of abstraction in a process or collaboration. A task represents some type of activity that cannot be broken down into more detailed activities at a level below. (In cases where it is necessary to break down activities in multiple levels, a sub-process, transaction, or call activity should be used.) A task can be assigned a certain type. Assigning a type to a task gives more information about the characteristics of the task.

Symbol	Name + description
	
	Sub-process A sub-process is a composite activity within a process, which in itself is modeled using activities, gateways, events, and sequence flows. In this way, a sub-process could be considered as "a process within a process". A sub-process can be visualized showing its contents, but the contents can also be hidden by "collapsing" the sub-process. It can be accessed by clicking the ⊞ symbol in the object.  (collapsed sub-process)
	Transaction A transaction is a specific type of a sub-process. A transaction consists of a logical coherent set of activities, which can be based on a transaction protocol. The activities in a transaction are processed in a way where the execution of ALL activities is successful, or unsuccessful. The color of a transaction is similar to the color of a sub-process, but a transaction has a double edge. It is a graphical object in a process and is visualized as a single transaction, but it can also be expanded in order to model the details on a lower level of abstraction. In order to do this, click the ⊞ symbol in the object.

Symbol	Name + description
	 (collapsed transaction)
	Call activity A call activity is a sub-process that can be used in case you want to invoke a generic defined process (*global process*) that is used in multiple processes within a model package. A global process needs in principle to be modeled just once, after which it can be invoked via a call activity from multiple models within a model package. A call activity can also invoke a global task. Global tasks are activities that are not described in detail as part of the model. Global tasks are created in the model browser. A call activity is only displayed as collapsed. The process that is invoked can be opened by clicking the ⊞ symbol in the object. 
	Choreography task A choreography task is a single activity at the lowest level of abstraction in a choreography. A choreography task represents some type of activity that cannot be broken down into more detailed activities at a level below. (In cases where it is necessary to break down activities in multiple levels, sub-

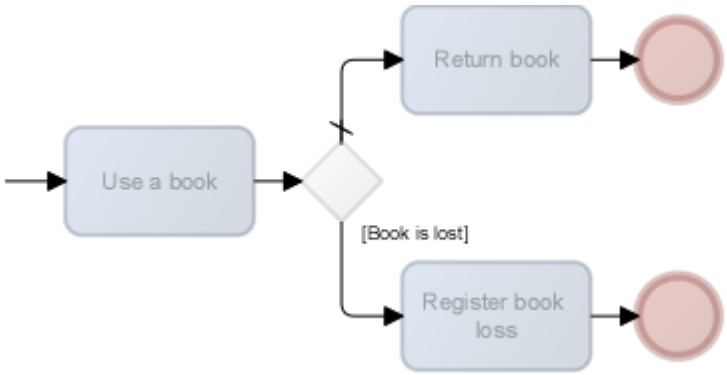
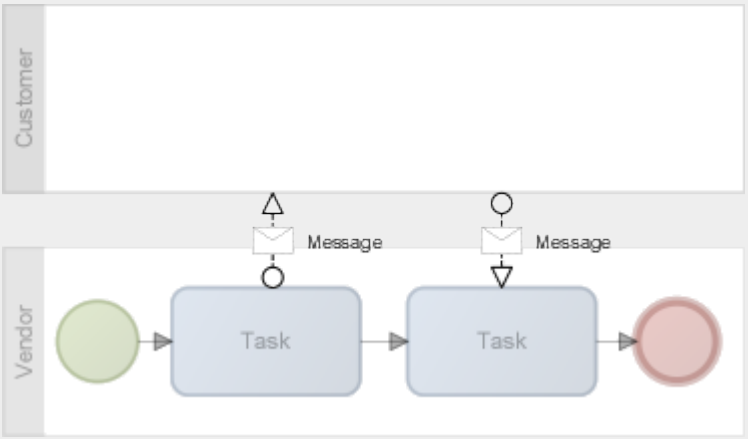
Symbol	Name + description
	choreographies or call choreographies should be used.) A choreography task represents a set of one or more exchanges of information (messages), and consists of two participants. 
	**Sub-choreography** A sub-choreography is a composite activity within a choreography, which in itself is modeled using other activities (choreography). In this way, a sub-choreography could be considered as a choreography within a choreography. A sub-choreography can be visualized showing its contents, but the contents can also be hidden by "collapsing" the sub-choreography. The details can be accessed by clicking the ⊞ symbol in the object. 

