

REQUIREMENTS MODELING: FLOW, BEHAVIOR, PATTERNS, AND WEBAPPS

IDENTIFYING EVENTS WITH THE USE CASE

- an event occurs whenever the system and an actor exchange information
- A use case is examined for points of information exchange

The homeowner uses the keypad to key in a four-digit password. The password is compared with the valid password stored in the system. If the password is incorrect, the control panel will beep once and reset itself for additional input. If the password is correct, the control panel awaits further action.

- An actor should be identified for each event
- the information that is exchanged should be noted
- any conditions or constraints should be listed

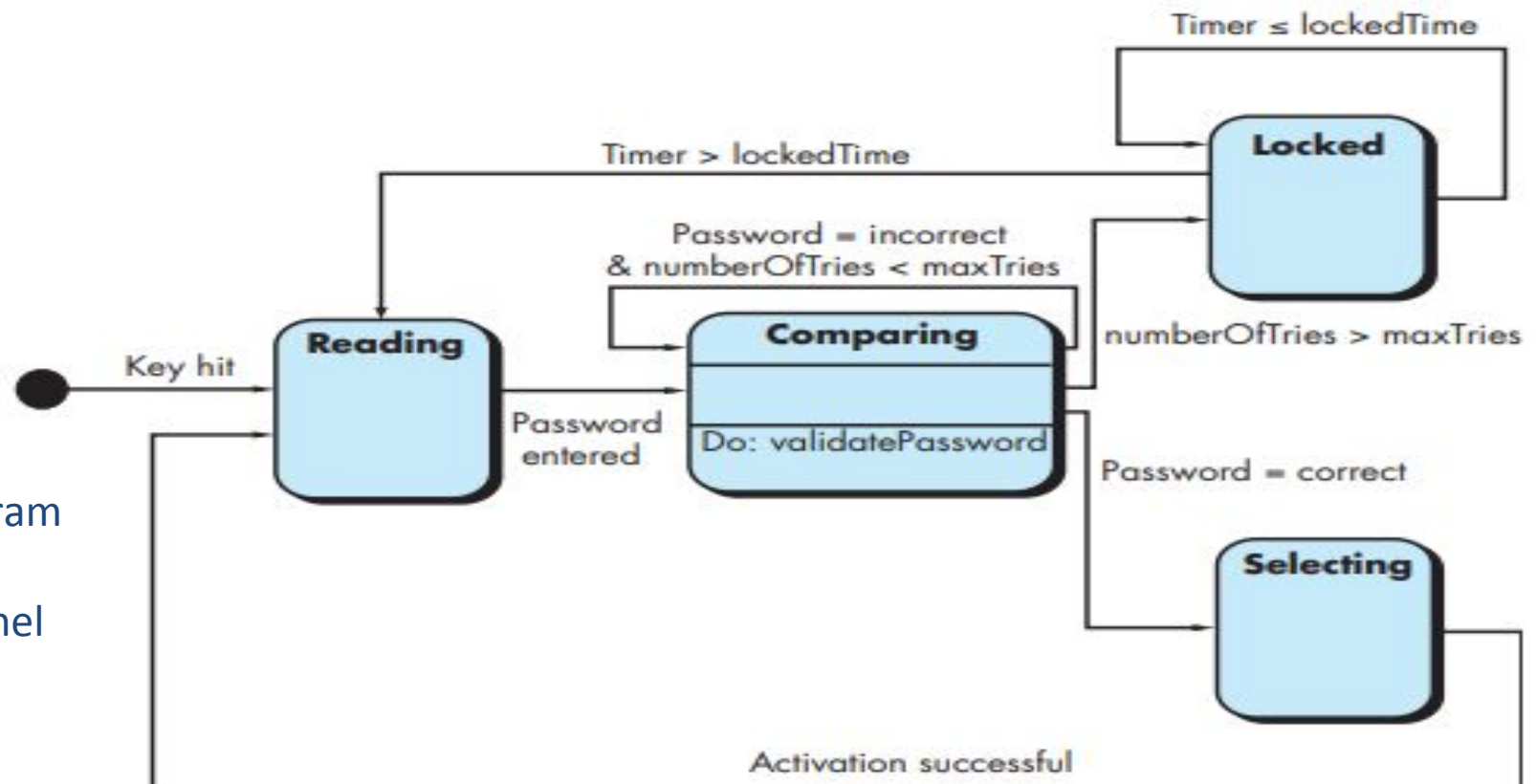
- Example of a typical event
 - Homeowner uses the keypad to key in four-digit password
 - The object 'Homeowner' transmits an event to 'control panel' object
 - Event might be 'password entered'
 - Information transmitted is four digits that constitute the password
 - The event 'password entered' does not explicitly change flow of control
 - The event 'password compared' has an explicit impact on flow of control
- Once all events have been identified, they are allocated to objects involved
- Objects are responsible for generating events/recognizing events
 - Homeowner generates the 'password entered' event
 - Control panel recognizes the binary result of the 'password compared' event

