State Machine diagram

A State Machine diagram falls under the behavioral diagramming family.

The behavior of objects of a Class is defined in terms of States and Events, using a State Machine connected to the Class under a construction.

The State Machine is a specification of the sequence of states an object or an interaction undergoes in response to events during its life, combined with its responsive actions. The State Machine can represent the sequence of states of a particular collaboration, e.g. collection of objects, or even the whole system, which is also considered a collaboration. The abstraction of all possible states defined in a State Machine is similar to the way Class diagrams are abstracted: all possible object types (classes) of a particular system are described.

Objects that do not present a very pronounced reactive behavior can always be considered to stay in the same state. In this case, their classes do not possess a State Machine.

State Machine diagrams (also called Statechart diagrams) represent the behavior of entities capable of dynamic behavior by specifying their response to the receipt of event instances. Typically, State Machine diagrams describe the behavior of Classes, but the Statecharts can also describe the behavior of other model entities. such as Use Cases, Actors, Subsystems, Operations, or methods.

A State Machine diagram is a graph that represents a State Machine. States and various other types of vertices (pseudostates) in the State Machine graph are rendered by the appropriate State and Pseudo States symbols, while Transitions are generally rendered by directed arcs that interconnect them. The states can also contain subdiagrams by a physical containment or tiling. Note that every State Machine has a top State containing all the other elements of the entire State Machine. The graphical rendering of this top state is optional.

The states are represented by the State symbols, while the Transitions are represented by arrows connecting the state symbols.

The State diagram is concerned with internal object changes, as opposed to the external object interaction in a Communication. Do not attempt to draw them for all classes in the system; they are only used for modeling a complex behavior. The State diagram shows all the possible states that objects or collaborations may have, and the events that cause the state to change. For example, an event can be another object sending a message that a specified time has elapsed, or that some conditions have been fulfilled. A change of a State is called a Transition. A Transition may also have an action connected to it that specifies what should be done in connection with the state transition.

Displaying inner elements

To display inner elements

- When creating a new State Machine diagram, right-click a State in the Containment tree, point to **Create Diagram**, and then click **State Machine Diagram**.

  > Inner elements are displayed on the State Machine diagram pane automatically upon creating the first diagram under the State element.

- When modifying a preexisting State Machine diagram, select a diagram pane or the State shape and perform one of the following steps:
  - From the shortcut menu, click **Display** > **Display Inner Elements**.
  - On the diagram toolbar, click  and select **Display Inner Elements**.

Working with behaviors

You can select Entry, Do Activity, Exit, deferrable triggers, and internal transitions directly in the compartment of the state shape. Thus, you can:

- Draw dependencies among particular behaviors on a state shape.
- Move behaviors from one state shape to another.
- Convert the selected behavior to another.

To convert a behavior to another behavior

---

1. On a state shape, select a behavior.
2. From the behavior's shortcut menu, select **Refactor** > **Convert To** > <*behavior*>

Related pages

- Creating diagrams
- Dragging in State Machine diagrams