**State Diagrams**

**Introduction**

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system. State diagrams are also referred to as State machines and State-chart Diagrams. These terms are often used interchangeably. So simply, a state diagram is used to model the dynamic behavior of a class in response to time and changing external stimuli. State-chart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

**Purpose of state diagrams**

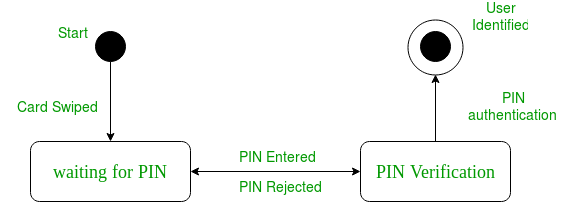
State diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events. State diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events. State diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of State diagram is to model lifetime of an object from creation to termination.

The main purposes of using a state diagram are:

* To model the dynamic aspect of a system.
* To model the life time of a reactive system.
* To describe different states of an object during its life time.
* Define a state machine to model the states of an object.

**Example state diagram**

This state diagram shows the different states in which the verification sub-system or class exist for a particular system.



**Components of a state diagram**

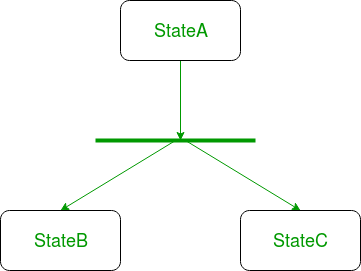
1. **Initial state –** We use a black filled circle represent the initial state of a System or a class.

UML-State-Diagram

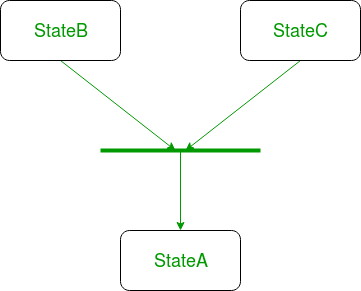
1. **Transition –** We use a solid arrow to represent the transition or change of control from one state to another. The arrow is labelled with the event which causes the change in state.

UML-State-Diagram

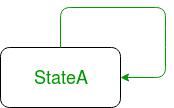
1. **State –** We use a rounded rectangle to represent a state. A state represents the conditions or circumstances of an object of a class at an instant of time.  
   UML-State-Diagram
2. **Fork –** We use a rounded solid rectangular bar to represent a Fork notation with incoming arrow from the parent state and outgoing arrows towards the newly created states. We use the fork notation to represent a state splitting into two or more concurrent states.



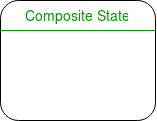
1. **Join –** We use a rounded solid rectangular bar to represent a Join notation with incoming arrows from the joining states and outgoing arrow towards the common goal state. We use the join notation when two or more states concurrently converge into one on the occurrence of an event or events.



1. **Self-transition –** We use a solid arrow pointing back to the state itself to represent a self-transition. There might be scenarios when the state of the object does not change upon the occurrence of an event. We use self-transitions to represent such cases.



1. **Composite state –** We use a rounded rectangle to represent a composite state also. We represent a state with internal activities using a composite state.



1. **Final state –** We use a filled circle within a circle notation to represent the final state in a state machine diagram.

UML-State-Diagram

Sources:

<https://www.techopedia.com/definition/16446/state-diagram>

<https://www.tutorialspoint.com/uml/uml_statechart_diagram.htm>

<https://www.geeksforgeeks.org/unified-modeling-language-uml-state-diagrams/>