Symbol	Name + description
	
	**Call choreography** A call choreography is a sub-choreography that can be used in case you want to invoke a generic defined choreography (*global choreography*) that is used in different choreographies. A global choreography needs in principle to be modeled just once, after which it can be invoked via a call choreography from multiple models within a model package. A call choreography is only displayed as collapsed. The process that is invoked can be opened by clicking the ⊞ symbol in the object. 
	**Gateway** A gateway is used in order to control splits and joins of sequence flows in a process or collaboration. A gateway can be assigned a type which defines the behavior of the gateway.

Symbol	Name + description
	
	Global task A global task is used for defining a general activity that is not specified in more detail within the model. A global task does not have a graphical representation, and can only be invoked from a process or collaboration through the use of a call activity. Global tasks are created in the model browser.  

CONNECTING OBJECTS (RELATIONSHIPS)

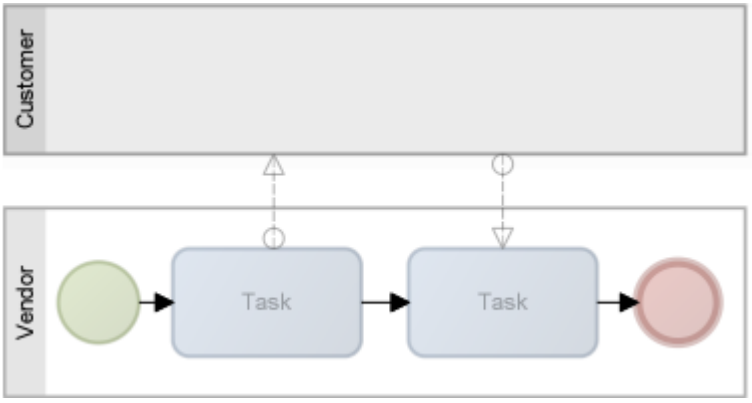
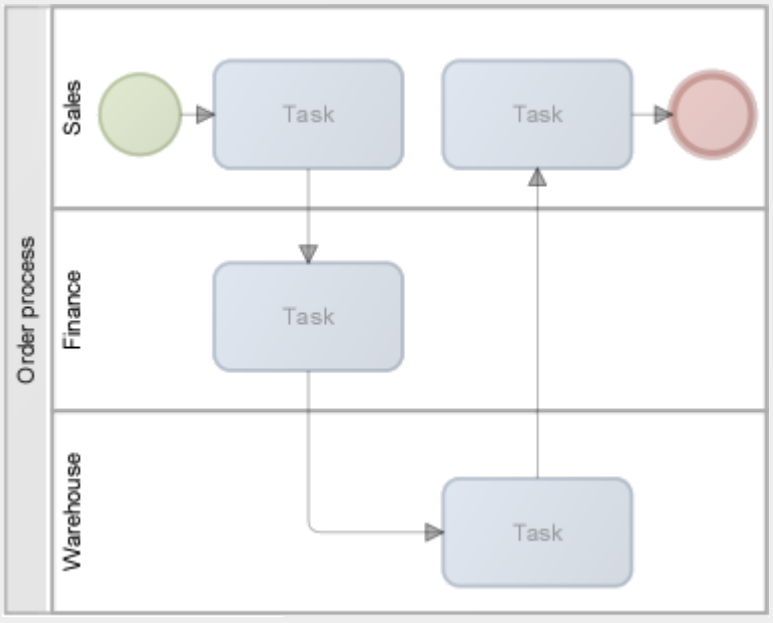
Symbol	Name + description
	Sequence flow A sequence flow is used to define the order in which the activities in a process and choreography are executed.

Symbol	Name + description
	In case a sequence flow is used in a split or join, the sequence flow can be assigned a property, making it either a *default* flow or a *conditional flow.* 
	**Message flow** A message flow represents the flow of information in terms of messages sent and received between two participants. In a collaboration diagram, participants are represented by pools. 
	**Association** In Enterprise Studio the BPMN association and data association are considered similar. Both are represented using the same icon.

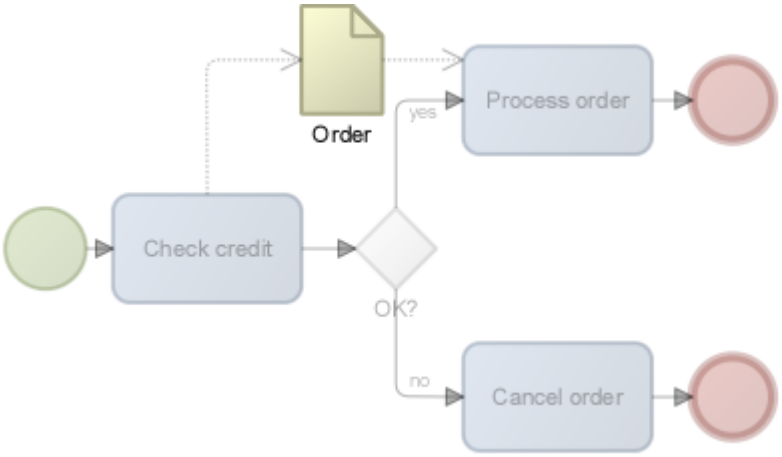
Symbol	Name + description
	An association is used to connect information and flow objects. An association is used to connect data objects and data stores to for example a task or event. Data objects can also be connected to each other by using associations. 

