

## **Lecture 5**

### **Finite State Machines**

References: H. Gomaa, Chapter 10 - *Software Modeling and Design*, Cambridge University Press, February 2011

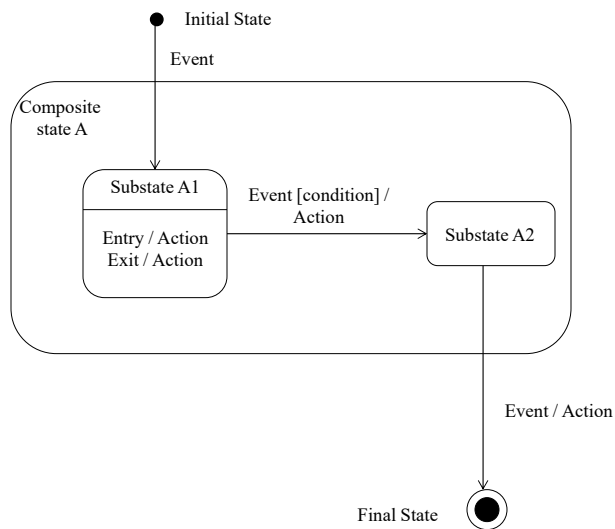
1

### **Finite State Machines and Statecharts**

- Many software systems are state-dependent
  - Action depends not only on input event
  - Also depends on state of system
- Finite State Machine
  - Fixed number of states
  - Only in one state at a time
- Statechart
  - Graphical representation of finite state machine
  - States are rounded boxes
  - Transitions are arcs
- Statechart relates states, events and actions

2

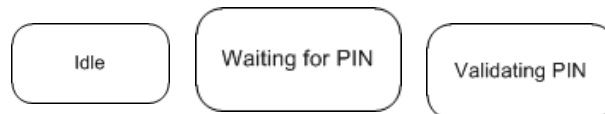
**Figure 2.7 UML notation for statechart:  
composite state with sequential substates**



3

## States

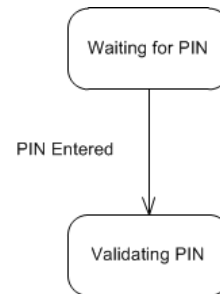
- State
  - A recognizable situation
  - Exists over an interval of time
  - E.g.,



4

## Events

- Event
  - A discrete signal that happens at a point in time
    - E.g., PIN entered
  - Conceptually has no duration
    - Duration very short compared to time in state
  - Causes change of state
    - Referred to as state transition
    - E.g., from
      - Waiting for PIN
      - to
      - Validating PIN

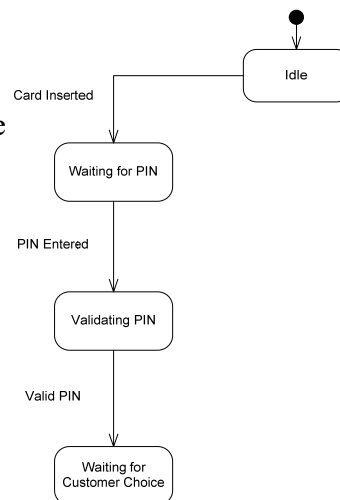


5

**Figure 10.1 Example of  
Events and States  
(ATM Statechart)**

Example of Event sequence

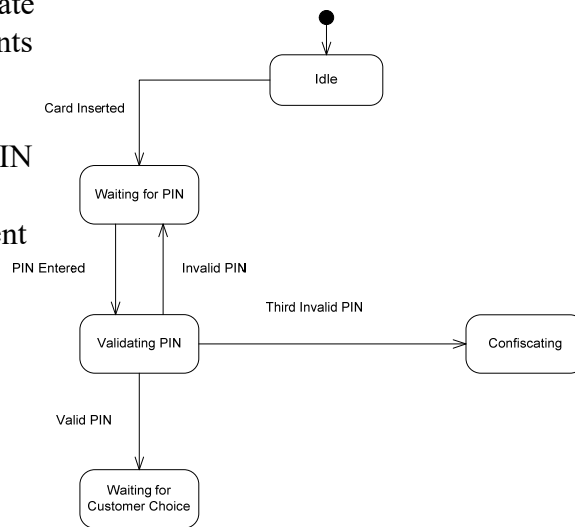
- Card Inserted
- PIN Entered
- Valid PIN



6

**Figure 10.2: Example of Alternative Events**

- Alternative State Transitions from Validating PIN state
  - Three possible events
    - Valid PIN
    - Invalid PIN
    - Third Invalid PIN
  - Each event causes transition to different state



