State Diagram

State Diagram

- It describes different states of a system during its lifetime. And these states are changed by events. So State diagrams are useful to model reactive systems. Reactive systems can be defined as a system that responds to external or internal events.
- It describes the flow of control from one state to another state.

State Diagrams

- States are defined as a condition in which an object exists and it changes when some event is triggered.
- So the most important purpose of State diagram is to model life time of an object from creation to termination.
- Used only when the behaviour of an object is complex and more detail is needed.

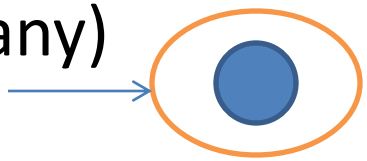
State Diagram Features

- Event – something that happens at a specific point
  - Alarm goes off
- Condition – something that has a duration
  - Alarm is on
  - Fuel level is low
- State – an abstraction of the attributes and relationships of an object (or system)
  - The fuel tank is in a too low level when the fuel level is below level *x* for *n* seconds

The following are the basic notational elements that can be used to make up a diagram:

- Filled circle, pointing to the initial state

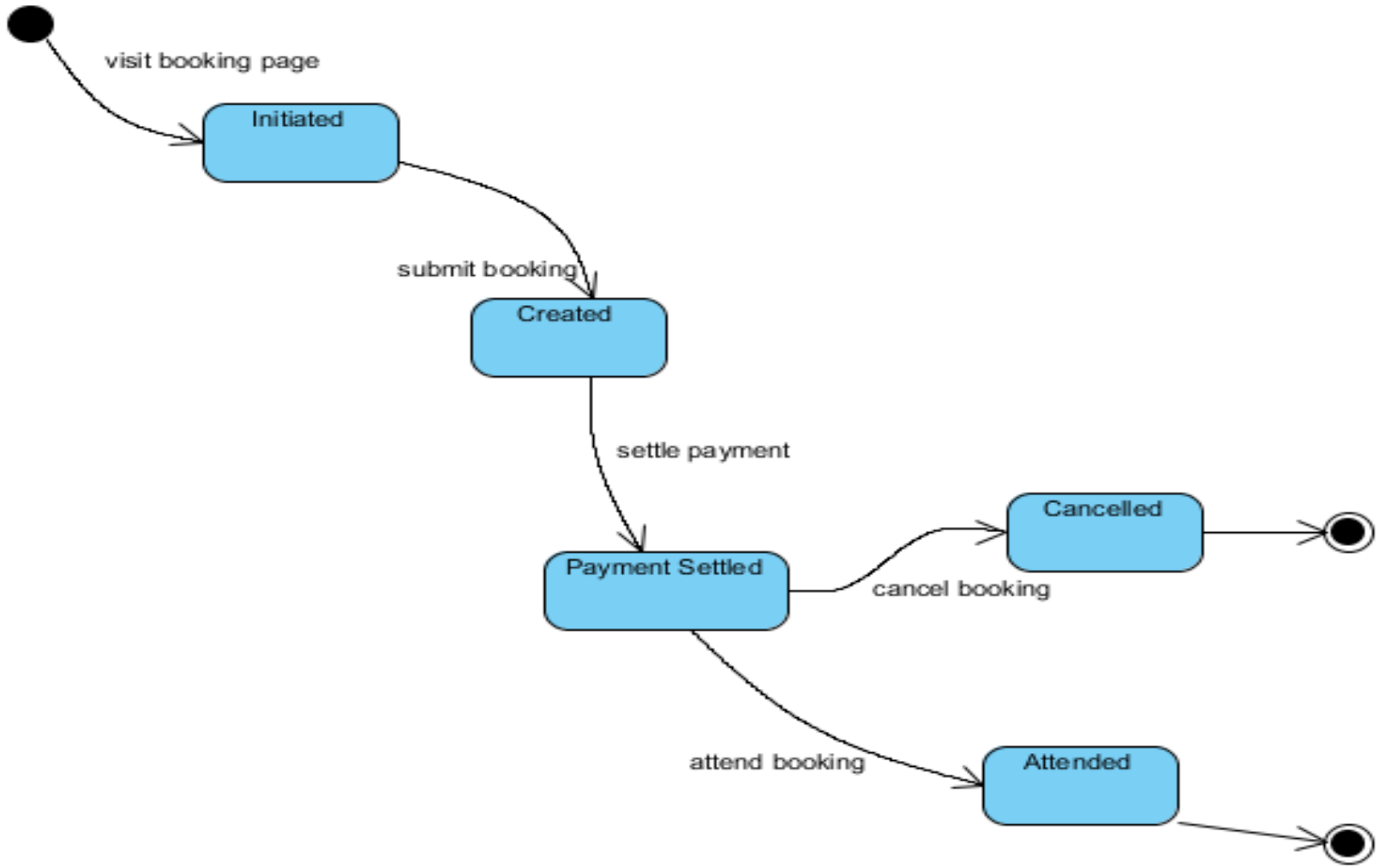
- Hollow circle containing a smaller filled circle, indicating the final state (if any)



