Lecture 17: States and State Diagrams

Kenneth M. Anderson

Object-Oriented Analysis and Design

CSCI 6448 - Spring Semester, 2001
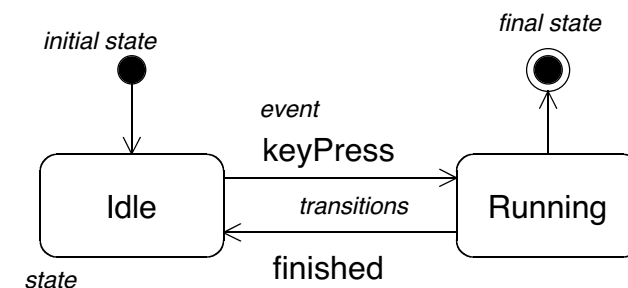

States

- A *state* is a condition or situation during the life of an object that
    - satisfies some condition
    - performs some activity
    - or waits for some event
- An object *transitions* from one state to another when an *event* occurs
- As discussed previously, an event is an instantaneous and atomic transition between states; no state occurs during an event

---

Parts of a State

- Name - an identifier for the state; states can be anonymous (e.g. not named)
- Entry/Exit actions - actions that are executed when entering and exiting a state, respectively
    - an action is an atomic computation that results in a change of state of the object or the return of a value
- Internal Transitions - transitions that do not cause a change in state
- Substates - a state may contain substates that can be arranged either sequentially or concurrently
- Deferred Events - a list of events not handled in this state but which are postponed and handled in some other state

---

Simple Example

Note: a state diagram shows the states for a <u>single</u> object

Events

- An event is a specification of a significant occurrence
  - The act of turning a system off, for instance, can be specified by a shutdown event;
    - when the system turns offs, a shutdown event is generated (and those who are interested in this event can respond)
  - In OO systems, events will correspond to signals, method calls, passage of time, and the state change of other objects
    - e.g. object b switching from state 1 to state 2 can generate an event to which some other object may respond

Transitions

- A transition is a relationship between two states
  - indicating that an object in the first state will perform certain actions and enter the second state
  - when a specified events occurs and specified conditions are satisfied
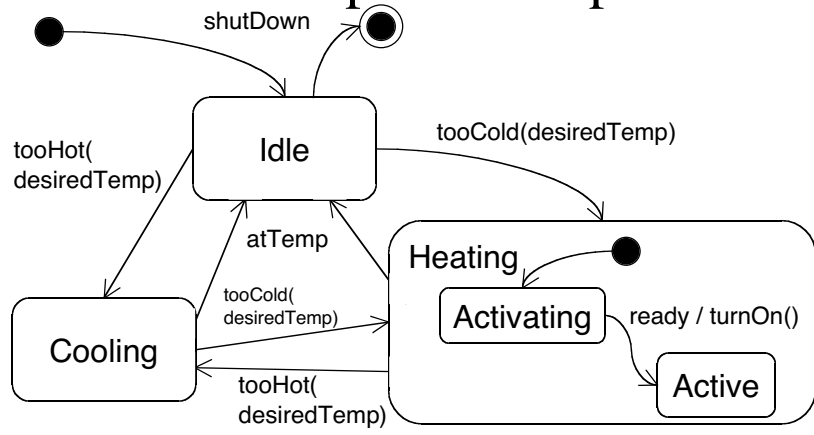
Transitions, continued

- More formally, a transition has the following parts
  - Source State - the initial state
  - Event Trigger - the event whose reception triggers the transition
  - Guard Condition - a boolean expression that enables/disables the transition
  - Action - an action that occurs during the transition
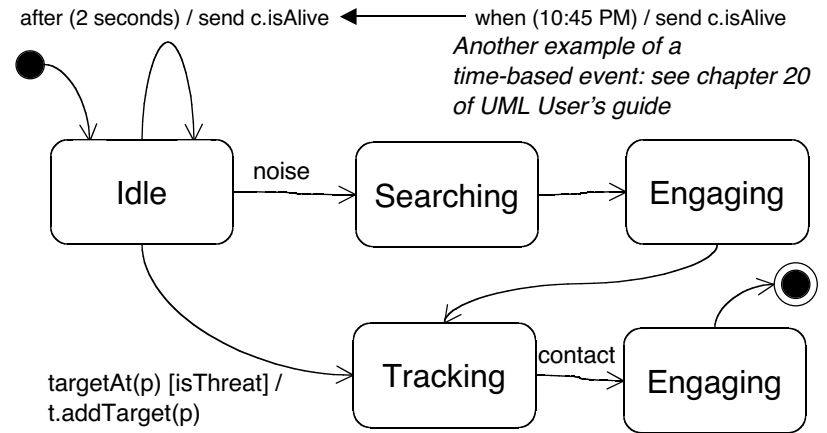 - Target State - the destination state

Transition Labels

- Transitions are labeled in the following format
  - Event [Guard] / Action
  - Each of these parts are optional
- Examples
  - Graduation [person.isGraduated] / throw mortar board
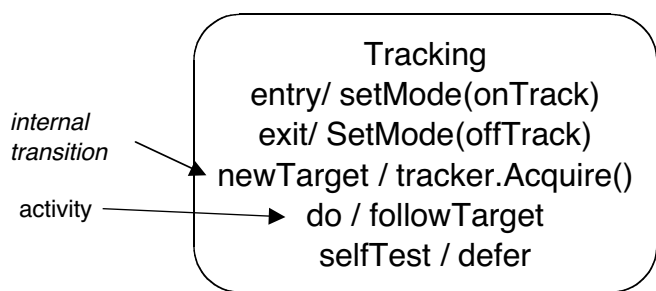 - / get first element
  - [object.isValid]

More complex example

shutDown

tooHot(desiredTemp)

Idle

tooCold(desiredTemp)

atTemp

Heating

tooCold(desiredTemp)

Cooling

Activating

ready / turnOn()

Active

tooHot(desiredTemp)

Transition Example

after (2 seconds) / send c.isAlive

when (10:45 PM) / send c.isAlive
Another example of a time-based event: see chapter 20 of UML User's guide

Idle

noise

Searching

Engaging

targetAt(p) [isThreat] / t.addTarget(p)

Tracking

contact

Engaging

Advanced States

internal transition

activity

Tracking
entry/ setMode(onTrack)
exit/ SetMode(offTrack)
newTarget / tracker.Acquire()
do / followTarget
selfTest / defer

Events with an action of a "defer" are saved until a subsequent state that lists a transition with the name of the deferred event; once that state is reached the deferred event occurs as if it was happening for the first time.

Concurrent States

Cancelled

cancel order

[item not present]

Waiting

Checking

[item present]

Dispatching

Delivered

Authorizing

Authorized

invalid credit card

Rejected