State Representations

- two different characterizations of states must be considered
- the state of each class as the system performs its function
- the state of the system as observed from the outside
- State of the class takes on both passive and active characteristics
 - Passive state
 - Current status of all of an object's attributes
 - Current position and orientation attributes of player
 - Active state
 - Current status of the object as it undergoes continuing transformation/processing
 - Player might have the following activities
 - Moving, at rest, injured, being cured,
- Event must occur to force an object to make transition from one active state to another

State Representations

- two different characterizations of states must be considered
 - the state of each class as the system performs its function
 - the state of the system as observed from the outside



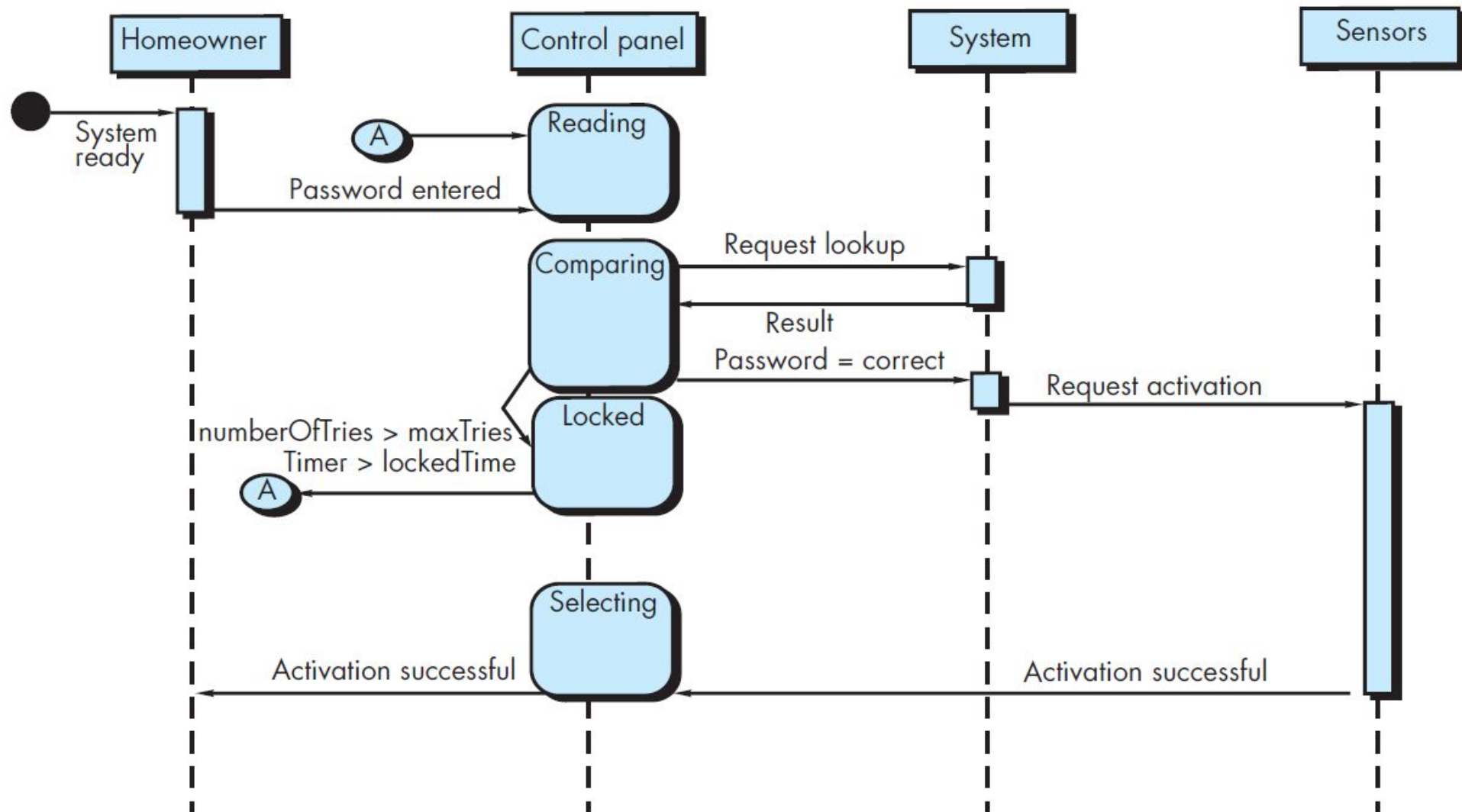
State diagram
for the
ControlPanel
class