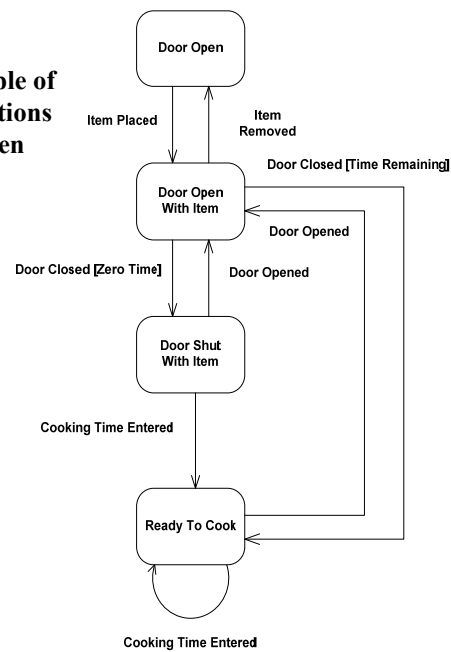
7

## Events and Conditions

- State transition label
  - Event [Condition]
- Condition is a Boolean function
  - Conditions are optional on statecharts
  - Condition is true for finite period of time
- When event arrives, condition must be *true* for state transition to occur.
- If condition is *false*, state transition does not occur

8

**Figure 10.6 Example of Events and Conditions (Microwave Oven statechart)**

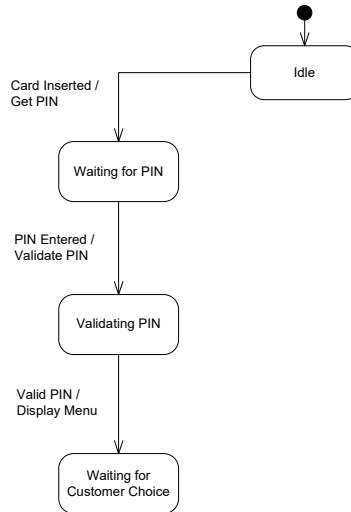


9

## Actions

- State transition label
  - Event / action(s)
  - Event [condition] / action(s)
- Action
  - Executed as a result of state transition
  - Executes instantaneously at state transition
  - Terminates itself
  - Is optional

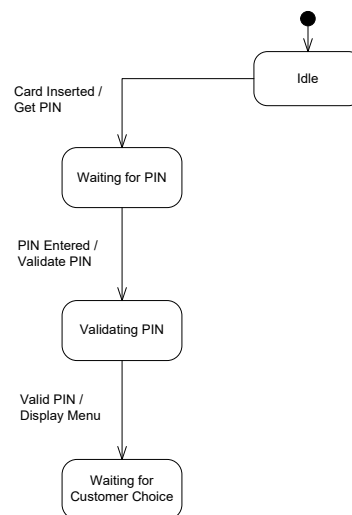
10

**Figure 10.7 Example of actions**

11

## Event and Action - Cause and Effect

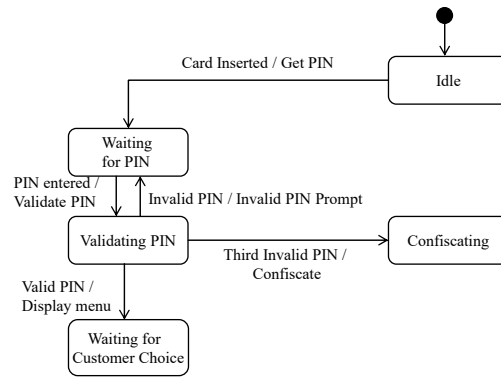
- Event
  - Input to state machine
  - Cause of state transition
- Action
  - Effect (result) of state transition
  - Output from state machine



12

**Figure 10.8 Example of alternative state transitions and actions**

- From Validating PIN
  - Three possible events
    - Valid PIN
    - Invalid PIN
    - Third Invalid PIN
  - Each event causes
    - Transition to different state
    - Different actions

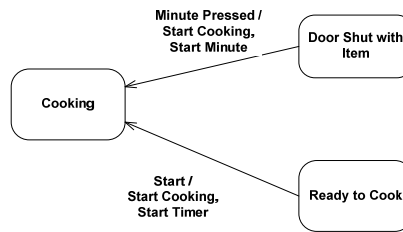
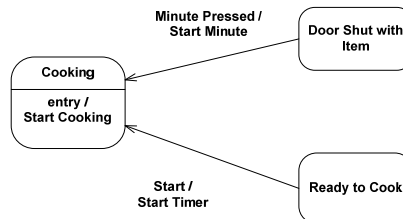


