State Machine Diagram

Here is a list of steps that will help you get started in developing state machine diagrams:

1. **Review the class diagram and select the classes that might require state machine diagrams.** Only include those classes that have multiple status conditions that are important for the system to track. Then, begin with the classes that appear to have the simplest state machine diagrams, such as the SaleItem class for RMO.

2. **For each selected class in the group, make a list of all the status conditions you can identify.** At this point, simply brainstorm. Remember that these states must reflect the states for the real-world objects that will be represented in software. Sometimes, it is helpful to think of the physical object, identify states of the physical object, and then translate those that are appropriate into corresponding system states or status conditions. It is also helpful to think of the life of the object. Think of activities done to the object or by the object. Often, the object will be in a particular state as these actions are occurring.

3. **Begin building state machine diagram fragments by identifying the transitions that cause an object to leave the identified state.** For example, if a sale is in a state of Ready to be shipped, a transition such as beginShipping will cause the sale to leave that state.

4. **Sequence these state-transition fragments in the correct order.** Then, aggregate these combinations into larger fragments. As the fragments are being aggregated into larger paths, it is natural to begin to look for a natural life cycle for the object.

5. **Review the paths and look for independent, concurrent paths.** When an item can be in two states concurrently, there are probably one or more concurrent paths.

6. **Look for additional transitions.** Often, during a first iteration, several of the possible combinations of state-transition-state are missed. One method to identify them is to take every paired combination of states and ask whether there is a valid transition between the states. Test for transitions in both directions.

7. **Expand each transition with the appropriate message event, guard-condition, and action-expression.** Include with each state appropriate action-expressions.

8. **Review and test each state machine diagram.** Review each of your state machine diagrams to make sure the names of the states describe the object's status condition, to make sure you have identified all the transitions, to check for all concurrent paths, and to ensure you have the exception conditions.