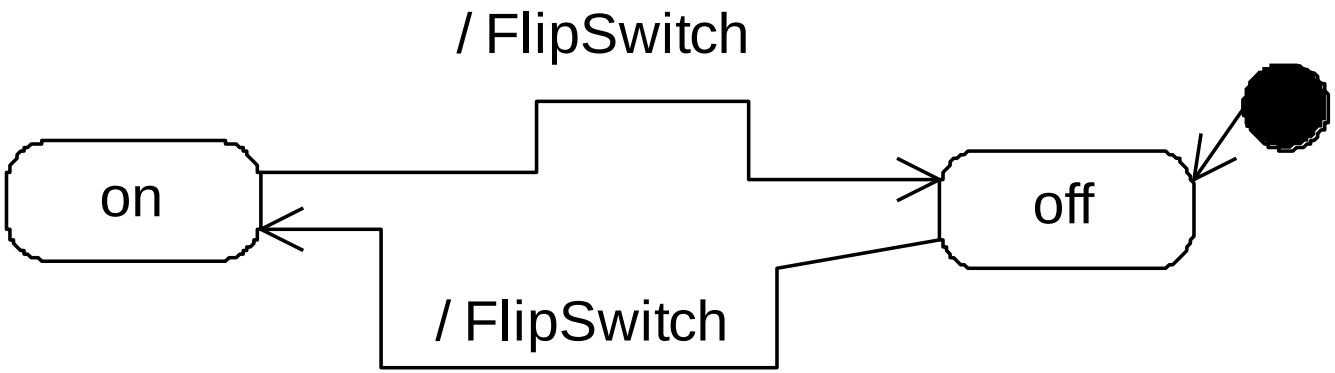
- Rounded rectangle, denoting a state

- Arrow, denoting transition. The name of the event (if any) causing this transition labels the arrow body.

