CHAPTER 13

Statechart Diagrams

MAJOR TOPICS

OBJECTIVES

At the completion of this chapter, you will be able to:

- Interpret statechart diagrams.
- Develop statechart diagrams to illustrate the internal workings of individual classes.

PRE-TEST QUESTIONS

The answers to these questions are in Appendix A at the end of this manual.

1. What is a statechart diagram?

   .....

   .....

2. When is a superstate used in a statechart diagram?

   .....

   .....

INTRODUCTION

In an earlier chapter, you used class diagrams to illustrate how individual use cases will be realized by a set of classes and static relationships. In this chapter, you will learn how statechart diagrams can be used to illustrate the complete life cycle of a single object as it exists throughout all use cases.

Statechart diagrams are similar to the activity diagrams you learned about in an earlier chapter. As with activity diagrams, a closed circle represents the start of a sequence diagram. Objects transition through a series of states until they destruct. States are represented by rectangles with rounded corners, and transitions between states are represented by arrows.

STATES

States are represented by rectangles with rounded corners. Each state is labeled with a state name. States may also be labeled with an activity. Figure 13-1 is an example of a state. The state name is Adding Assets. Below the state name, the activity "do/add assets" means that as long as the object remains in this state, it will be in the process of adding assets.

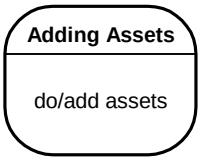

Figure 13-1: State example

TRANSITIONS

The transition from one state to another is represented by an arrow. Figure 13-2 is a statechart diagram for the CheckOutController class. The transition from the starting point to the first state, Adding Assets, is automatic. As soon as a new CheckOutController object is instantiated, it is moved into the Adding Assets state. Two transitions exit the Adding Assets state; these transitions are labeled to indicate when each will be followed.

The labels associated with transitions take the form of Event [Guard] / Action. All three sections of the label are optional. The Event portion of the label describes an event that will precipitate the transition. A Guard condition is a true-or-false condition that must be true in order for the transition to be followed. The Action is a description of the action that is taken following a transition.

In Figure 13-2, two transitions exit the Adding Assets state. Notice that the Guard conditions associated with these transitions are mutually exclusive: Only one can be true at a time.

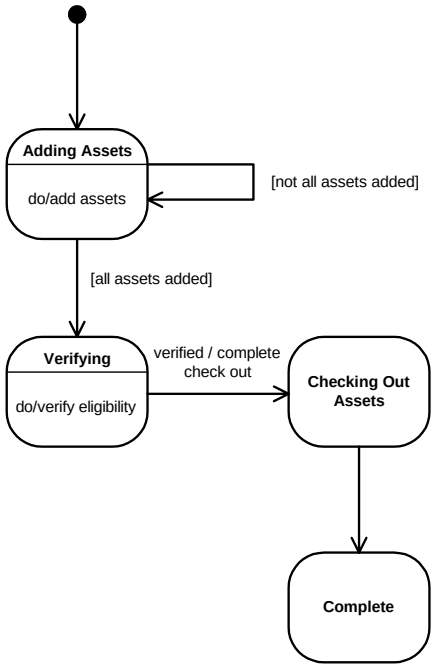

Figure 13-2: Statechart diagram example

SUPERSTATES

Situations often arise in which multiple states want to transition to the same state through a single transition type. In the library system example, a librarian may choose to cancel the check-out process at any time before the CheckOutController object reaches the Complete state. All three of the states (Adding Assets, Verifying, and Checking Out Assets) can transition to the Cancelled state if the user decides to cancel. Figure 13-3 illustrates the difficulty in diagramming this situation.

