SAP Signavio

SAP Signavio Solutions

A practical, detailed, and convenient guide for designing optimal business processes

Table of contents

Introduction

This modeling guidelines manual exists to provide business process management (BPM) professionals with a practical, detailed, and convenient reference guide for **designing optimal business processes**. This manual is best used as a guide for companies seeking to better understand BPM and the industry standard modeling language – Business Process Model and Notation (BPMN), version 2.0.

Content in this manual is suggested material that may 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 this manual are intuitive to follow, even if you have little or no experience with BPM. The next chapter discusses the themes of BPM 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 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 are defined in the fourth chapter. The final chapter describes the modeling rules, which are divided into the following five categories: architecture, notation, naming, process structure, and layout.

To learn more about the goals and the descriptions of the rules, visit us online.

Business process management

Business process management, or BPM, is used to help visualize organizational processes that better align IT with business needs. It 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) 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 into optimizing process flow. BPM is everything from analyzing to designing to executing business processes. The following questions help frame the theories behind the practice.

What is BPM?

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.

What is a process?

A process is a sequence of defined operations, tasks, and activities that can be executed by humans 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.

What are the goals of BPM?

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 workflows to optimize productivity
- Create a high level of transparency for quality
- Management, certification, and accreditation departments
- Control existing and potential costs through process simulation
- Plan optimal usage of available resources
- Improve allocation of resources and lower costs with process simulation and optimization

Is there a general approach to process modeling?

We recommend that you establish a method for designing process models. The following steps can be used as a general means to draft your processes.

Step 1: Define 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: Describe and model "as-is" (current) business processes. These processes can have forks, since there is rarely a happy path for any process.

Step 3: Describe and model "to-be" (future) processes. After unnecessary steps are eliminated, this is the optimized state of processes.

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

What are the necessary process elements?

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

- What process are we designing? *(for example, company procedure)*
- Who is designing the process? *(for example, process owners)*
- Who will be using the process? *(for example, management and process participants)*
- Which methods will be used? *(that is, the necessary tools required to execute the process)*
- Which parameters should be measured? *(for example, costs per activity)*
- What are possible error frequencies? *(for example, how often a particular bottleneck will occur)*

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

Core elements of BPMN

The core elements of Business Process Model and Notation are the most widely used objects within it. This set of elements consists of activities, gateways, events, data objects, artifacts, roles, and connectors. The following descriptions will help you understand the uses for each of the core elements.

Activities

Activities describe the different steps in processes. They should be labeled by the action describing the task (for example, 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.

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 (XOR) gateways are used if only one condition can be chosen. An XOR gateway awaits one incoming branch before triggering the outgoing flow.
- Parallel (AND) gateways activate outgoing flows, and the junction awaits incoming flows before the process flow can continue.
- Inclusive (OR) gateways activate one or more outgoing flows, and the junction awaits 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 or 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.

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 a 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 (for example, receiving an e-mail), and throwing intermediate events are modeled when there is something to do to continue the process (for example, sending an e-mail). End events are the ones that close the running process.

Data objects and artifacts

Data objects can be attached to and used by activities. They represent information, such as IT systems, paper documents, or electronic documents.

Artifacts are additional ways to represent information, for example, annotations or arrangements. These can be directly linked and associated with other elements.

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 (for example, Google). Lanes can then further define the departmental units within pools (for example, Google Marketing and Communications or Google Finance).

Connectors

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

Organization of models

It is important to define a structural approach for models. The overall process architecture is best represented by organizing models into hierarchies that contain specific folders. We describe dividing the folder hierarchy into five different levels, from 0 to 4, explaining them 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.

