Modeling Guidelines Manual

[Insert company name here]
July 2014
Author: John Doe
john.doe@johnydoe.com

Table of Contents

1. Introduction

Signavio created the Modeling Guidelines Handbook with the purpose of providing business process management (BPM) professionals with a practical and convenient, yet detailed reference guide for designing optimal business processes. This handbook is best used as a guide for companies seeking to better understand BPM and the industry standard modeling language, BPMN 2.0.

Please note that content in the Modeling Guidelines Handbook is suggested material that is intended to help organizations achieve better results with BPM. It can of course be modified and tailored to your company's specific needs.

The five chapters in the handbook are easy to follow, even if you have little-to-no experience with BPM. After reading through this introduction, you will go on to the second chapter which discusses the themes of Business Process Management by answering the following questions: What is BPM? What is a process? What are the goals of BPM? Is there a general approach to process modeling? What are the necessary process elements? The third chapter explains the core elements of Business Process Model & Notation (BPMN). These elements are activities, gateways, events, data objects, artifacts, roles, and connectors. The general organization of models and the five possible levels for process architecture will be defined in the fourth chapter. The final chapter will describe the modeling rules, which are divided into the following five categories: architecture, notation, naming, process structure, and layout.

Please feel free to learn more about the goals and the description of the rules on the website: www.modeling-guidelines.org.

2. Business Process Management (BPM)

Business Process Management is used to help visualize organizational processes that better align IT with business needs. BPM most notably leads to benefits that help improve operational efficiency by increasing return on investment (ROI) for many projects as well as lowering total costs of ownership (TCO). BPM can also be used to obtain certification(s) in order to be compliant with standards such as ISO 9001.

Once the processes are modeled and simulated, a stakeholder can identify costs for process execution, which provide insight to optimizing process flow. Altogether, BPM is everything from analyzing to designing to executing business processes. The following questions will help frame the theories behind the practice.

2.1. What is Business Process Management?

Business Process Management is a systematic approach that combines people and technology to improve organizational performance and enhance customer satisfaction. Processes can be captured, documented, executed, measured, optimized, and controlled to help the company achieve strategic business goals efficiently and effectively.

2.2. What is a process?

A process is a sequence of defined operations, tasks, and activities that can be executed by humans and/or machines. Processes are set in place to help organizations reach multiple goals that add value to the customer and benefit the company.

In viewing a process, one can see that different participants are represented by Pools and Lanes. The activities, which are completed by the process participants are called tasks and produce a specific service or product for a company.

2.3. What are the goals of Business Process Management?

BPM has many different goals, but ultimately they all seek to establish optimal business processes and promote effective communication that leads to improved operational performance:

- ✓ To integrate technology and documents in your workflow to optimize the productivity
- ✓ To create a high level of transparency for the quality management, certification and accreditation departments.
- ✓ The control the existing and potential costs via process simulation
- ✓ To plan the optimal usage of available resources
- ✓ To improve allocation of resources and lower costs with process simulation and optimization

2.4. Is there a general approach to process modeling?

We recommend that you establish a method for designing process models.

The steps described here can be used as a general means to draft your processes.

Step 1: Defining the "happy path" – the ideal status of a process. This is a linear model that has only one pathway and no limitations or splits.

Step 2: Describing and modeling the "As-is" (current) business processes. These processes can have forks since there is rarely a happy path for any process

Step 3: Describing and modeling the "To-be" (future) processes – the optimized state in which a process should be after having eliminated unnecessary steps

Step 4: Implementation of Step 3 – This step includes the technical implementation, the correct execution, and the control of the processes.

2.5. What are the necessary process elements?

The elements used in process diagrams and visuals create easily understandable and transparent operational workflows. Modelers and collaborators can ask the following when designing and optimizing their processes:

1) What process are we designing? (e.g. company procedure)
2) Who is designing? (e.g. process owners)
3) Who will be using the process? (e.g. management & process participants)
4) Which methods? (e.g. the necessary tools required to execute the process)
5) Which parameters to measure? (e.g. costs per activity)
6) What are possible error frequencies? (e.g. rate of complication)

3. BPMN-Core Elements

BPMN core elements are the most widely used objects within the business process modeling notation. This set of elements consists of activities, gateways, events, data objects, artifacts, roles, and connectors. The following descriptions will help you quickly understand the uses for each of the core elements.