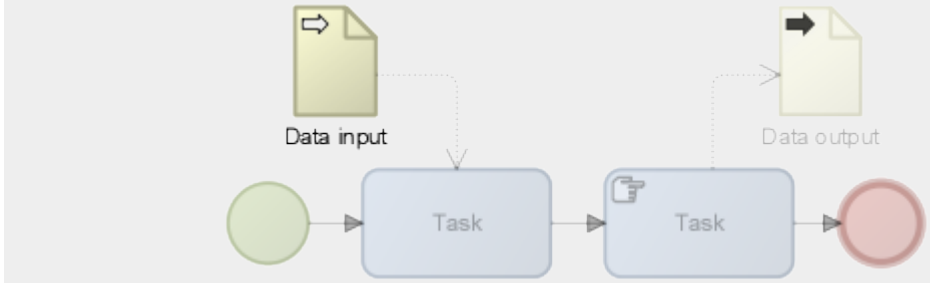
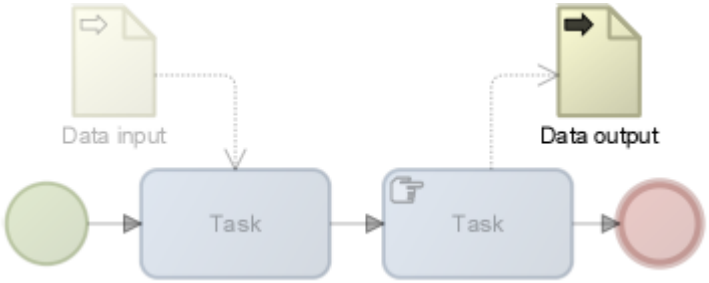
SWIMLANES

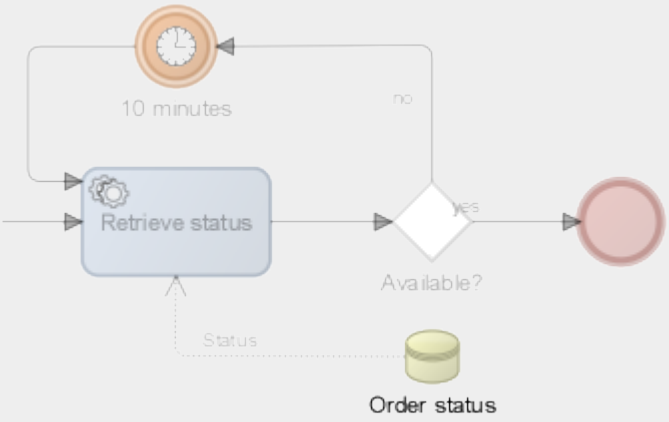
Symbol	Name + description
	**Pool** A pool represents a participant of a collaboration. At the same time, a pool acts as a lane. A pool can consist of multiple lanes, describing a collaboration with multiple participants. External participants of a collaboration are described using so called "black box" pools. This black box pool does not contain any content, except for the name of the external participant. In BPMN, external participants are not specified.

Symbol	Name + description
	
	**Lane** The role of a lane is to specify and categorize activities. A lane is a subdivision of a process or pool, and covers the full scope (length) of a process. Usually, a lane represents a role or business unit involved in executing activities that are part of the process. 

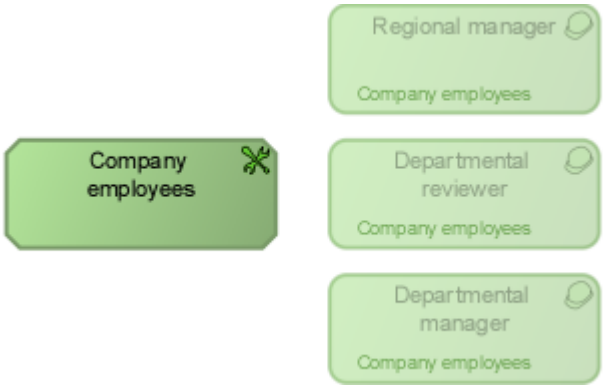
DATA

Symbol	Name + description
	**Data object** A data object describes the required inputs of an activity, as well as the resulting outputs of an activity. A data object can represent a single object, or a collection of objects. A collection of objects should be marked using the ||| control, visible at the selected data object. Data objects can be used in collaborations as well as processes. 
	**Data input** A data input is a specification of a data object and represents information needed to start an activity. It provides *input*. Similar to a data object, a data input can represent a single object, or a collection of objects. A data input cannot have any incoming data associations.