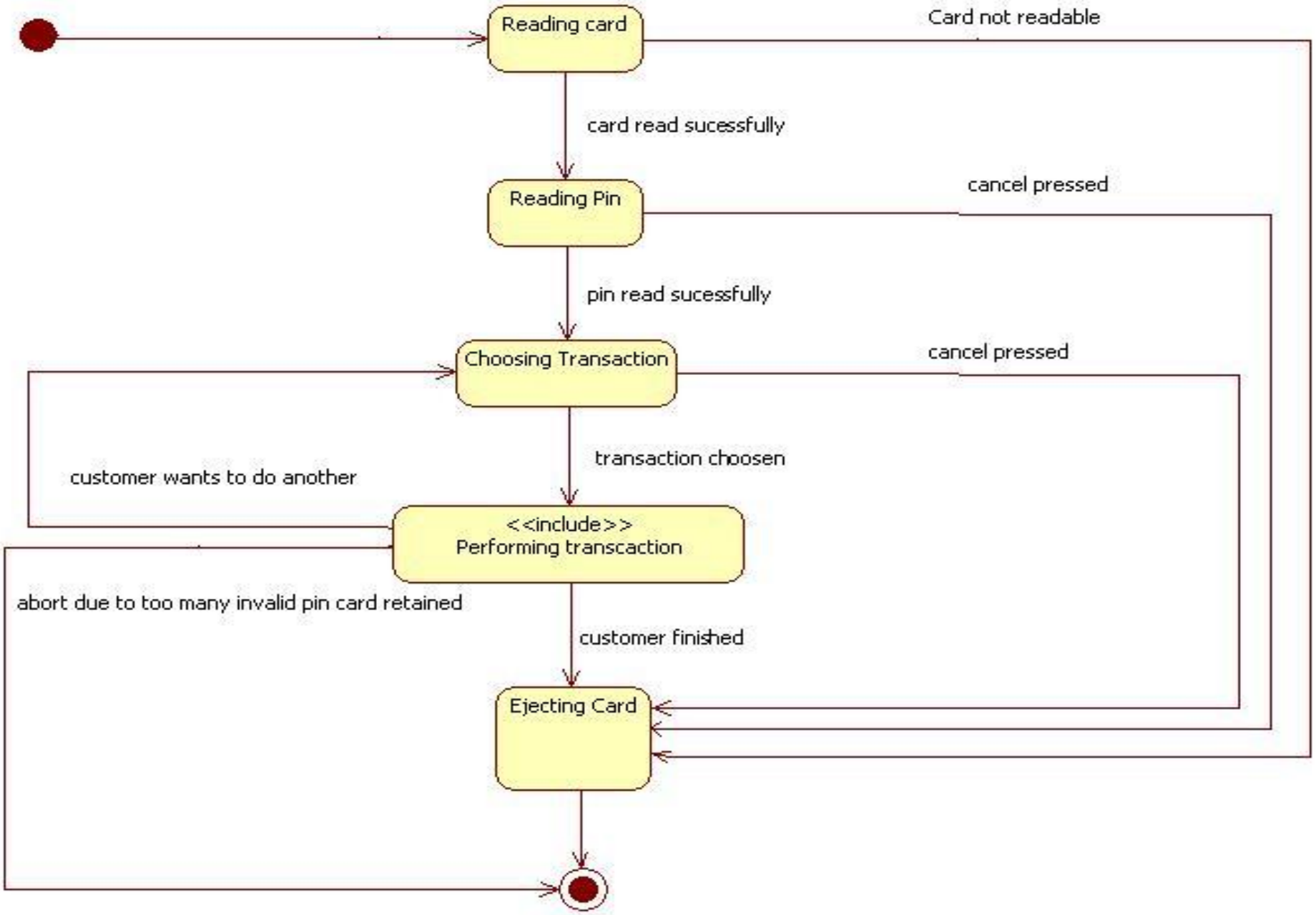
State Diagram for Online Booking of a Show