3.1. Activities

Activities describe the different steps in processes. They should be labeled by the action describing the task (e.g. Cook a soup). There are two different types of activities in BPMN. The first type is a simple activity, which is a task in the process. The second type is a collapsed subprocess, where more than one activity is modeled in a separate process and linked to a task in the parent process.

3.2. Gateways

Gateways are used in processes when alternative paths need to be modeled. They split the procedure in two (or more) ways, and merge the paths later in the process. There are five different kinds of gateways in BPMN:

- Exclusive gateways (XOR-Gateways) are used if only one condition can be chosen. The XOR gateway awaits one incoming branch before triggering the outgoing flow.
- Parallel gateways (OR-Gateways) activate all outgoing flows, and the junction awaits all incoming flows before the process flow can continue.
- Inclusive gateways (OR-Gateways) activate one or more outgoing flows, and the junction awaits all active incoming flows.
- Event-based gateways are those that are followed by catching events or received tasks. Sequence flow is routed to the subsequent event/task which happens first.
- Complex gateways are used if no other gateway can be used. This gateway requires a text annotation to define its behavior.

3.3. Events

There are three types of events, and they are categorized by how they affect process flow: start, intermediate, and end. Start Events are those that can begin process flow. Intermediate Events are those that represent catching or throwing intermediaries. Catching Intermediate Events are used if there is a trigger to continue the process (e.g. receive an email), and Throwing Intermediate Events are modeled when there is something to do for continuing the process (e.g. send an email). The End Events are the ones that close the running process.

3.4. Data Objects and Artifacts

Data objects can be attached to and used by activities. They represent information (e.g. IT-systems, paper documents, or electronic documents).

Artifacts are additional ways to represent information (e.g. annotations or arrangements). These can be directly linked and associated with other elements.

3.5. Roles

Roles characterize participants, and are depicted in models as pools and lanes which encompass the content of process diagrams. A pool can be used to define an organization (e.g. Google). Lanes can then further define the departmental units within pools (e.g. Google Marketing or Google Sales).

3.6. Connectors

Connectors are the physical links between the BPMN elements. They are used to represent a sequence in which the different activities are to be executed.

There are three different flows: Sequence flows, message flows, and associations.

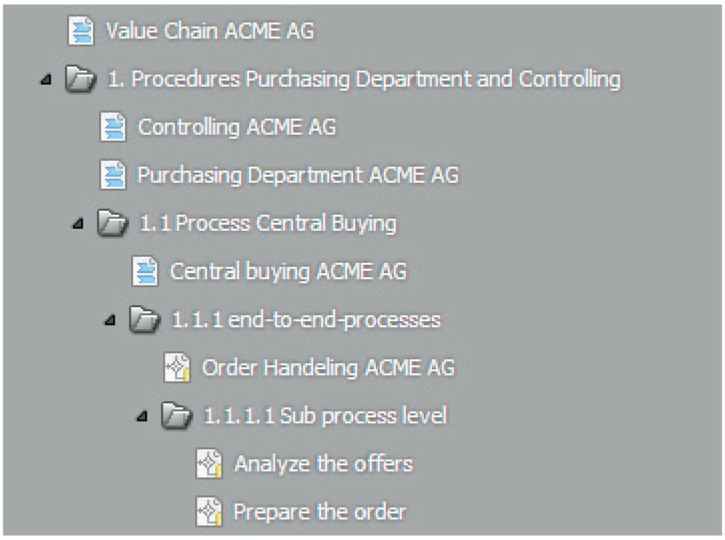
- Sequence flows are modeled between activities, gateways, and events.
- Message flows connect two different organizational units or pools, and represent the message exchange between them.
- Associations are used to connect activities with different data objects.

4. Organization of the Models

It is important to define a structural approach for models. The overall process architecture is best represented in by organizing models into hierarchies that contain specific folders. We describe diving the folder hierarchy into five different levels, from zero to four, which we explain in detail after the brief descriptions below.

Level 0 should have a value chain to define the global representation of the company. Level 1 should define the departments and interdepartmental processes. Level 2 should represent the different departments and units and level 3 should be the end-to-end-processes (modeled as BPMN diagram). In level 4 should be modeled the subprocesses, modeled as BPMN diagram as well.