13

## Entry and Exit Actions

- Entry action
  - If same action is executed on every transition into state
  - Use entry action
    - Label inside state box
      - Entry / action
    - Instead of action on every transition into state
    - E.g., Start Cooking

14

**Figure 10.10 Example of entry action****Fig. 10.11a: Actions on state transitions****Fig. 10.11b: Entry action**

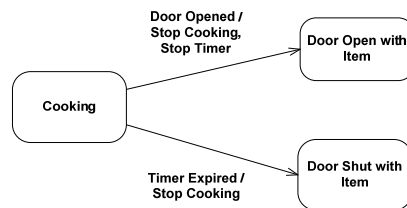
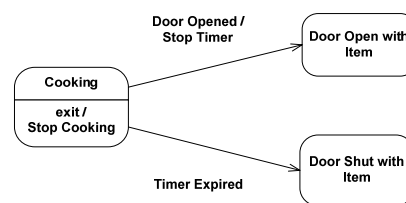
15

## Entry and Exit Actions

- Exit action
  - If same action is executed on every transition out of state
  - Use exit action
    - Label inside state box
      - Exit / action
    - Instead of action on every transition out of state
    - E.g., Stop Cooking

16



**Figure 10.11 Example of exit action****Fig. 10.11a Actions on state transitions****Fig. 10.11b: Exit action**

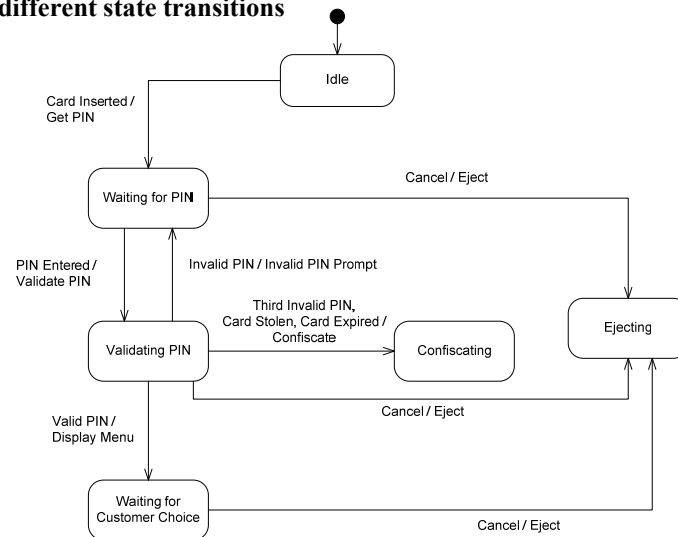
17

## Hierarchical Statecharts

- Disadvantages of flat statecharts
  - Complex state machines get very cluttered
  - Limited capability for managing complexity
- Hierarchical statecharts
  - Based on Harel Statecharts
  - Notation for hierarchical decomposition of state machines
    - Composite state decomposed into substates
    - Default entry states
    - Transition out of composite state corresponds to transition out of every substate