Symbol	Name + description
	
	**Data output** A data output is a specification of a data object and represents information that may be the result (*output*) from an activity. Similar to a data object, a data output can represent a single object or a collection of objects. A data output may have no outgoing data associations. 
	**Data store** A data store represents stored information that can be retrieved or updated by the process, and that persists throughout the process.

Symbol	Name + description
	

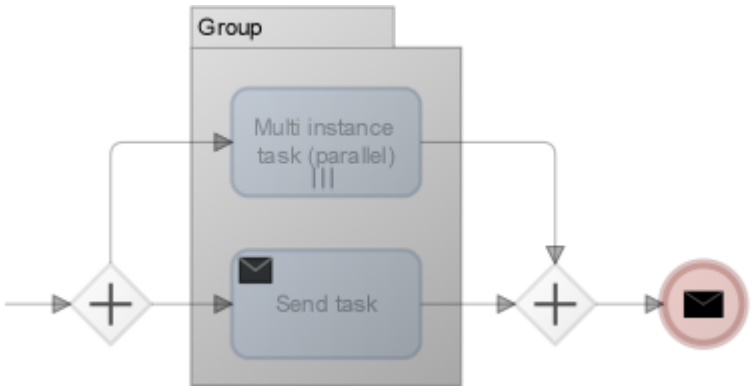
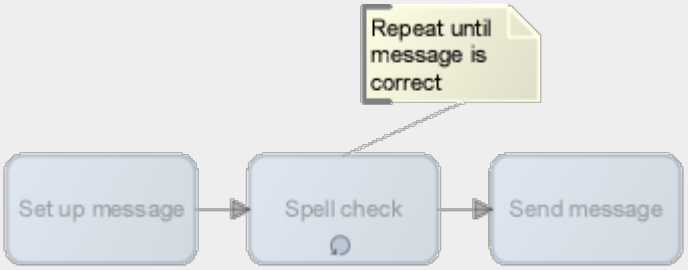
RESOURCES

Symbol	Name + description
	Resource A resource is used to define resources that perform an activity or are responsible for an activity. Activities in a process or collaboration can refer to a resource. Resources are described in a resources diagram. A resource can be linked to a resource role. This assignment is created from within the resource role. 
	Resource role A resource role is used to define the entities that can be referred to by activities in processes and collaborations. A

Symbol	Name + description
	resource role is a more detailed specification of a resource. If resource roles are described in a resources diagram, they can be assigned to activities from within a process or collaboration. A resource role can be assigned a certain type. Assigning a type to a resource role gives more information about the characteristics of the resource role. 

ARTIFACTS

Symbol	Name + description
	Group The graphical shape group represents the BPMN concept *group*. A group visually organizes objects and relationships of a diagram. A group is a graphical aid only, it does not have formal semantics; a group is not an activity nor any other flow object, and is not bound by pools or lanes. This means a group can transcend the level of a pool.

Symbol	Name + description
	
	Comment The graphical shape comment represents the BPMN concept *annotation*. A comment can be used to add textual documentation to a diagram, intended to inform the reader of the diagram. Comments can be linked to a specific object part of the diagram, using an *edge* (**Graphic shapes** group), without any consequence or influence to the process. 

Appendix B - Controls for BPMN objects

The list below shows the controls that are available for each object that can be used in a BPMN diagram. Not all controls are available on all objects. In most cases the controls are located at the top left corner of a selected object.

In addition to the controls mentioned in the list there may be object controls available that can be used in modeling, like the navigator, smart connector and arrow controls for adding new elements. These controls are generic and available in almost every modeling language and method in Enterprise Studio.

The generic controls are discussed in the Enterprise Studio User Guide (PDF).

CONTROLS

Symbol	Name + function
	**Create new process with lanes** Creates within the selected pool a new process with by default two lanes (in case of a pool with a horizontal orientation). **Create sub-lanes** Adds a new lane to an existing, selected lane. As a result, a lane set is created (in case of a lane with a horizontal orientation).
	**Create new process with lanes** Creates within the selected pool a new process with by default two lanes (in case of a pool with a vertical orientation). **Create sub-lanes** Adds a new lane to an existing, selected lane. As a result, a lane set is created (in case of a lane with a vertical orientation).
	**Insert lane above** Adds a new lane on top off the existing, selected lane (in case of a lane with a horizontal orientation).
	**Insert lane below**

