State

A State is a condition during the lifetime of an object or an interaction during which the object meets certain conditions, performs an action, or waits for an event. The State is defined by the concepts of duration and stability. An object can not be in an unknown or undefined State. A State may have two compartments to provide more information about that State:

- The first compartment is the name compartment, it contains the State name, for example: running, going up.
- The second compartment is the activity compartment, it contains the events and actions of the State.

You can specify State properties in the State Specification window. In the same window, you can find the description of each property. Descriptions are presented in the description area of the Specification window

Simple State

A simple State is a State without any regions, internal Vertices or Transitions.

Composite State

You can create a composite State by converting a simple State.

A composite State either contains one region or is decomposed into two or more orthogonal regions. Each region has a set of mutually exclusive disjoint subvertices and a set of transitions. A given State may only be decomposed in one of these two ways.

Any State enclosed within a region of the composite State is called a subState of that composite State. It is called a direct subState when it is not contained by any other State; otherwise it is referred to as an indirect subState.

Each region of the composite State may have an initial pseudoState and a final State. A transition to the enclosing State represents a transition to the initial pseudoState in each region.

You can specify composite State properties in the State Specification window. In the same window, you can find the description of each property. Descriptions are presented in the description area of the Specification window.

Orthogonal State

An orthogonal State is a composite State with at least 2 regions.

Submachine State

A submachine State specifies the insertion of the specification of a submachine State Machine. The State machine that contains the submachine State is called the containing State machine. The same State machine can be a submachine more than once in the context of a single containing State machine.

The submachine State is semantically equivalent to a composite State. The regions of the submachine State machine are the regions of the composite State. The entry, exit, behavior actions, and internal transitions, are defined as part of the State. The submachine State is a decomposition mechanism that allows factoring of the common behaviors and their reuse.

You can specify submachine properties in the State Specification window. In the same window, you can find the description of each property. Descriptions are presented in the description area of the Specification window.

On this page

Related Pages

- Model Elements
- Stereotype
- State Machine diagram

Related References

- Assigning behavior to state
- Managing regions
- Changing state to composite/submachine /orthogonal state
- Connection Point Reference
- State Invariant
  - Defining State Invariant