If a process architecture is designed in a folder structure, it can be easily displayed like the following image:



Value Chain ACME AG
1. Procedures Purchasing Department and Controlling
Controlling ACME AG
Purchasing Department ACME AG
1.1 Process Central Buying
Central buying ACME AG
1.1.1 end-to-end-processes
Order Handeling ACME AG
1.1.1.1 Sub process level
Analyze the offers
Prepare the order

The approach to this section will be specific to each company, and should be modified by an employee of the quality department. The following representation is an example method to organize different models, and should only be used as a reference for your company. Please review the level definitions and example below to begin defining your process architecture hierarchy.

4.1. Level 0: Value Chain

We recommend modeling level 0 as a value chain. A value chain is used to depict the structure of management processes, core processes, and support processes. This visualization is most widely used for management, division leads, and employees to navigate the corporate process structure.

4.2. Level 1: Departments and Interdepartmental Processes

Level 1 shows allows process participants to explore different departments and the interdepartmental processes.

4.3. Level 2: Department / Unit Processes

Level 2 represents the different departments or units. These different departments are also called unit processes.

4.4. Level 3 (BPMN): End-to-end Processes

In level 3, the processes are modeled in a BPMN diagram. The third level represents the end-to-end-processes only.

4.5. Level 4 (BPMN): Optional Subprocesses

Level 4 represents the subprocesses of level 3. This is an optional level.

5. Description of the Rules

The rules below will help you outline high quality, standard-compliant, and consistent process models. You can modify them or use them as is to form a basis for approaching business process modeling in your organization. This will ensure that everyone has a mutual understanding for the flow of the collection of models by referencing the same material. You will be happy to know that once these are defined in your workspace, the Signavio Process Editor automatically checks your models against these rules, and warns if any are violated or forgotten.

The rules we describe are separated into five categories. The first defines the rules for the architecture of process models. The modeling convention for process architecture defines how to incorporate open comments (created by other users), and also elements like activity descriptions and dictionary links. In the second category, we highlight the rules for notating a process. Process notation involves using glossary terms different elements or the description of activities. The third rule describes different conventions and styles for naming and labeling processes. It is much easier to follow a process structure that implements consistent naming of processes and the elements linked to them. In the fourth category, we describe the rules for the process structure, which define the semantic and syntactic conventions for process models. This category also defines the rules for checking potential mistakes like deadlocks or other syntactic elements in the model. The last category defines the rules for the process layout. This set of rules outline how to: properly distance the elements; maintain an appropriate diagram size; correctly define the edge direction of a model (e.g. from left to right).

5.1. Convention for the architecture

Usage of unique diagram names

Goal

The diagrams should get unique names to develop a structure in the modeled diagram. The names of the individual diagrams should not be duplicated.

Description

The name of a diagram should be unique and the name should reflect the purpose of the process. The name can also be a pattern of numbers. For a naming example, if an employee summarized an order, (including “Write a bill” and “Refill the warehouse stock”), the name of the process could be “Handling of Orders.” If there are a high number of models, then the use of numbering in each model will be useful. One possibility for a diagram name can be “1.4 Handling of Orders,” and the sub-diagram name can be “1.4.1 Invoice of an Order.”

Incorporation of open comments

Goal

Open comments can contain useful information for the process. Incorporating open comments is necessary to ensure that every comment is accepted from the process modeler.

Description

Comments could be useful information for the process. It is important to consider open comments. If the comments are relevant to the diagram, then they should certainly be factored into the process.

In the comment function, there are three different options for the marking of a comment: “Factored In”, “Ignore”, and “Delete”. The arrow on the left side turns green at the selection of “Factored In”. It turns gray at the selection of “Ignore”, and the comment will be deleted at a selection of “Delete”. A comment should be “Factored In”, when it is relevant for the process. It should be “Ignored” or “Deleted”, when it has no meaning for the process. The changes must be saved after one of the selections are executed.

5.2. Convention for the notation

Usage of activity descriptions

Goal

An activity can be interpreted in several ways. To make sure that the activity is understood properly, a document in which the activity is described should be installed.

Description

The descriptions of activities take place in the Signavio Process Editor on the right hand side under the link ‘documentation’. This is where the activity should be described and documented in more detail.

Definition of required dictionary links of roles

Goal

A role should be available in the dictionary. With this, information about the role can be centrally maintained. It can also be derived, at which point the role is used in processes.

Description

In the dictionary, terms can be centrally documented and renamed. This makes sure that all usage of dictionary terms can be centrally adapted. Ehen naming a role, an already existing term can be picked if this role and its name are already in the dictionary. Should the required role not be in the dictionary, then a new term can be put in the dictionary via the dictionary icon 📘.