Symbol	Name + function
	Adds a new lane below the existing, selected lane (in case of a lane with a horizontal orientation).
	**Insert lane left** Adds a new lane at the left hand side of the existing, selected lane (in case of a lane with a vertical orientation).
	**Insert lane right** Adds a new lane at the right hand side of the existing, selected lane (in case of a lane with a vertical orientation).
	**Hide pool/Show pool** Hides the selected pool, or shows it again after hiding it. In case of a pool consisting of multiple lanes, the pool can be hidden. **Switch from catch to throw/Switch from throw to catch** For switching between catch and throw. An intermediate event can be of type *catching* or *throwing*. Of type catching means that the process awaits the event trigger signal. Whereas throwing means that the process throws the event signal directly and continues.
	Align with lanes This makes the pool the same size as the lanes of a process, so that they form a visual unity.
	**Break alignment with lanes** Breaks the alignment between a pool and the lanes in it.
	**Change to vertical orientation** By default, lanes and participants in a diagram are positioned horizontally. Switches the lane positioning from horizontal to vertical.
	**Change to horizontal orientation** Switches the lane positioning from vertical to horizontal.
	**Toggle collection** Pool: With this you can set the minimum and maximum number of participants in a pool.

Symbol	Name + function
	Data object: Indicates that the data object represents a collection of information entities.
☆	**Create new process** Creates a new process within the selected pool. **Create called process** Creates a new process, which is invoked from the call activity it is created in. **Set initiating participant** Sets the selected participant as the initiating participant in a choreography task.
⊞	**Collapse** Collapses a sub-process or transaction. This can be done when details have been laid down within a sub-process or transaction, and you want to hide these details.
	Go to: process name Opens in a new window the process that is part of a pool.
⊗	**Unlink existing process** Removes the link between the pool and a process within that pool.
✕	**Remove** Removes the lane and its contents from a pool. **Remove called element** Removes the reference to a called process or global task.
	Show resource roles/Hide resource roles Creates a link between the resource role and the selected lane, task, sub-process, transaction or call activity. Upon clicking the control, a pop-up window will appear from which you can select the resources that need to be linked to the object by clicking the "plus" icon and then selecting the desired resource. Moreover, existing links to resource roles can be removed using

Symbol	Name + function
	this control.
	Toggle compensation Turns a task into a compensation task. A compensation task is an activity that is used as an alternative in case of failure during execution of another (normal) activity. It specifies that this is the alternative activity that is being performed.
	**Set existing process** Links an existing process to the selected pool. The process appears in the pool after selecting. **Set called choreography** Links the choreography that should be called. The name of the called choreography appears after selecting. **Set called element** Creates a link between a call activity and the process or global task that must be called. The name of the process or global task appears in the call activity after selecting the object. For the above objects, upon clicking the control a pop-up window will appear from which you can select the desired object. **Toggle direction** Indicates the direction of the association between a data store or data object (incl. input and output) and an activity or event, or turns of the direction. Each time the control is clicked, the direction changes or disappears (following a loop). You can choose from no direction (default), one direction left or right, or both directions. **Make anonymous/Undo anonymous** Hides the name of the selected lane set, or shows it again.
	**set <object type>** This control lets you indicate a specific type of the object. Upon clicking the control, the various types that are available for an object will be shown. This may vary by object. Click on a type to

Symbol	Name + function
	assign to an object. Whether it is possible to assign only one or multiple types, depends on the object. The assigned object type will be indicated with an icon within the object. **Set flow characteristic** This control lets you configure the flow characteristic. Upon clicking the control, the various characteristics that are available for a flow will be shown. Select a characteristic to assign it. The assigned characteristic is visualized by the display of an icon at the starting point of the flow.  indicates a standard flow,  indicates a conditional flow that is going out from an activity, which has a second outgoing conditional flow.
	**Toggle visibility** Links a message flow item to the selected message flow, and visualizes this. The way this is visualized can be configured by clicking the envelope icon multiple times (following a loop). The envelope icon is by default white. Clicking the white envelope turns it gray. Yet another click hides the envelope. Please note that this action does not unlink the message flow item from the message flow. To achieve that, you should click the cross sign at the top right corner of the envelope icon. This cross sign is only displayed when a linked message flow item exists.
	Add to choreography task Links the selected message flow to a choreography task. The choreography task can be selected from the pop-up window that appears after clicking the control.
	**Set as triggered by event** Turns a sub-process into an event driven sub-process. This means that the sub-process has its own trigger and that it is not