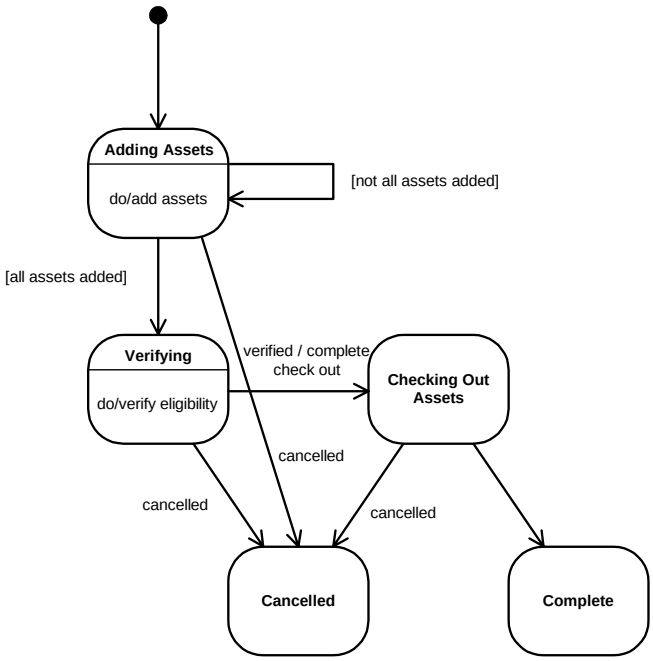

Figure 13-3: CheckOutController statechart diagram without superstate

Instead, you can use a superstate. A superstate contains other states. Figure 13-4 is the CheckOutController class statechart diagram using a superstate. The Adding Assets, Verifying, and Checking Out Assets states are contained within a superstate called Active. As substates, they inherit any transition in which the Active superstate can participate. A single transition between the Active state and the Cancelled state makes this diagram functionally equivalent to Figure 13-3.

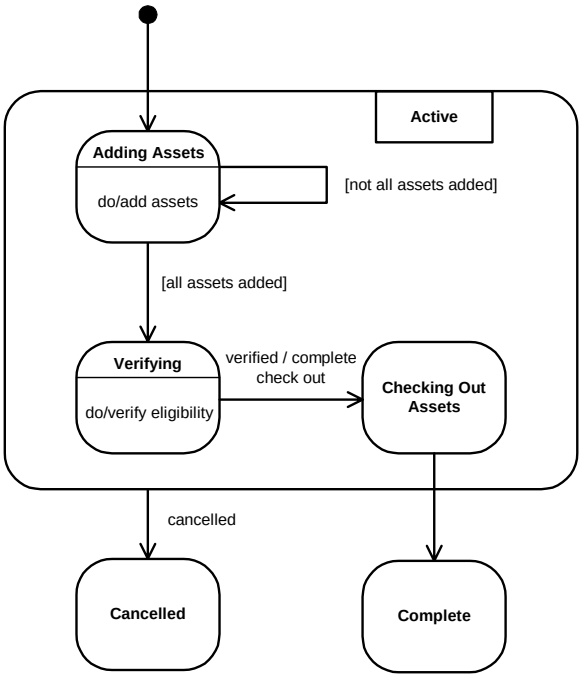

Figure 13-4: CheckOutController statechart diagram with superstate

Exercise 13-1: Creating statechart diagrams

1. Consider the design classes you identified in Exercise 15-1. Determine which design classes would benefit from statechart diagramming.
2. Using Argo/UML, create a statechart diagram for each of the design classes you identified in Step 1 of this exercise.

SUMMARY

Statechart diagrams are used to illustrate the complete life cycle of a single object as it exists throughout all use cases. Statechart diagrams illustrate the series of state through which an object will transition between its instantiation and destruction. States are represented by rectangles with rounded corners. Transitions are represented by arrows. Superstates can be used to make a statechart diagram easier to read.

POST-TEST QUESTIONS

The answers to these questions are in Appendix A at the end of this manual.

1. What do you use to represent a state in a statechart diagram?

   .....

   .....

2. What do you use to represent a transition in a statechart diagram?

   .....

   .....

.....

.....