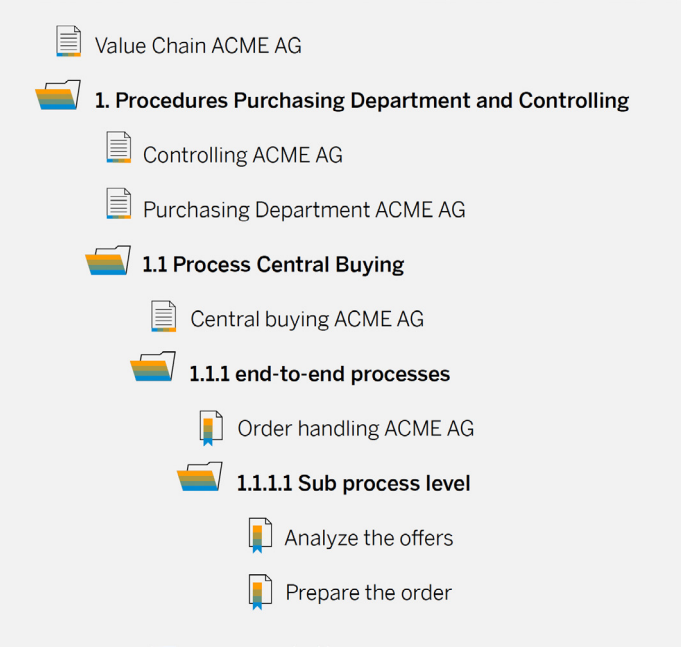
Level 3 should be the end-to-end processes (modeled as a BPMN diagram).

Level 4 should be the subprocesses, modeled as a BPMN diagram as well.

If a process architecture is designed in a folder structure, it can be easily displayed like the image shown below.

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

Figure: Example of a folder structure



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.

Level 1: Departments and interdepartmental Processes
Level 1 allows process participants to explore different departments and the interdepartmental processes.

Level 2: Department and unit processes
Level 2 represents different departments or units. These different departments are also called unit processes.

Level 3 (BPMN): End-to-end processes
In Level 3, processes are modeled in a BPMN diagram. Level 3 represents end-to-end processes only.

Level 4 (BPMN): Optional subprocesses
Level 4 represents the subprocesses of Level 3. This level is optional.

The overall process architecture is best represented by organizing models into hierarchies that contain specific folders.

Description of the rules

The rules below 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 BPM in your organization. This helps ensure that everyone has a mutual understanding of the flow of the collection of models by referencing the same material. Once these rules are defined in your SAP Signavio Process Manager solution workspace, your models are automatically checked against them, and you are warned if any rules 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 elements such as activity descriptions and dictionary links. In the second category, we highlight the rules for notating a process. Process notation involves using glossary terms for different elements or the description of activities. The third rule describes different conventions and styles for naming and labeling processes. It is easier to follow a process structure that implements consistent naming of processes and the elements linked to them. In the fourth category, we describe process-structure rules, which define the semantic and syntactic conventions for process models. This category also defines the rules for checking potential mistakes such as deadlocks or other syntactic elements in the model. The last category defines the rules for the process layout. This set of rules outlines how to properly distance elements, maintain an appropriate diagram size, and correctly define the edge direction of a model (for example, from left to right).

Convention for the architecture

Usage of unique diagram names

Goal: The diagrams should have 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 is a high number of models, the use of numbering in each model is useful. One possibility for a diagram name can be "1.4 Handling of Orders," and the subdiagram 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 can be useful information for the process. It is important to consider open comments. If the comments are relevant to the diagram, they should 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*. The comment will be deleted at the 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 is executed.

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 SAP Signavio Process Manager 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. When 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, a new term can be put in the dictionary through the dictionary icon.

Definition of required dictionary links of 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 the data object and its name are already in the dictionary. Should the required data object not be in the dictionary, a new term can be put in the dictionary through the dictionary icon.

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 requires 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 must 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 must be entered.

Convention for the process structure

Absence of multimerges

Goal: Multimerges 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 multimerge.

Description: Multimerges arise through false combined gateways and lead to multiple executions of the following flow. First, the links where gateways have been installed should be checked, since this is where most errors occur. If one task has multiple sequence flows, it should be checked, because this might also be a multimerge. Multimerges arise if, for instance, a parallel gateway is picked for linkage and this linkage is combined with an exclusive gateway. A multimerge can also occur when one task is combined with one or several sequence flows. This problem is best resolved through the addition of gateways to ensure the correct separation of sequence flows.

Absence of deadlocks

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

Description: Deadlocks are blockages in the process model, and they 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 is picked for linkage and this is combined again with a parallel gateway. They may arise through 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 a false or double linkage.

Description: Cycles occur through a 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 SAP Signavio Process Manager, 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 its 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 split 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 both 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 needs 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, an end event is also modeled, and vice versa. By filling in the missing events, this rule is fulfilled.

Consistency between superprocesses and subprocesses

Goal: There should be consistency when using superprocesses and subprocesses. The pool of the superprocess must have the same name as the pool of the subprocess.