Symbol	Name + function
	part of the normal flow.
	Toggle ad hoc Turns a sub-process into an ad hoc sub-process. An ad hoc sub-process comprises a number of embedded inner activities and is executed in a more flexible order than normal processes. An ad hoc sub-process is not a normal process that runs from a start event to an end event. It consists only of activities, sequence flows, gateways and intermediate events, and also data objects and data stores.
	Toggle loop Turns a (choreography) task, sub-process or sub-choreography into a repeated activity. By clicking this control repeatedly, different types of loops can be configured, or the loop characteristic can be switched of: Standard loop Multi-instance - parallel Multi-instance - sequential
	Toggle message decorators Displays with a choreography task the linked message flow from a collaboration, or hides this again. This control is only available for a choreography task if a link is created in the collaboration to the choreography task.
	Create reference Creates a reference data object of the selected data object. Using a reference data object allows you to use multiple instances of a data object in a diagram. A reference data object is marked with the following icon in the object: . Upon clicking this icon, the data object that is referenced to will be highlighted in the diagram.
	Set as non-interrupting

Symbol	Name + function
	If a boundary event is non-interrupting, the activity that this boundary event is linked to will not be interrupted upon the receipt of a signal by the event during the execution of the activity. As a result, the alternative (*exception*) flow will be activated after the execution of the activity is finished. The normal flow will be activated as well. So both flows will run in parallel.
○	**Set as interrupting** If a boundary event is interrupting, the activity that this boundary event is linked to will be interrupted upon the receipt of a signal by the event during the execution of the activity. As a result, the alternative (*exception*) flow will be activated after the execution of the activity is finished. The normal flow will not be activated, so either the exception or the normal flow will run.
➡	**Show link menu/Hide link menu** If a boundary event of type error is attached to the boundary of a sub-process, or call activity linked to a sub-process containing one or more error end events, the boundary event can be linked to an end event from this sub-process in order to establish an *error throw-catch*. The control is also available in boundary escalation events in the above situation. In this situation, the boundary event can be connected to an escalation end event or intermediate event in order to establish an *escalation throw-catch* between the sub-process and the boundary event.

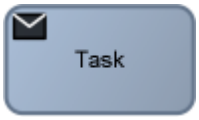
Appendix C - BPMN object types and markers

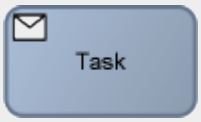
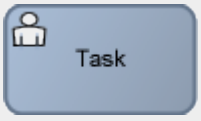
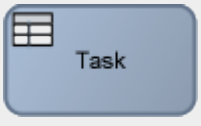
Various BPMN objects can be characterized in order to add additional information in a model. The available characterizations are listed in the tables below. The characterizations are marked using an icon that appears in the object upon the assignment of a characterization.

> The list below is not complete. For a more detailed explanation and application of the BPMN objects, their characterizations and properties, please refer to the official BPMN specification, or visit the website of the Object Management Group (OMG).

Task characterizations are used to specify what type of action should be executed.

TASK TYPES

Symbol	Type + description
	**Send task** A simple task for sending a message to an external participant (from the perspective of the process). Upon sending the message, the task is completed. 
	**Receive task** A simple task for waiting on a message from an external participant (from the perspective of the process). Upon receiving the message, the task is completed.

Symbol	Type + description
	
	**Manual task** A manual task that is executed without the support of an application or machine that executes the business process. 
	**User task** A typical workflow task that is executed by a person supported by a software application. 
	**Service task** A task that uses some sort of service, for example a web service or an automated application. The task is (usually) executed fully automated. 
	**Business rule task** A mechanism for the process to deliver input for a "business rule engine" (software system) and receiving the output of calculations from a business rule engine. 

Symbol	Type + description
	Script task A task that is executed out by a "business process engine" (BPE). For this task a script needs to be defined that can be interpreted by a business process engine. As soon as the task is ready for execution, the business process engine will execute the script. Upon completion of the script, the task will be completed as well. The task is executed fully automated. 

Besides the **event types** listed below, events can also be specified as (non-)interrupting, catching, throwing or as a boundary event. Whether or not a characterization is available for an event, depends on the type of event (start, intermediate, boundary, end), and its characterization.

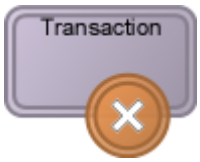
EVENT TYPES

Symbol	Type + description
none	**Untyped** Events without a characterization such as a starting point or change of status. Does not have a trigger or result specified. 
	Message event Messages are being sent or received. For a start event this means that the process is triggered by an external signal: a message. In case of an intermediate event, the event sends or receives a message as a signal between the process and an external entity.