- The state of a class takes on both passive and active characteristics
- ***passive state*** is simply the current status of all of an object's attributes
- The ***active state*** of an object indicates the current status of the object as it undergoes a continuing transformation or processing
- An event (trigger) must occur to force an object to make a transition from one active state to another
- A guard is a Boolean condition that must be satisfied in order for the transition to occur

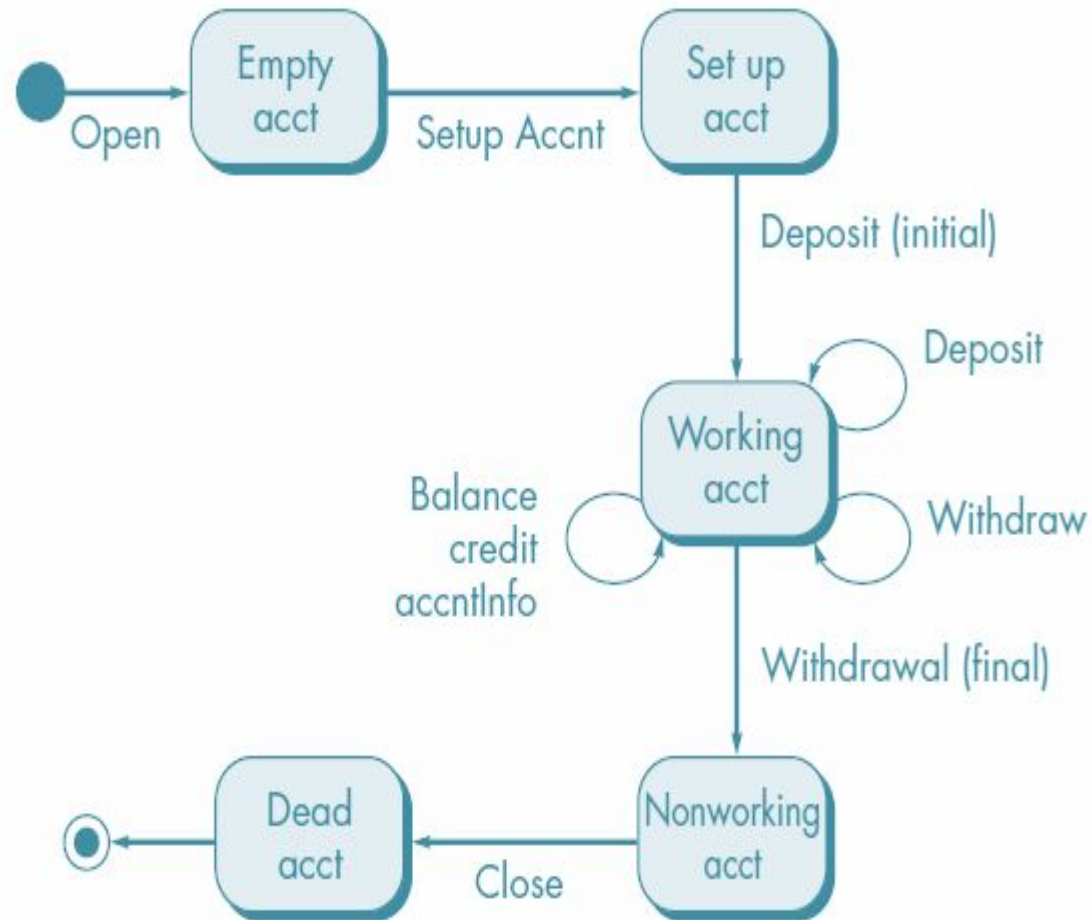
if (password input = 4 digits) then *compare* to stored password