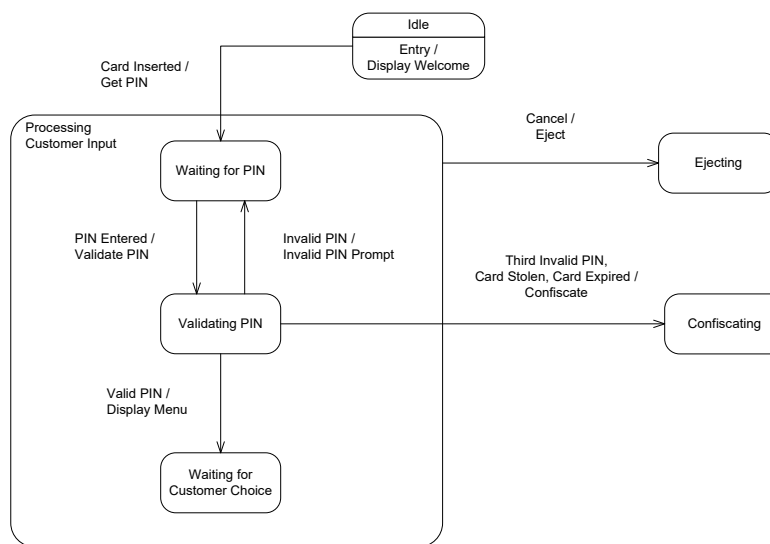
18

**Figure 10.9 Example of same event and action on different state transitions**



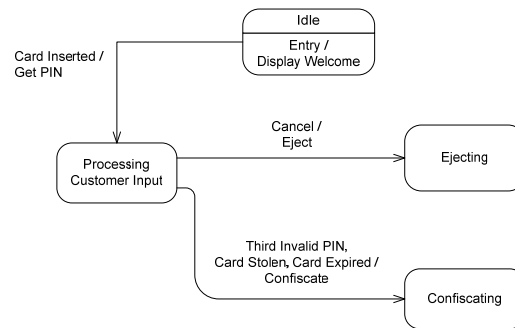
19

**Figure 10.12 Example of hierarchical statechart showing composite state and substates**



20

**Figure 10.12 Example of hierarchical statechart showing composite state without substates**



21

## Hierarchical Statecharts

- **Sequential** decomposition
  - When object is in composite state
    - It is in one and only one of substates
  - Transition into composite state
    - Must be to one and only one of substates
- **Aggregation** of state transitions
  - If same event causes transition out of every substate
  - Then aggregate into transition out of composite state

22

### Guidelines on Statecharts

- State name must be passive not active
  - Represents time period when something
    - is happening, e.g., Waiting for PIN
    - Identifiable situation, e.g., Idle, Initial
- State names must be unique
- Must be able to exit from every state
- Flat statechart
  - Statechart is only in one state at a time
- Hierarchical statechart
  - **sequential** decomposition
    - Statechart is only in one substate at a time

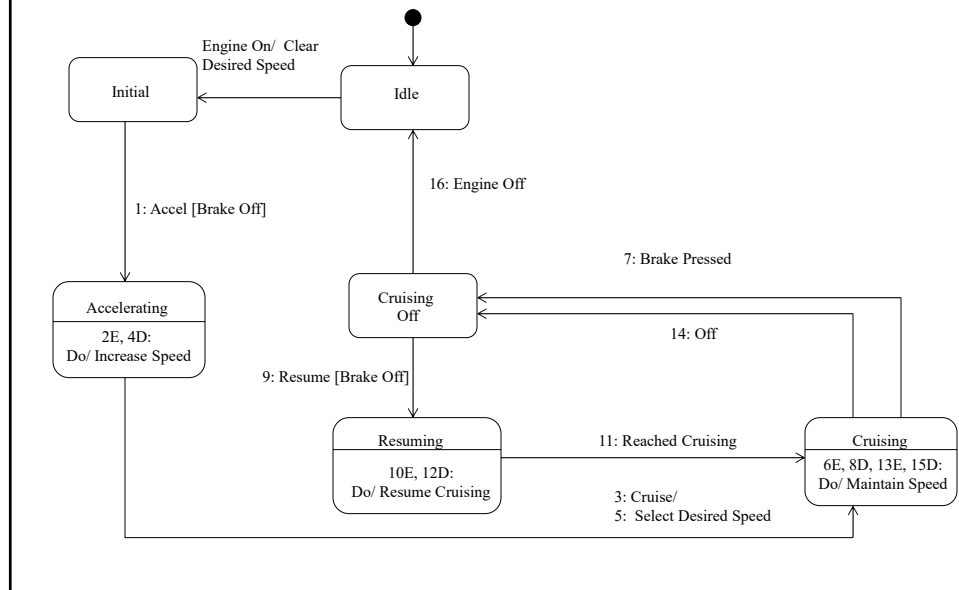
23

### Guidelines on Statecharts

- Event is the cause of the state transition
  - Event happens at a moment in time
  - Event name indicates something has just happened
    - e.g., Card Inserted, Door Closed
- Action is the result of the state transition
  - Action is a command, e.g., Dispense Cash, Start Cooking
  - Action executes instantaneously
  - Activity executes throughout a given state

24

## Statechart from use case with activities



25

## Guidelines on Statecharts

- More than one action possible with a state transition
  - No sequential dependency between actions
- Condition is a Boolean value
  - Event [Condition]
  - State transition only occurs if
    - Event happens & Condition is True
  - Condition is True over some interval of time
- Actions and conditions are optional