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 consistency is maintained. To achieve the same number and order of pools, they can be either manually created or included in the subprocess through copying and pasting.

Correct usage of conditional and default flows

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

Description: Conditional flows are sequential flows that have a condition that should be fulfilled for the process to continue. The default flow stands for a normal sequence flow. This occurs when none of the conditions are correct.

The conditional flow needs to be described as, for instance, “invoice amount 5000,” so it is clear which path takes place in the process. Conditional and normal flows should not be mixed. There may be just one default flow modeled behind a gateway.

Message exchange between pools

Goal: To obtain a logical process flow, the modeled pools need to be in correlation with each other and must 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 SAP Signavio Process Manager, the pools can be linked with the context menu by dragging and dropping.

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: Only gateways should be used for linking or merging within processes. An activity can have only one outgoing sequence flow. A gateway should have either one 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 linking or merging within processes, they need to have multiple incoming or outgoing flows.

Description: Only gateways should be used for linking or merging within processes. As noted above, a gateway should have one 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 be used by BPMN-designated elements only. Therefore, the difference between transmitter and receiver has to be considered.

Description: Sending activities must have outgoing message flows, and receiving activities must have incoming message flows. Going against this rule violates 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 assures that a process stays clear and comprehensive. There is one 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 it describes the triggering of a process execution. There should be only one start event; using multiple start events often leads to misunderstanding. If this rule is breached, different start events must be used correctly (that is, exclusive to each other).

Usage of only one start event in a subprocess

Goal: This rule ensures that a subprocess stays clear and comprehensive. There is one exact start of a subprocess, 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 for execution must be described. Using several starting points often leads to a misunderstanding, so there should be only one start event. If this rule is breached, different start events must be used correctly (that is, exclusive to each other).

Convention for the layout

Usage of sufficient distance between elements

Goal: There should be sufficient distance between elements. The process should be modeled as clearly and understandably as possible.

Description: The sufficient distance between the connected elements should be 75%, which is automatically executed in SAP Signavio Process Manager. The elements should 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 SAP Signavio Process Manager, it is possible to model other elements.

Compliance of maximum diagram size

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

Description: The maximum diagram size in SAP Signavio Process Manager is DIN-A3. Compliance of the diagram size is compatible with the modeling of subprocesses; specific tasks include more than one process step. Processes can also be cut into smaller diagrams with linked events.

Usage of defined edge direction of message flows

Goal: Message flows should 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 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. Keep in mind that different elements are combined in the reading direction (that is, from left to right).

Consistent edge folding in associations

Goal: An association should be modeled consequent, without any folding.

Description: In SAP Signavio Process Manager, 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 is used to get a consistent layout in the workflow.

Description: If it is necessary for a sequence flow to be folding, it should be consistent. In SAP Signavio Process Manager, the edge folding is executed at 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 is used to get a consistent layout in the workflow.

Description: It is important 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 and clarity of the process.

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

Absence of edge overlays

Goal: Overlapping of edges is not allowed. The comprehension and the clarity of the model suffers if the existing rule is violated.

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

Absence of node intersections

Goal: The nodes should not overlap other nodes. The comprehension and clarity of the model suffers if the existing rule is violated.

Description: Overlapping nodes disrupt the reading and understanding of the process. If nodes overlap each other, the meaning of the model can be interpreted incorrectly. Overlapping nodes should be moved so there is distance between them.

Definition of correct modeling direction

Goal: Correct modeling should be defined before beginning the modeling of processes. This definition should be consistent in the workflows.

Description: The standard modeling direction can be changed in SAP Signavio Process Manager. 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 necessary.

Appendix

The following is a checklist for the rules.

Convention for the architecture
- ☐ Usage of unique diagram names
- ☐ Incorporation of open comments

Convention for the notation
- ☐ Usage of activity descriptions
- ☐ Definition of required dictionary links of roles
- ☐ Definition of required dictionary links of data objects

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

Convention for the process structure
- ☐ Absence of multimerges
- ☐ 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

Convention for the layout
- ☐ Usage of sufficient distance between elements
- ☐ Compliance of 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 sequence flows
- ☐ Consistent incoming and outgoing behavior of message flows
- ☐ Absence of edge overlays
- ☐ Absence of node intersections
- ☐ Definition of correct modeling direction