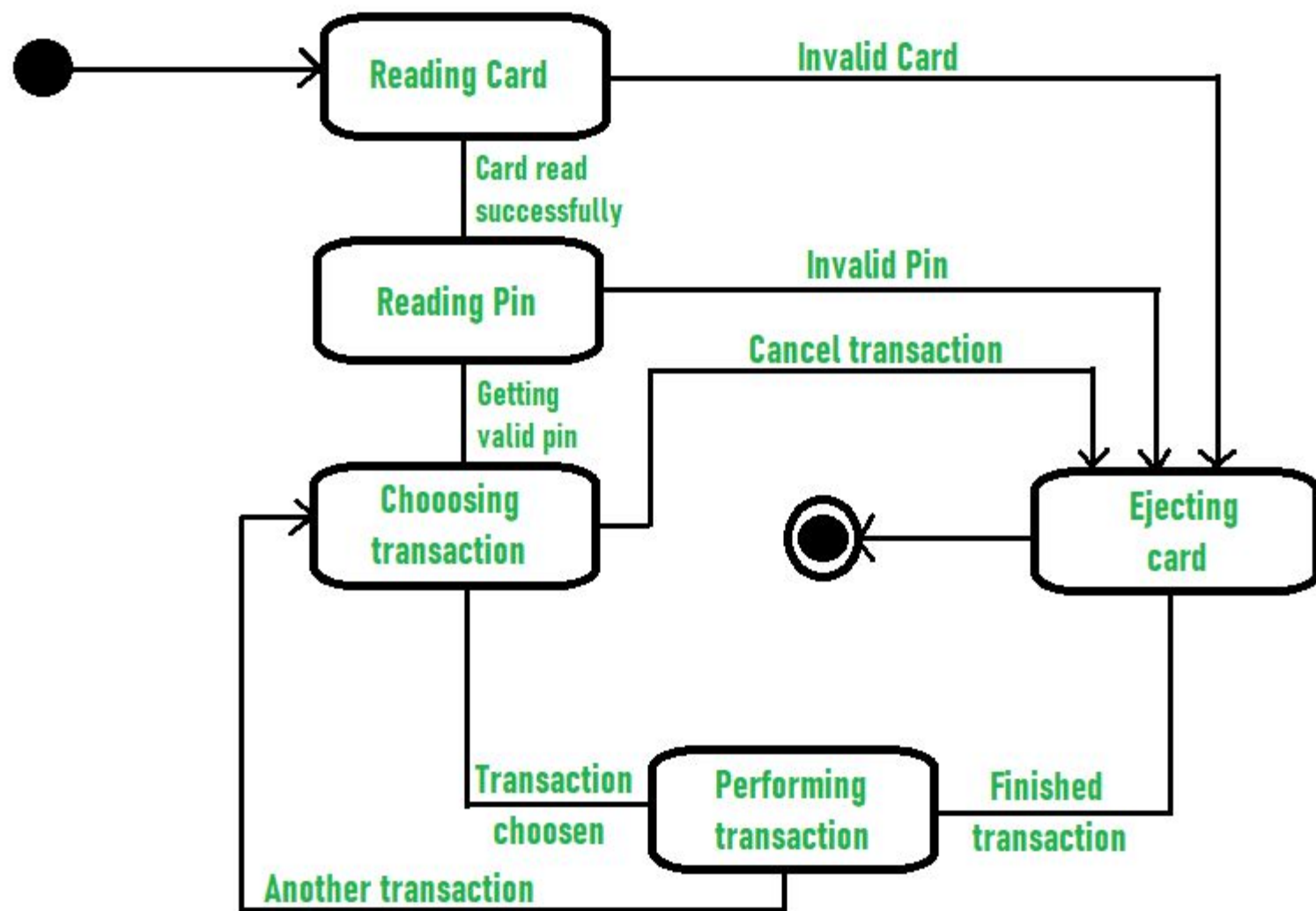
Sequence diagram

- indicates how events cause transitions from object to object

FIGURE 11.2Sequence diagram (partial) for the *SafeHome* security function

- Account class has the following operations:
 - `open()`, `setup()`, `deposit()`, `withdraw()`, `balance()`, `summarize()`, `creditLimit()`, and `close()`





State Transition Diagram for ATM System

