State Diagram

A state diagram shows the behavior of classes in response to external stimuli. Specifically a state diagram describes the behavior of a single object in response to a series of events in a system. Sometimes it's also known as a Harel state chart or a state machine diagram. This UML diagram models the dynamic flow of control from state to state of a particular object within a system.

What is the Difference between a State Diagram and a Flowchart?

A flowchart illustrates processes that are executed in the system that change the state of objects. A state diagram shows the actual changes in state, not the processes or commands that created those changes.

Steps to Drawing a State Diagram

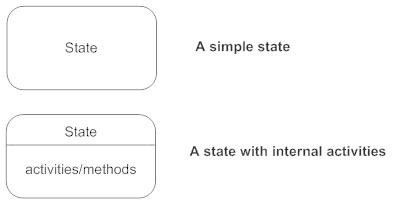
Before you begin your drawing find the initial and final state of the object in question.

Next, think of the states the object might undergo. For example, in e-commerce a product will have a release or available date, a sold out state, a restocked state, placed in cart state, a saved on wish list state, a purchased state, and so on.

Certain transitions will not be applicable when an object is in a particular state, for example a product can be in a purchased state or a saved in cart state if its previous state is sold out.

Basic Statechart Diagram Symbols and Notations

**States**  
States represent situations during the life of an object. You can easily illustrate a state by using a rectangle with rounded corners.



**Transition**  
A solid arrow represents the path between different states of an object. Label the transition with the event that triggered it and the action that results from it. A state can have a transition that points back to itself.

Transitions - State diagram

**Initial State**  
A filled circle followed by an arrow represents the object's initial state.

Inititial state - State diagram

**Final State**  
An arrow pointing to a filled circle nested inside another circle represents the object's final state.

Final state - State diagram

**Synchronization and Splitting of Control**  
A short heavy bar with two transitions entering it represents a synchronization of control. The first bar is often called a fork where a single transition splits into concurrent multiple transitions. The second bar is called a join, where the concurrent transitions reduce back to one.