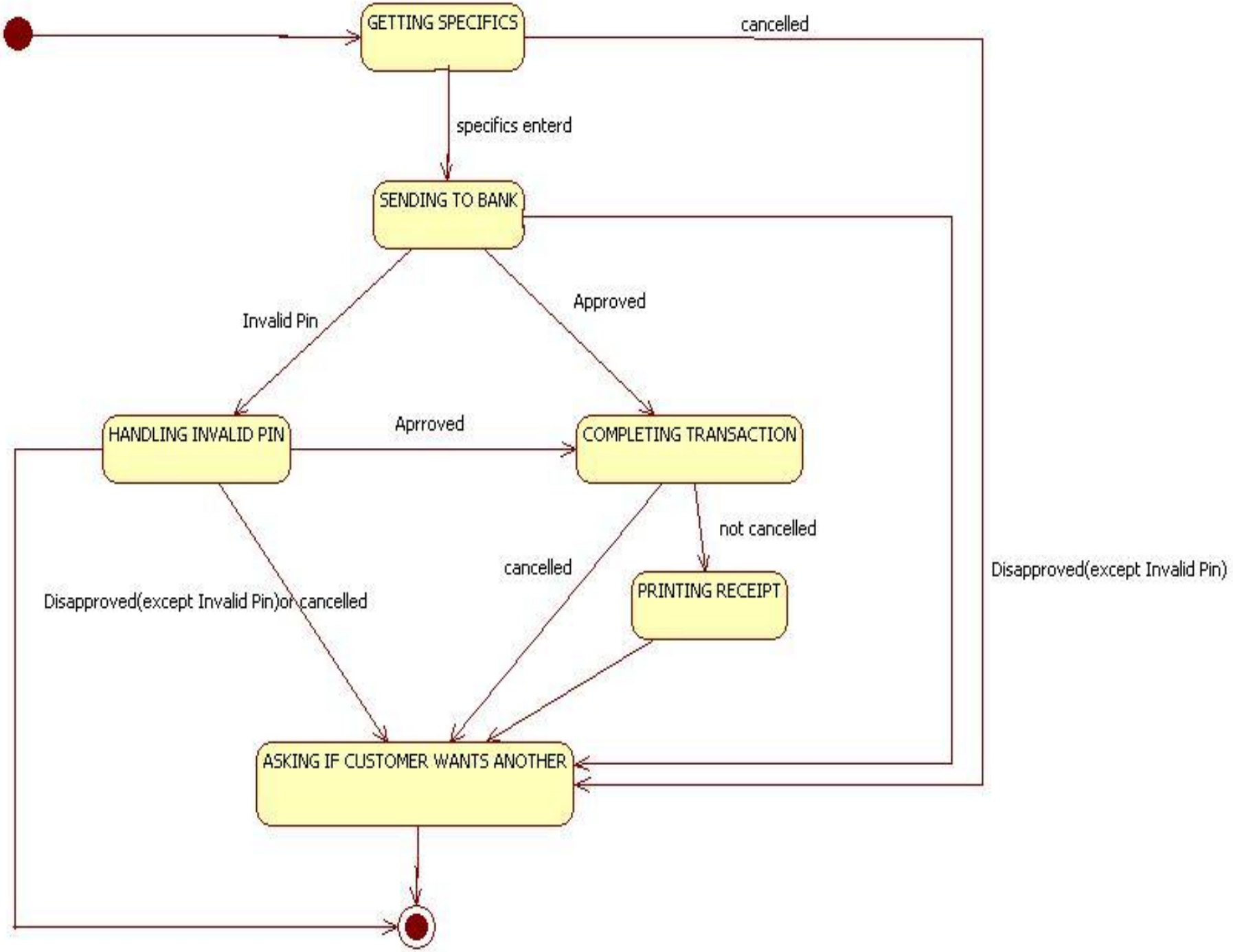
Example: on/off Switch



State Diagram Of ATM 1


Reading card
Card not readable
card read sucessfully
Reading Pin
cancel pressed
pin read sucessfully
Choosing Transaction
cancel pressed
transaction choosen
customer wants to do another
<<include>>
Performing transcaction
abort due to too many invalid pin card retained
customer finished
Ejecting Card