Symbol	Type + description
	In case of a boundary event, the event is triggered by a message that can be received. In case of an end event, the event sends a message to an external participant. 
	**Timer event** In case of a start event, this characterization is used to show that the process is executed based on a predefined time schedule, one time only, or recurring. For an intermediate event, this characterization can be used to configure a certain delay. This means that a delay of a specific duration is applied, or that the process waits until a specific time or date before it continues to execute. In case of a boundary timer event, the event behaves similar to a combination of a stopwatch and an alarm clock. The event is triggered as soon as the process reaches the activity that is linked to the event. Whatever happens next depends on the event being interruption or non-interrupting. 
	**Conditional event** This type of event reacts to certain conditions. The event is triggered as soon as some condition is satisfied. In case of a start event, it indicates the start of a process when a business condition becomes true. In case of an intermediate event, it is used as the flow needs to wait until a business condition is fulfilled. It can be used in the sequential flow to indicate that it is necessary to wait until an business condition is fulfilled.

Symbol	Type + description
	In case of a boundary event, it indicates an exception flow, which will be activated when the condition is fulfilled. 
△	**Signal event** Is used to provide signaling between processes. For a start event this means that the process is triggered by an external signal, for example the signal that a new customer was created in the ERP system. In case of an intermediate or boundary event, the event awaits a signal. For an end event, a signal is sent that is received either externally, or by a linked catching signal within the process model. 
⇨	**Link event** This is a mechanism for linking two parts of a process. Can only be used for intermediate events. Link events can be used to create loops in the process, or to avoid long sequence flows. 
	Error event This type of event is used for error handling. A start error event catches defined errors (*catching*). A boundary error event catches and processes defined errors. For an intermediate event this characterization can only be used as an interrupting boundary event.

Symbol	Type + description
	In case of an end error event it indicates that a defined error is generated (*throwing*) when the process is ended. 
	**Escalation event** This is a variation of the error event. In case of a start event, the event reacts to escalation to another role in the organization. This event type is only used in an event sub-process. For an intermediate event, the catch and throw behavior is equal to the behavior of an error event, but with the following differences: • The boundary event is non-interrupting by default. • Escalation does not mean an error, but merely some additional processing that is triggered during a process activity. In case of an end event, the catch and throw behavior is equal to that of an error event, the only difference being that in case of escalation the process or sub-process will not be terminated if one or more parallel paths are still active. 
	**Cancel event** Indicates that a transaction should be canceled immediately, and provides a response (trigger). For an intermediate event, the cancel event is only used for boundary events attached to a transaction.

Symbol	Type + description
	In case of an end event, the cancel event is a variant of an error event. The cancel event is only used within transactional sub-processes, and is always interrupting. 
	**Compensation event** For an end event this indicates that the process was finished and that compensation is required. For an intermediate event in the subsequent flow of a process, this indicates that compensation is required (*throwing*). In case of use on the boundary of a task, it indicates that this task will be compensated upon activation of the event (*catching*). 
	**Terminate event** Can only be used for an end event. Initiates the immediate termination of all activities in a process. 
none	**Multiple event** Can be used for all event types. For a start event this characterization indicates that multiple events can potentially start the process. Occurrence of just one of these events is required to actually trigger the start of the process. For an intermediate and boundary event, this characterization means that multiple triggers are assigned to the event.

Symbol	Type + description
	In case of an end event, this characterization indicates that many results are possible after completion of the process. All of these results need to be achieved. 
none	**Parallel multiple event** Only start, intermediate and boundary events can be characterized as parallel multiple event. For a start event this characterization indicates that multiple events can potentially start the process. Occurrence of all of these event is required to actually trigger the start of the process. In case of a catching intermediate event and a boundary event, this characterization indicates that multiple triggers are assigned to the event. 

GATEWAY TYPES

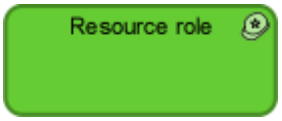
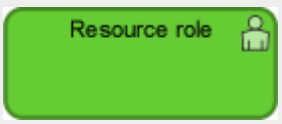
Symbol	Type + description
none or ✕	**Exclusive gateway** In case of a split, only one of the outgoing flows is to be followed, based on some condition. A split of this kind is also called an *XOR-split*. An exclusive gateway can be displayed in two ways, with and without an X in it. In case of a join, the flow continues after one of the flows has entered the gateway. A join of this kind is also called an *XOR-join*.

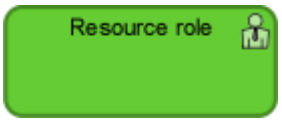
Symbol	Type + description
	
○	**Inclusive gateway** In case of a split, one or more of the exiting flows will be selected for continuation of the process, depending on the conditions that are defined for the gateway. A split of this kind is also called an *OR-split*. In case of a join, the corresponding incoming flow of all selected flows must have entered the gateway before the process is allowed to continue. A join of this kind is also called an *OR-join*. 
+	**Parallel gateway** In case of a split, all outgoing flows are to be followed in parallel. A split of this kind is also called an *AND-split*. In case of a join, all entry flows need to have entered the gateway before the process is allowed to continue. A join of this kind is also called an *AND-join*. 
◎	**Event-based gateway** Represents a point in the process where the flow splits up. All the available alternatives are based on the events that occur at that point in the process. This is always followed up by incoming events or catching tasks. 
*	**Complex gateway**

Symbol	Type + description
	This is a complex split or join that cannot always be defined by gateways. It is used for all types of splits and joins that cannot be represented by the default types of gateways. 

