,ch12.18796 Page 205 Wednesday, May 9, 2001 4:05 PM

CHAPTER 12

Activity Diagrams

Figure 12-1: Behavioral Model View—Activity Diagrams

Activity diagrams (Figure 12-1) lie within the behavioral view of a system and render the activities or actions of a class participating in behavior. These diagrams describe the behavior of a class in response to internal processing rather than external events as in statechart diagrams (Chapter 11). Activity diagrams describe the processing activities within a class. These diagrams contain action states, action flows, and object flows. They may be constructed using the following technique:

1. Identify a class participating in behavior whose processing activities are to be specified.
2. Model swimlanes.
3. Model action states.
4. Model action flows.
5. Model object flows.
6. Refine and elaborate as required.

Activity diagrams

- Are variations of state machines or statechart diagrams (Chapter 11), in which states are activities representing operations, and transitions are triggered by the completion of actions within states.
- May be associated with classes, methods, or use cases (Chapter 8). They describe activities among a set of objects.
- May have the various expressions (guard conditions and actions) they utilize expressed using pseudocode or another language.
- Are types of interaction diagrams that focus on the flow of control and information within behavior.
- May have the various expressions (conditions, etc.) they utilize be expressed using pseudocode or another language.
- May be used to validate use cases.
- May be used for business modeling.

Swimlanes

Swimlanes (Figure 12-2) are graphical constructs that represent a partitioned set of actions. They are used to model the responsibilities of one or more objects for actions within an overall activity; that is, they divide the activity states into groups and assign these groups to objects that must perform the activities.

Figure 12-2: Swimlanes

Swimlanes

- Are denoted as regions separated from neighboring swimlanes by vertical solid lines.
- Have no significance to their order.
- May contain activities or action states.
- May have transitions that cross the lanes to get to their actions or target action states.

- Must have a name or identifier string that represents the name of the object responsible for the actions within the swimlane.

Action States

Action states (Figure 12-3) are states representing the execution of atomic actions or operations. They are used to model atomic actions of entities or steps in the execution of an algorithm.

Figure 12-3: Action State

Action states

- Are denoted as shapes with straight top and bottom lines and convex arcs on the sides.
- Are regular statechart diagram states with other constraints. That is, they may use other notation of statechart diagram states.
- Are shorthand for states with an internal action and at least one outgoing transition for the implicit event of completing the internal action.
- Must have an internal action expression that indicates the action within the state. These actions are equivalent to actions within statechart diagrams (Chapter 11).
- Must be assigned to a single swimlane. The object responsible for the swimlane's activities performs the action states in the swimlane.
- Must not have an entry action.
- Must not have an exit action.
- Must not have internal transitions.
- May have incoming transitions. These are action flows or object flows.
- May have outgoing transitions. These must not be based on explicit events. These are action flows or object flows.
- May use the links and attributes of the owning object to specify the action.
- Need not be unique within a diagram. That is, they may appear more than once within the same diagram. These are distinct states with the same action.
- May be activity states when representing nonatomic actions or operations. These activities require time to complete.
- Must defer all nonrelevant events so they are consumed when they become relevant. Relevant events are those events that appear on outgoing transitions from a state, and nonrelevant events are those that do not appear on outgoing transitions from the state.

Action Flows

Action flows (Figure 12-4) are associations between action states. They are used to model relationships between the action states of an entity. Action flows are activities performed by objects on objects. An entity in one action state will perform the specified action and enter another action state when the first action is completed.

Figure 12-4: Action Flow

Action flows

- Are denoted as solid paths.
- Specify the flow of control between states.
- Are regular statechart diagram transitions with other constraints. That is, they may use other notation of statechart diagram transitions.
- May have square brackets containing a guard condition. Guard conditions
  - Are Boolean conditions that must be satisfied in order to enable the transition to fire.
  - May use the attributes and links of the object that owns the state machine.
- May have a forward slash followed by an action expression that indicates the resulting actions when the transition fires. These actions are equivalent to actions within statechart diagrams (Chapter 11).
- May be omitted between two action states when an object flow exists between the states. That is, when an action produces some output, and this output becomes the input of a subsequent action, the object flow relationship implies an action flow.

Object Flows

Object flows (Figure 12-5) are associations between action states and objects. They are used to model the utilization of objects by action states and the influence of action states on objects.

Figure 12-5: Object Flows

Object flows

- Are denoted as dashed paths.
- Specify the flow of objects between action states.
- May have object symbols (Chapter 7). Objects
  - May be the output of actions. This means that the action creates the object or modifies the object. This relationship is depicted as a dashed arrow from the action to the object symbol.
  - May be the input to actions. This means that the action uses the object. This relationship is depicted as a dashed arrow from the object symbol to the action.
  - May be manipulated by any number of activities. The output object of an action state may be input to one or more action states.
  - May appear multiple times on the same diagram. Each appearance denotes a different point in the life of the object. The object's state may be specified and placed in square brackets appended to the name of the object (Chapter 7).

,ch12.18796 Page 210 Wednesday, May 9, 2001 4:05 PM