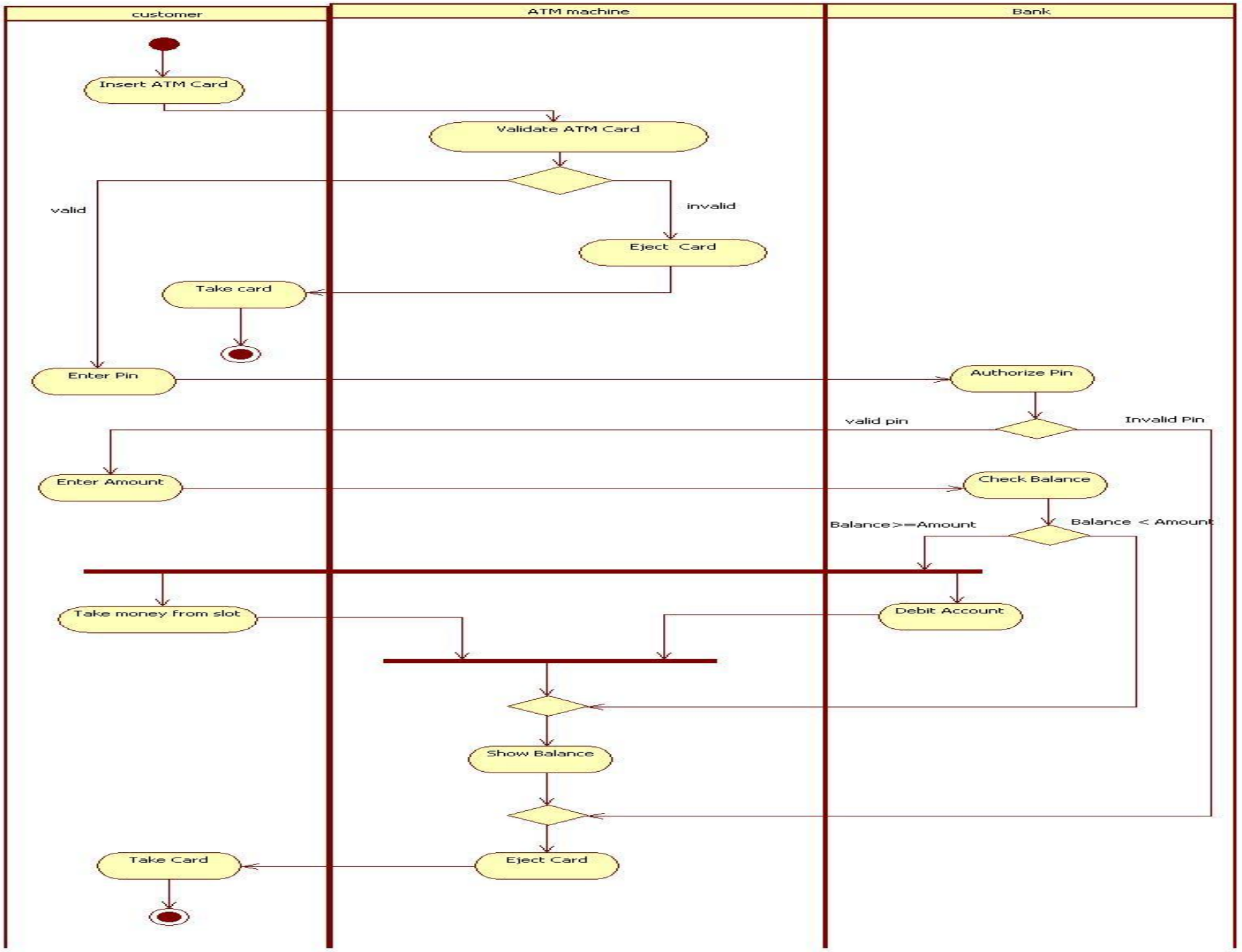
State Diagram Of ATM 2


GETTING SPECIFICS
cancelled
specifics enterd
SENDING TO BANK
Invalid Pin
Approved
HANDLING INVALID PIN
Aprroved
COMPLETING TRANSACTION
not cancelled
cancelled
Disapproved(except Invalid Pin)or cancelled
PRINTING RECEIPT
Disapproved(except Invalid Pin)
ASKING IF CUSTOMER WANTS ANOTHER

Activity diagrams

- are graphical representations of workflows of stepwise activities and actions[1] with support for choice, iteration and concurrency. Activity diagrams show the overall flow of control.
- Activity diagrams are constructed from a limited number of shapes, connected with arrows.[4] The most important shape types:
- *rounded rectangles* represent *actions*;
- *diamonds* represent *decisions*;
- *bars* represent the start (*split*) or end (*join*) of concurrent activities;
- a *black circle* represents the start (*initial state*) of the workflow;
- an *encircled black circle* represents the end (*final state*).
- Arrows run from the start towards the end and represent the order in which activities happen.
- Hence they can be regarded as a form of flowchart. Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops

Example of ATM(Withdraw)


customer
ATM machine
Bank
Insert ATM Card
Validate ATM Card
valid
invalid
Eject Card
Take card
Enter Pin
Authorize Pin
valid pin
Invalid Pin
Enter Amount
Check Balance
Balance>=Amount
Balance < Amount
Take money from slot
Debit Account
Show Balance
Take Card
Eject Card

Sequence diagrams

Sequence diagrams

- Represent sequence of interactions among objects with messages between them for different scenarios of a system
- Made up of objects, links, and messages
- Models flow of control of a system by time ordering