Definition of required dictionary links on data objects

Goal

A data object should be deposited in the dictionary. With this, information about the object can be centrally maintained. It can also be derived, at which point the object is used in processes.

Description

In the dictionary, terms can be centrally documented. Templates and forms can be centrally installed. When naming a data object, an already existing term can be picked if this data object and its name was already in the dictionary. Should the required data object not be in the dictionary, then a new term can be put in the dictionary via the dictionary icon 📘.

5.3. Convention for the naming

Consistent naming of subprocesses

Goal

The name of the collapsed subprocess should receive the same name as the diagram.

Description

This rule obtains the linkage of the superprocess with the subprocess. The subprocess should have the same name as the task in the superprocess, for example, "handle an order." A task can also have the name of the process, for example, "order handling." The representation must be consequent.

Definition of required element names of activities

Goal

An activity required a name. The naming of the used activity should represent the task of the employee in this process step.

Definition of required element names of data objects

Goal

A data object must always have a name, because it has to be clear which document is required in the process model.

Description

A data object can be named by a double click in the object where the name of the document must be entered. The name of the data object should make the document clear.

Definition of required element names of events

Goal

All events should have a name. The naming of used events should represent the state of the process.

Description

Events should always have a name. The name must be entered by a double click in the event. The name of the event should represent the state of the process.

Definition of required element names of roles

Goal

Roles must always have a name. The name of the used role should represent the responsible person for the process.

Description

To name a role, it has to be clicked on in the head of the role, where the name has to be entered.

5.4. Convention for the process structure

Absence of Multi-Merges

Goal

Multi-Merges are the opposite of deadlocks. They occur at a false usage of gateways and lead to multiple executions of the following flow. Avoiding this is necessary, since the process often shows an unexpected behavior within a multi-merge

Description

Multi-merges arise through false combined gateways and lead to a multiple execution of the following flow. First at all, the links where gateways have been installed should be checked since this is where most errors occur. Then, if one task has multiple sequence flows, it should be checked because this might be a multi-merge too. Multi-merges arise if for instance a parallel gateway was picked for linkage and this linkage is combined with an exclusive gateway. A multi-merge can also occur when one task is combined with one or several sequence flows. This problem is best resolved with the illustration in gateways.

Absence of deadlocks

Goal

Deadlocks arise at a false usage of gateways and block the further process procedure. Avoiding this is necessary, since the process is not executable when a deadlock exists.

Description

Deadlocks are blockings in the process model, which occur when gateways are used incorrectly. In this case the links in the process where gateways were installed should be checked. Deadlocks occur when an exclusive gateway was picked for linkage and this was combined again with a parallel gateway. They may arise by added intermediate events or multiple exclusive start events, which should be checked again.

Absence of subprocess relation cycles

Goal

Subprocess relations should be strictly hierarchical, due to better clarity. A cycle can develop through false/double linkage.

Description

Cycles occur by false linkage to subprocesses, so the required links should be checked. They may occur when a subprocess is linked to a superprocess. To open a subprocess in the Signavio Process Editor, the + (plus) sign needs to be clicked for the linkage to be opened.

Usage of a restricted number of expanded pools

Goal

To keep diagrams clear and comprehensive, it is best to avoid showing too many pools open in a diagram. Every BPMN pool represents their own process.

Description

Pools, that do not play an important role for the actual process should be closed or modeled as a black box. A black box is an empty pool with only the name of the role. For example, if the pool 'ACME AG' is important, the processes of the client do not play a role for the actual process. These should be modeled by a black box instead.

Absence of splits and join behavior on one element

Goal

The strict separation of branched and merging gateways should be observed, since it leads to better transparency and clarity of the process

Description

If sequence flows are combined at a gateway, a new gateway for further linkage needs to be modeled. A gateway should not have a convergent and merging function at the same time.

Consistent usage of start and end events

Goal

The use of start and end events is necessary to show the different states that begin and complete the modeled process.

Description

Since a process always need at least one start and one end event. They should be shown in the process model. It is necessary to make sure that if a start event is modeled, then an end event is modeled also and vice versa. By filling in the missing events this rule is fulfilled.

Consistency between superprocesses and subprocesses

Goal

There should be a consistency when using superprocesses and subprocesses. This means that the pool of the superprocess needs to be named the same as the pool of the sub process.