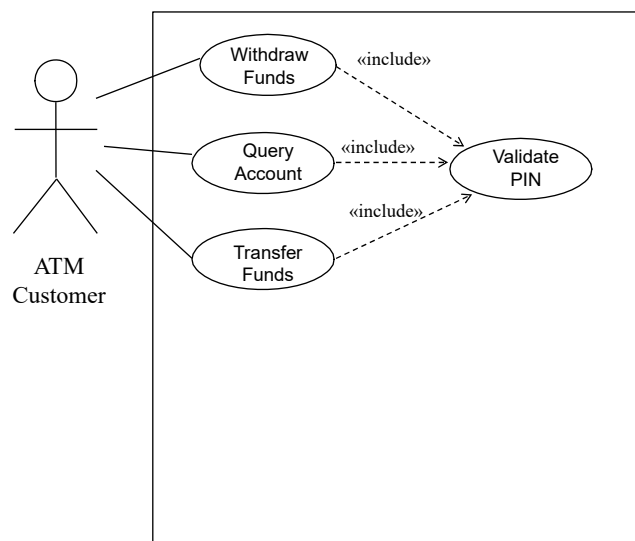
26

## Developing Statechart from Use Case

- Start with Main Sequence through use case
  - Consider sequence of interactions between actor and system
- Consider sequence of external events
  - Input event from external environment
  - Causes state transition to new state
  - Action may result from state transition
- If statechart participates in more than one use case
  - One partial statechart for each use case

27

### Banking System example



28

**Use case name:** Validate PIN.

**Summary:** System validates customer PIN.

**Actor:** ATM Customer.

**Precondition:** ATM is idle, displaying a “Welcome” message.

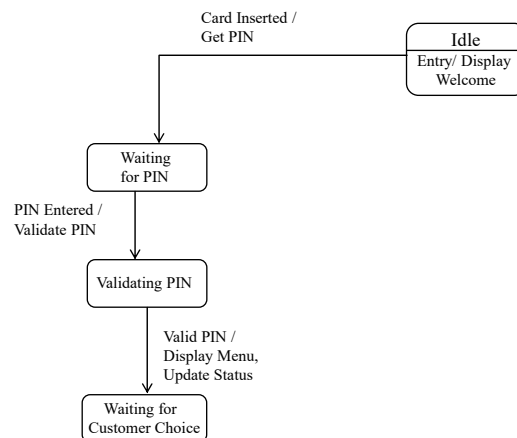
**Description of main sequence:**

1. Customer inserts the ATM card into the card reader.
2. If system recognizes the card, it reads the card number.
3. System prompts customer for PIN.
4. Customer enters PIN.
5. System checks the card’s expiration date and whether the card has been reported as lost or stolen.
6. If card is valid, system then checks whether the user-entered PIN matches the card PIN maintained by the system.
7. If PIN numbers match, system checks what accounts are accessible with the ATM card.
8. System displays customer accounts and prompts customer for transaction type: withdrawal, query, or transfer.

### Example of Inclusion Use Case (Page 374)

29

**Consider Main Sequence of Validate PIN use case**  
**Figure 10.14 Statechart for ATM Control**



30

## Developing Statechart from Use Case (continued)

- Consider alternative sequences of use case
  - Could result in additional states on statechart
  - Could result in additional state transitions
- Initially depict on flat statechart
- Examples
  - Develop alternative sequences on statecharts for
    - Validate PIN
    - Withdraw Funds

31

## Example of Inclusion Use Case (Page 374)

### Alternatives

Step 2: If the system does not recognize the card, the system ejects the card.

Step 5: If the system determines that the card date has expired, the system confiscates the card.

Step 5: If the system determines that the card has been reported lost or stolen, the system confiscates the card.

Step 7: If the customer entered PIN does not match the PIN number for this card, the system reprompts for the PIN.

Step 7: If the customer enters the incorrect PIN three times, the system confiscates the card

Steps 4-8: If the customer enters Cancel, the system cancels the transaction and ejects the card.

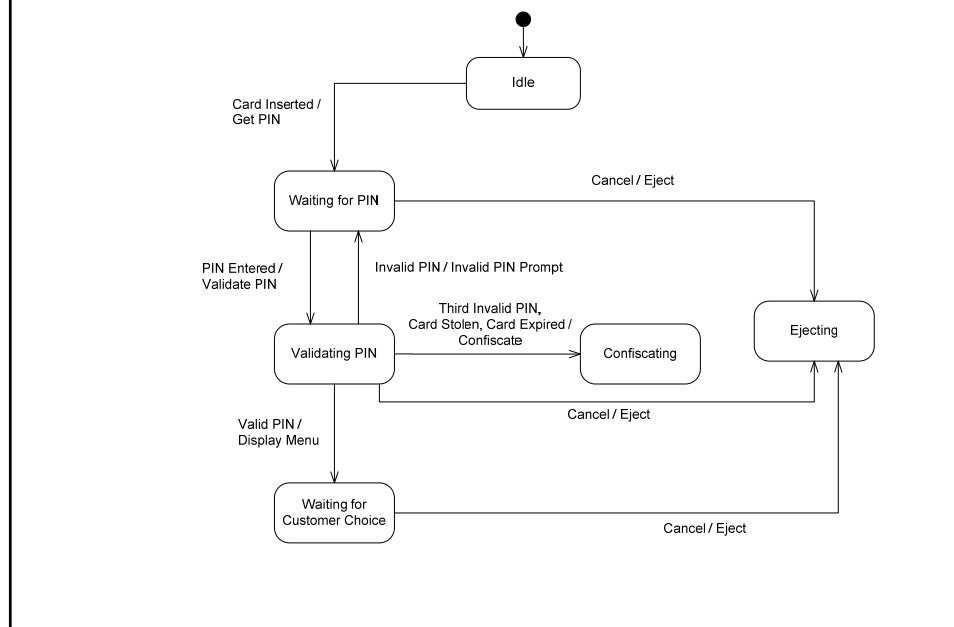
### Nonfunctional requirements:

- a) Security requirement: System shall encrypt ATM card number and PIN.
- b) Performance requirement: System shall respond to actor inputs within 5 seconds.

**Postcondition:** Customer PIN has been validated.

32



**Figure 10.9 Consider Alternative Sequences of Validate PIN use case**

33

### Example of Base Use Case Page 374-375

**Use Case Name:** Withdraw Funds

**Summary:** Customer withdraws a specific amount of funds from a valid bank account.

**Actor:** ATM Customer

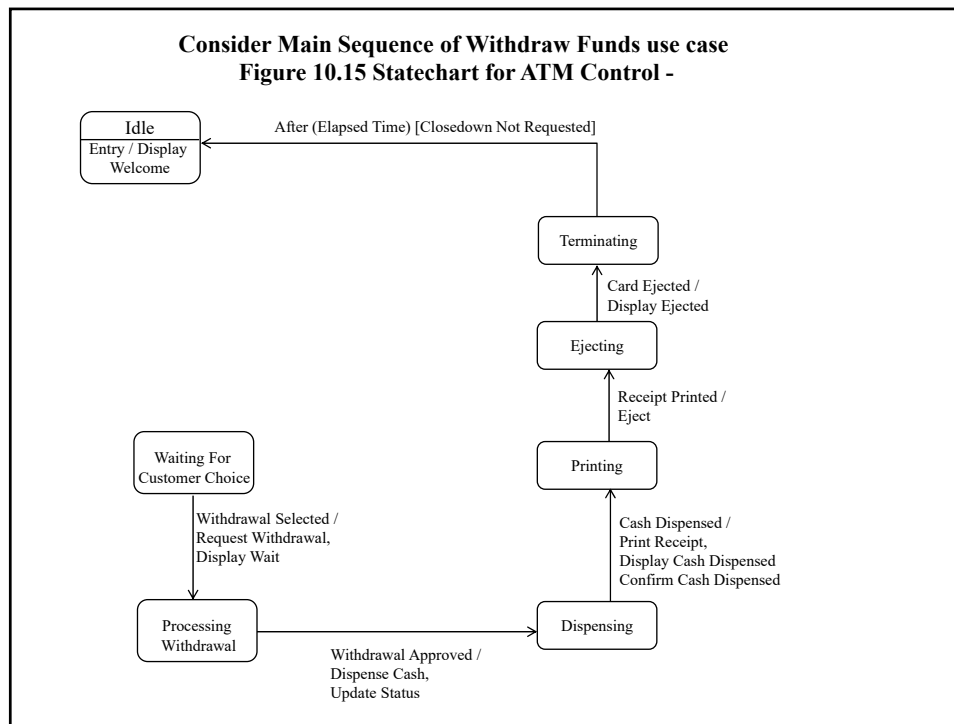
**Dependency:** Include `Validate PIN` use case.

**Precondition:** ATM is idle, displaying a Welcome message.

**Description of main sequence:**

1. Include `Validate PIN` use case.
2. Customer selects Withdrawal, enters the amount, and selects the account number.
3. System checks whether customer has enough funds in the account and whether the daily limit will not be exceeded.
4. If all checks are successful, system authorizes dispensing of cash.
5. System dispenses the cash amount.
6. System prints a receipt showing transaction number, transaction type, amount withdrawn, and account balance.
7. System ejects card.
8. System displays Welcome message.

34



35

### Example of Base Use Case Page 374-375

**Alternatives:**

Step 3: If the system determines the account number is invalid, then it displays an error message and ejects the card.

Step 3: If the system determines there are insufficient funds in the customer's account, then it displays an apology and ejects the card.

