UML STATE CHART DIAGRAM

STATE MACHINE DIAGRAM:

Use case and scenarios provide a way to describe system behavior in the form of interaction between objects in the system. Some times it is necessary to consider inside behavior of an object.

A state chart diagram shows the state of a single object, the events or messages that causes a transition from one state to another and the actions that result from a state change. As in activity diagram ,state chart diagram also contains special symbols for start state and stop state.

State chart diagram cannot be created for every class in the system, it is only for those class objects with significant behavior STATE.

UML notation for STATE is

To identify the states for an object its better to concentrate on sequence diagram. In our application the object for Account may have in the following states, initialization, open and closed state. These states are obtained from the attribute and links defined for the object. Each state also contains a compartment for actions.

ACTIONS:

Actions on states can occur at one of four times: on entry

on exit do

On event.

On entry: What type of action that object has to perform after entering into the state?

On exit: What type of action that object has to perform after exiting from the state?

Do: The task to be performed when object is in this state, and must to continue until it leaves the state.

On event: An on event action is similar to a state transition label with the following syntax: event(args)[condition]: the Action

STATE TRANSITION: A state transition indicates that an object in the source state will perform certain specified actions and enter the destination state when a specified event occurs or when certain conditions are satisfied. A state transition is a relationship between two states, two activities, or between an activity and a state.

We can show one or more state transitions from a state as long as each transition is unique. Transitions originating from a state cannot have the same event, unless there are conditions on the event.

Provide a label for each state transition with the name of at least one event that causes the state transition. You do not have to use unique labels for state transitions because the same event can cause a transition to many different states or activities.

Transitions are labeled with the following syntax: event (arguments) [condition] / action ^ target.sendEvent (arguments)

Only one event is allowed per transition, and one action per event.

STATE DETAILS:

Actions that accompany all state transitions into a state may be placed as an entry action within the state. Like wise that accompany all state transitions out of a state may be placed as exit actions within the state. Behavior that occurs within the state is called an activity.

An activity starts when the state is entered and either completes or is interrupted by an outgoing state transition. The behavior may be a simple action or it may be an event sent to another object

UML notation for State Details

Purpose of state chart diagrams: We use state chart diagram when working on real-time process control applications or systems that involve concurrent processing. It will also be used when showing the behavior of a class over several use cases.

State chart diagrams are used to model dynamic view of a system.

State chart diagrams are used to emphasizing the potential states of the objects and the transitions among those states.

State chart diagrams are used to modeling the lifetime of an object.

State chart diagrams are used to model the behavior of an interface. Although an interface may not have any direct instances, a class that realizes such an interface may. Those classes conform to behavior specified by the state machine of this interface.

State chart diagrams are used to focus on the changing state of a system driven by events.

This diagram is also for constructing executable systems through forward and reverse engineering.

To model reactive objects, especially instances of a class, use cases, and the system as a whole.

ELEMENTS OF STATE CHART DIAGRAMS:

1. **State:** It is a condition or situation during the life of an object during which it satisfies some condition, performs some activity, or waits for some event.
2. **Event:** It is the specification of significant occurrence that has a location in time and space.
3. **Transition:** It is a relation between two states indicating that an object in the first state will perform certain actions and enter the second state when a specified event occurs and conditions are satisfied.
4. **Action state:** An action state is shorthand for a state with an entry action and at least one outgoing transition involving the implicit event of completing the entry action.
5. **Sequential sub state:** A submachine state represents the invocation of a state machine defined elsewhere. The submachine state is depicted as a normal state with the appropriate “include” declaration within its internal transitions compartment. As an option, the submachine state may contain one or more sub states, which represent already created states.
6. **Concurrent sub state:** A concurrent state is divided into two or more sub states. It is a state that contains other state vertices. Naturally, any sub state of a concurrent state may also be a composite state of either type. Any state enclosed within a composite state is called a sub state of that concurrent state.
7. **Initial state:** A pseudo state to establish the start of the event into an actual state.
8. **Final state:** The final state symbol represents the completion of the activity.
9. **History state:** History state is a state machine describes the dynamic aspects of an object whose current behavior depends on its past.
10. **Vertical Synchronization:** This merge branch bar symbol is also known as a “Synchronization Bar”. It merges concurrent transitions to a single target. It splits a single transition into parallel transitions.
11. **Horizontal Synchronization:** This merge branch bar symbol is also known as a

“Synchronization Bar”. **12. Guard conditions:** Activity and state diagrams express a decision when conditions are used to indicate different possible transitions that depend on Boolean conditions of container object. UML calls those conditions as guard conditions.

13.Forks and joins: A fork construct is used to model a single flow of control that divides into two or more separate, but simultaneous flows

STATE CHART DIAGRAM FOR ONLINE VOTING SYSTEM