A **resource role** is used to define resources that perform an activity or are responsible for an activity. With the available types resource roles can be characterized in more detail from general to more specific.

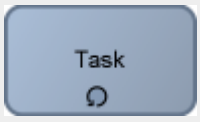
RESOURCE ROLE TYPES

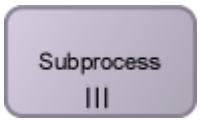
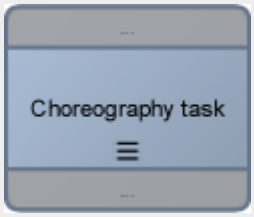
Symbol	Type + description
	**Performer** A resource role of type performer is the most general characterization. The performer is the resource that performs the activity or is responsible for it. The performer can be specified in the form of a specific individual, a group, an organization role or position, or an organization. 
	**Human performer** With the human performer type you explicitly specify that the resource is a person. 
	**Potential owner** A potential owner is a specialization of human performer. A resource role of type potential owner represents the person who is responsible for an activity and can decide to start the activity.

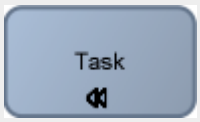
Symbol	Type + description
	

Markers show the output behavior of a task. The markers listed below can be present for a task. Some of these markers are documented using a control that is available for the task.

MARKERS

Symbol	Name + description
	Open This icon appears in a sub-process or transaction, after adding details within the object, and subsequently clicking the Collapse control in order to hide the details. Upon clicking the plus icon, the details will be shown in a separate diagram. **Jump to called element** For a call activity, this icon appears after linking the call activity with a process using the Set called element control. Upon clicking the plus icon, the called process will be shown in a separate diagram. From a call activity, it is also possible to call a global task, but no plus icon will appear in that case. 
	Loop Indicates that the (choreography) task, sub-process or sub-choreography is a repeated activity, of type *standard loop*. This is configured using the Toggle loop control. 
	Multi-instance - parallel

Symbol	Name + description
	Indicates that the (choreography) task, sub-process or sub-choreography is a repeated activity, of type *multi-instance - parallel*. This is configured using the Toggle loop control. 
☰	**Multi-instance - sequential** Indicates that the (choreography) task, sub-process or sub-choreography is a repeated activity of type *multi-instance - sequential*. This is configured using the Toggle loop control. 
~	**Ad hoc** Indicates that a sub-process is ad hoc. This is configured using the Toggle ad hoc control. An ad hoc sub-process comprises a number of embedded inner activities and is executed in a more flexible order than normal processes. 
⏪	**Compensation** Indicates that a task is a compensation task. This is configured using the Toggle compensation control. A compensation task is an activity that is used as an alternative in case of failure during execution of another (normal) activity.

Symbol	Name + description
	

Index

N

O

P

R

S

PDF guides

The following PDF guides are available for download from the BiZZdesign community:

HoriZZon

HoriZZon Web Portal Guide

Enterprise Studio

Advanced Modeling Guide
Amber Getting Started Guide
ArchiMate Getting Started Guide
Analysis Guide
BPMN Getting Started Guide
BiZZdesign Connect Guide
DMN Getting Started Guide
Enterprise Analytics Guide
Enterprise Portfolio Management Guide
Enterprise Studio Options Guide
Enterprise Studio User Guide
ERD Getting Started Guide
ERSM Getting Started Guide
License Management Guide
Metamodeler Guide
Reporting and Printing Guide
Scripting Reference
TDM Getting Started Guide
Team Platform Guide
Time Modeling and Analysis Guide

Contact BiZZdesign

Service desk	For questions and information regarding service and support. Phone: +31 - 53 - 4 878 171 E-mail: servicedesk@bizzdesign.com
Academy	For questions and information regarding training and education. Phone: +31 - 33 - 7 600 284 E-mail: academy@bizzdesign.com
Inside sales	For questions and information regarding commercial conditions, new modules, and prices for use. E-mail: sales@bizzdesign.com
Head office	For general questions and information. Phone: +31 - 53 - 4 878 151 E-mail: info@bizzdesign.com
Website	http://www.bizzdesign.com
Customer portal	http://community.bizzdesign.com