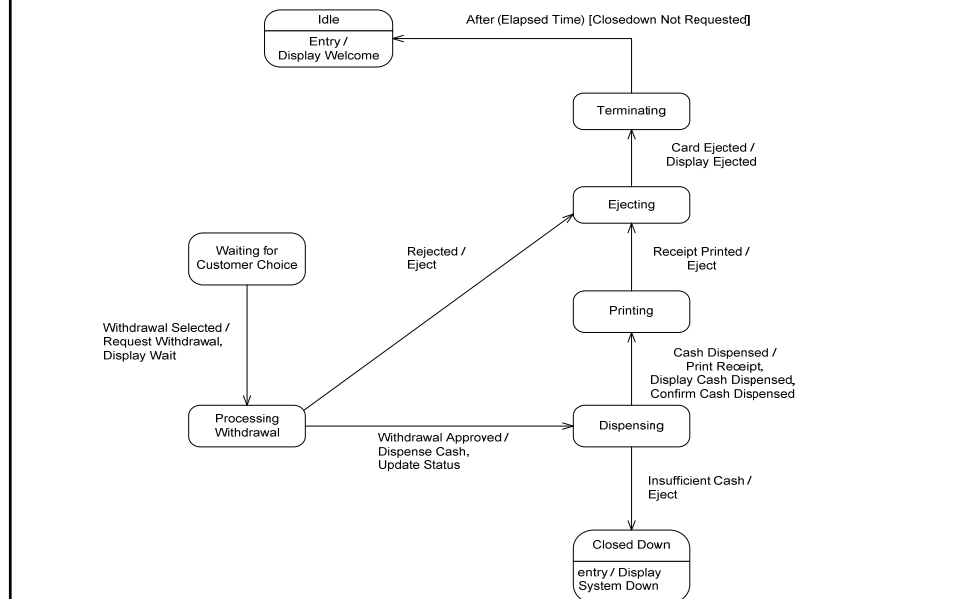
Step 3: If the system determines the maximum allowable daily withdrawal amount has been exceeded, it displays an apology and ejects the card.

Step 4: If the ATM is out of funds, the system displays an apology, ejects the card, and shuts down the ATM.

**Postcondition:** Customer funds have been withdrawn.

36

**Figure 10.16 Statechart for ATM Control  
– Withdraw Funds use case with Alternatives**



37

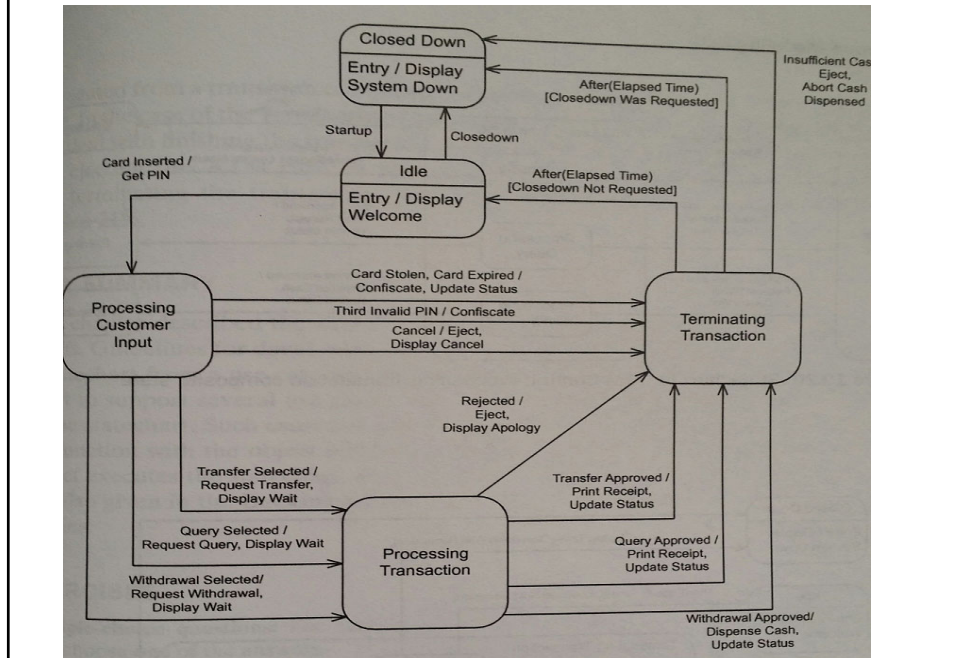
## Developing Statechart from Use Case (continued)

- Develop integrated statechart
  - Integrate partial statecharts to form complete statechart
  - Find common states on partial statecharts
- Example
  - Integrate statecharts for
    - Validate PIN
    - Withdraw Funds