Description

It is important that the pools in subprocesses and superprocesses have the same name and order. The same dictionary terms should be used so that the consistency is guaranteed. To achieve the same number and order of pools, they can be either manually created or included in the subprocess by copy and paste.

Correct usage of conditional and default flows

Goal

For models to be generally comprehensive, the correct usage of conditional and default flows are necessary. Therefore, conditional flows may only be used at certain points and there may be only one default flow per object.

Description

Conditional flows are sequent flows which have a condition that should be fulfilled for the process to continue. The default flow stands for a normal sequence flow. This comes into occurrence when none of the conditions are correct.

The conditional flow needs to be described as for instance 'invoice amount 5000' so it´s clear which path takes place in the process. Conditional and normal flows should not be mixed. Behind a gateway there may be just one default flow modeled.

Message exchange between pools

Goal

To obtain a logical process flow, the modeled pools need to be in correlation with each other and have to be linked to the main process.

Description

Each pool should be associated with at least one other pool about message flow, otherwise it is completely detached from the process flow. In the Signavio Process Editor, the pools can be simply linked via the context menu by drag and drop.

Usage of explicit splits

Goal

Splits should be explicitly modeled through a gateway and not indirectly modeled on activities. This is important for clarity and legibility of diagrams.

Description

Gateways are only used for linkage or merging the processes, and therefore, are used even to these functions. A gateway should own either on incoming and several outgoing sequence flows or several incoming and one outgoing sequence flow.

Usage of meaningful gateways

Goal

Since gateways are used only for linkage or merging within processes, they always need to have multiple incoming or outgoing flows.

Description

Gateways are only used for linkage or merging the processes, and therefore, are used even to these functions. A gateway should own either on incoming and several outgoing sequence flows or several incoming and one outgoing sequence flow.

Usage of message flows only in correct nodes

Goal

For distinct and comprehensive modeling, message flows should only be used by the BPMN designated elements. Therefore, the difference between transmitter and receiver has to be considered.

Description

In activities, it must be made sure that the sending activities have outgoing message flows and that the receiving activities have incoming message flows. Going against this rule will violate when the message events are modeled, where there is no incoming or outgoing message flow modeled.

Usage of only one start event in a process

Goal

This rule guarantees that a process stays clear and comprehensive. There is on exact start of a process and it should be defined. If a process has more than one start event, misinterpretations can occur.

Description

The clear definition of the start event is important, because they describe the triggering of an execution. When using multiple start events, it often leads to a misunderstanding. There should only be one start event. By a breach of this rule, it should be ensured that the different start events are correctly used (i.e. exclusive to each other).

Usage of only one start event in a subprocess

Goal

This rule guarantees that a subprocess stays clear and comprehensive. There is one exact start of a sub process and it should be defined. If a subprocess has more than one start event, misinterpretations can occur.

Description

For the clear definition of the start event in the subprocess, the triggers must be described to the execution. At the use of several starting points, it often leads to a misunderstanding, so there should only be one start event. At a breach of this rule, it should be ensured that the different start events are correctly used (i.e. exclusively to each other).

Conventions for the layout

Usage of sufficient distance between elements

Goal

There should be sufficient distance used between elements. The process should be modeled as clear and understandable as possible.

Description

The sufficient distance between the connected elements should be 75%, which will be automatically executed in the Signavio Process Editor. The elements should always be modeled with the same distance to each other. The elements can be ordered automatically, so it is necessary to click on the element. With the context menu of the Signavio Process Editor, it is possible to model other elements.

Compliance of a maximum diagram size

Goal

Large process diagrams are difficult to read and comprehend. Additionally, they tend to contain more errors. Therefore you should try to keep your models small by using subprocesses in separate diagrams.

Description

The maximum diagram size in the Signavio Process Editor is DIN-A3. The compliance of the diagram size is compatible with the modeling of subprocesses; specific tasks including more than one process step. Processes could also be cut into smaller diagrams with linked events.

Usage of the defined edge direction of message flows

Goal

Message flows will be modeled for the message exchange between pools. The modeling direction should always be vertical to the pool.

Description

Message flows should be modeled vertically to the activities.

Usage of the defined edge direction of sequence flows

Goal

The modeling of sequence flows should not be modeled against the defined edge direction. The process model should be modeled as clearly as possible and unnecessary sequence flows should not be placed in the model.

Description

Sequence flows should always be modeled with the chosen modeling direction. It should be considered, that the elements are combined in reading direction (i.e. from left to right).

Consistent edge folding in associations

Goal

An association should be modeled consequent, without any folding.

Description

In the Signavio Process Editor, there are two possibilities for the representation of associations. An association can be modeled over the context menu. For example, a text annotation can be integrated by a simple click (in the context menu). This text annotation can be created automatically and can be moved manually.

Consistent edge folding in message flows

Goal

Message Flows should be modeled constant without any folding. This rule is for the comprehension and readability of the models.

Description

The message flows should be modeled vertically to the pool. If an edge folding in message flows is necessary, then it should be consistent.

Consistent edge folding in sequence flows

Goal

If sequence flows are in the process, they should be constant. The consistent edge folding in sequence flows will be used to get a consistent layout in workflow.

Description

If it is necessary for a sequence flow to be folding, it should be consistent. In the Signavio Process Editor, the edge folding will be executed in a 90 degree angle.

Consistent incoming and outgoing behavior of sequence flows

Goal

If sequence flows are folding in the process, they should be constant. The consistent edge folding in sequence flows will be used to get a consistent layout in the workflow.

Description

To realize the consistent incoming and outgoing behavior of sequence flows and how the sequence flows between the activities are modeled. The sequence flow can go out either up, down or right and come in left, down or up.

Consistent incoming and outgoing behavior of message flows

Goal

The consistent incoming and outgoing behavior of message flows is for the comprehension of clarity of the process.

Description

If message flows are modeled to represent the message exchange between two pools, it should be consistent. The modeled outgoing flow edge of the pool (under or upper) should be consistent with the incoming (under or upper) edge of the relevant tasks.

Absence of edge overlays

Goal

The overlapping of edges is not allowed. The comprehension and the clarity of the model will suffer under the violation of the existing rule.

Description

An overlap disrupts the reading and understanding of the process. If edges overlap each other, the meaning can be interpreted wrong in reference to a join of edges. Overlapping edges should be moved so there is a distance between each other. If it is possible, the edges should not go in the same nodes (or come out of the same node).

Absence of node intersections

Goal

The nodes should not overlap other nodes. The comprehension and clarity of the model will suffer under the violation of the existing rule.

Description

Overlapping nodes disrupt the reading and understanding of the process. If nodes overlap each other, the meaning of the model can be interpreted wrong. The overlapping nodes should be moved so there is a distance in between each other.

Definition of the correct modeling direction

Goal

The correct modeling should be defined before beginning the modeling of processes. This definition should be consistent in the workflows.

Description

The standard modeling direction could be changed in the Signavio Process Editor. The slide on the right side of the tool has the selection between horizontal and vertical modeling directions. A consequent compliance in the chosen direction is absolutely necessary.

Appendix

Checklist for the rules

1.1. Convention for the architecture

- □ Incorporation of open comments
- □ Usage of unique diagram names

1.2. Convention for the notation

- □ Usage of activity descriptions
- □ Definition of required dictionary links of roles
- □ Definition of required dictionary links of data objects

1.3. Convention for the naming

- □ Consistent naming of subprocesses
- □ Definition of required element names of activities
- □ Definition of required element names of data objects
- □ Definition of required element names of events
- □ Definition of required element names of roles

1.4. Convention for the process structure

- □ Absence of Multi-Merges
- □ Absence of deadlocks
- □ Absence of subprocess relation cycles
- □ Usage of a restricted number of expanded pools
- □ Absence of split and join behavior on one element
- □ Consistent usage of start and end events
- □ Consistency between superprocesses and subprocesses
- □ Correct usage of conditional and default flows
- □ Message exchange between pools
- □ Usage of explicit splits
- □ Usage of meaningful gateways
- □ Usage of message flows only in correct nodes
- □ Usage of only one start event in a process
- □ Usage of only one start event in a subprocess

1.5. Convention for the layout

- □ Usage of sufficient distances between elements
- □ Compliance of a maximum diagram size
- □ Usage of defined edge direction of message flows
- □ Usage of defined edge direction of sequence flows
- □ Consistent edge folding in associations
- □ Consistent edge folding in message flows
- □ Consistent edge folding in sequence flows
- □ Consistent incoming and outgoing behavior of message flows
- □ Consistent incoming and outgoing behavior of sequence flows
- □ Absence of edge overlays
- □ Absence of node intersections
- □ Definition of correct modeling direction