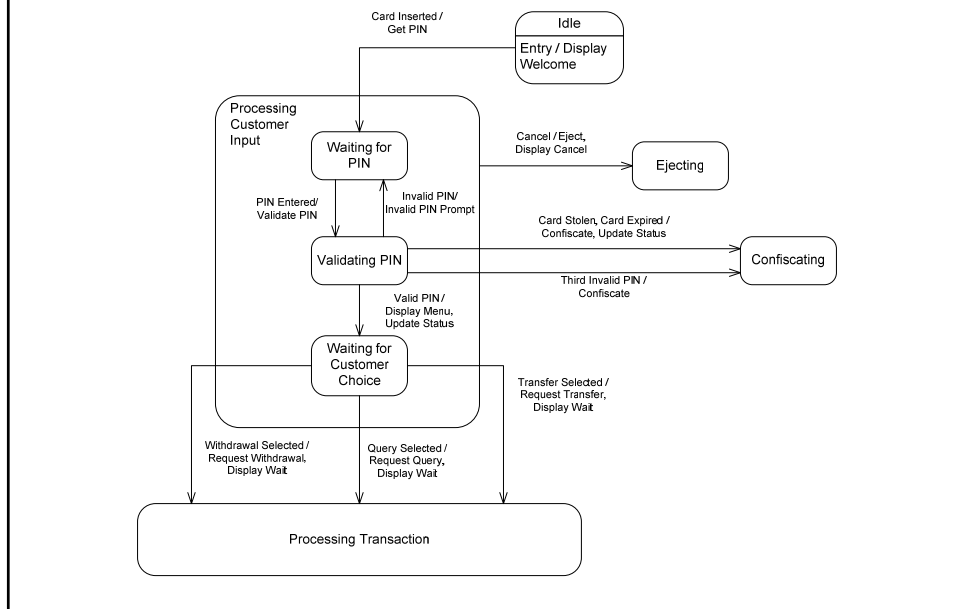
38

40

- Develop hierarchical statechart
  - States aggregated to form composite state
  - Event causing transition from several states
    - Create composite state with one transition out of composite state
    - Instead of many transitions out of substates

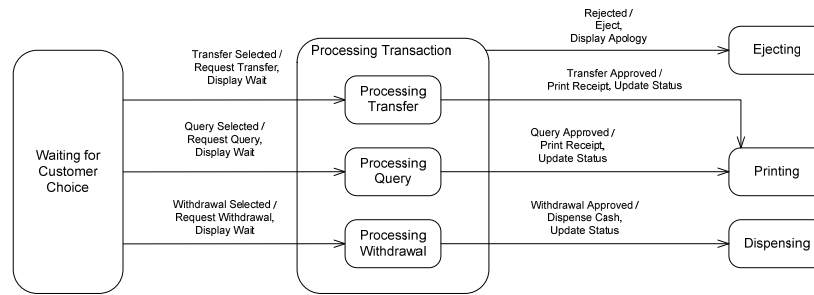
**Figure 10.18 Top level statechart for ATM Control**

41

**Figure 10.19 Statechart for ATM Control - Processing Customer Input composite state**

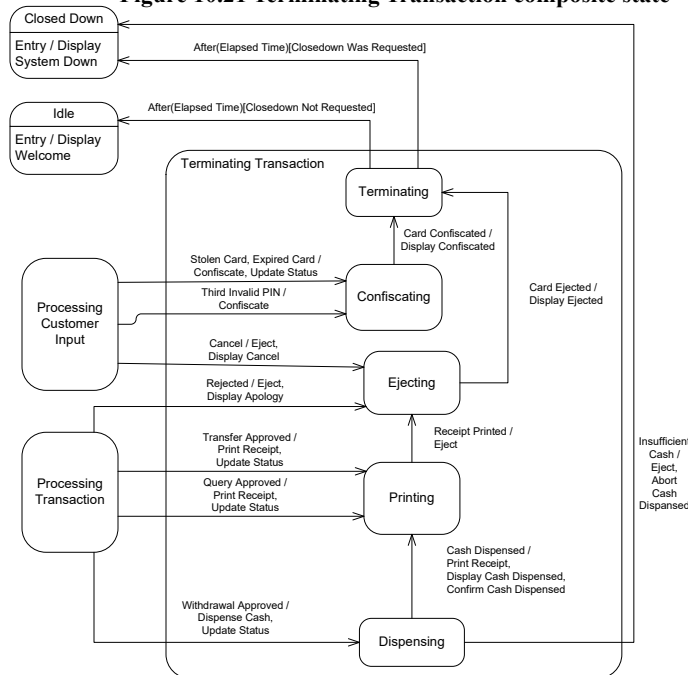
42

**Figure 10.20 Statechart for ATM -  
Processing Transaction composite state**



43

**Figure 10.21 Terminating Transaction composite state**



44