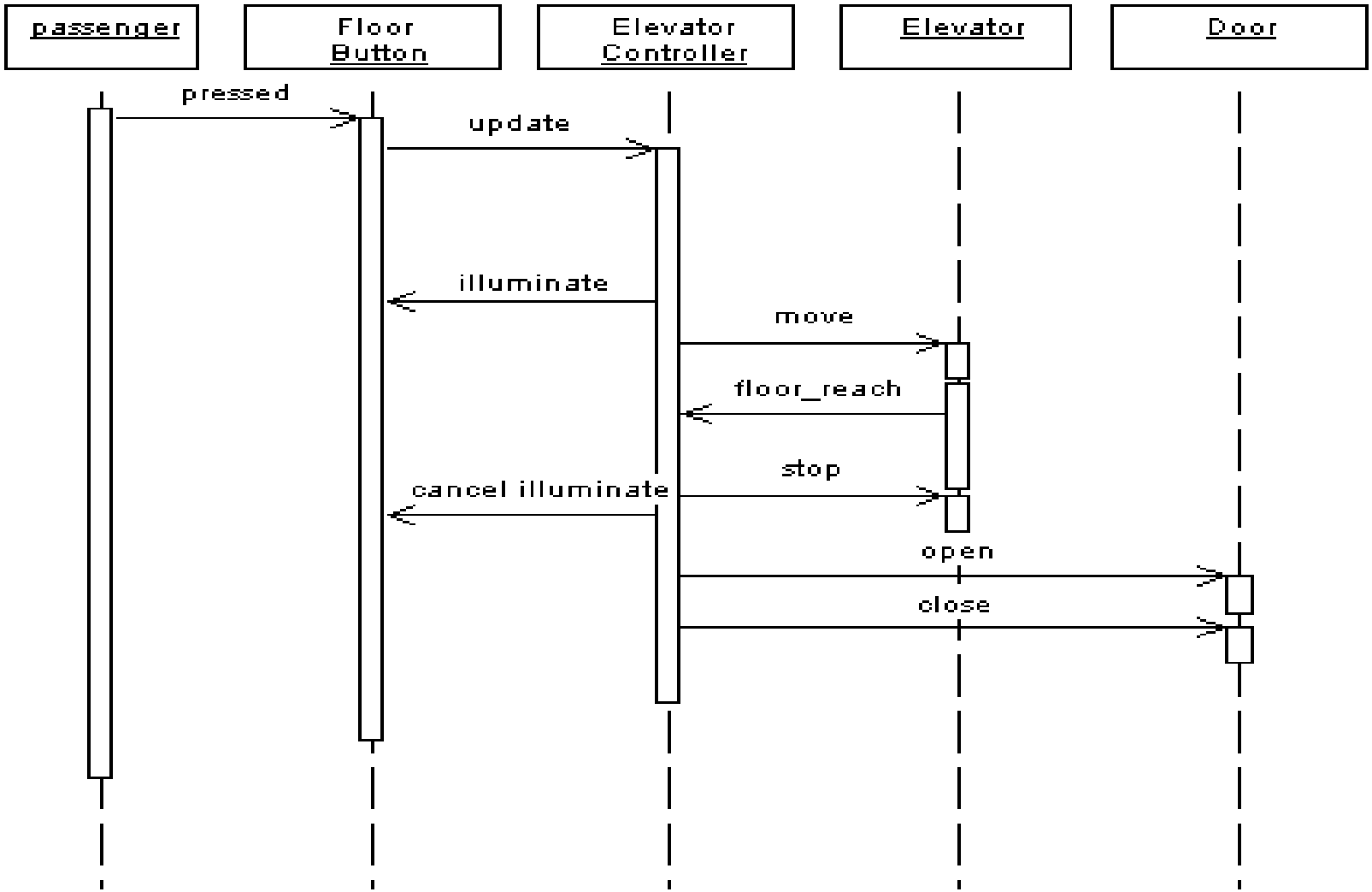
  Emphasizes passing messages with respect to time
- It is a kind of interaction diagram that shows how objects operate with one another, as well as with external entities and in what order

Sequence Diagram(cont..)

- It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.
- A sequence diagram shows, as parallel vertical lines (*lifelines*) - different processes or objects that live simultaneously,

  and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.

Example: Sequence Diagram of an Elevator Control System



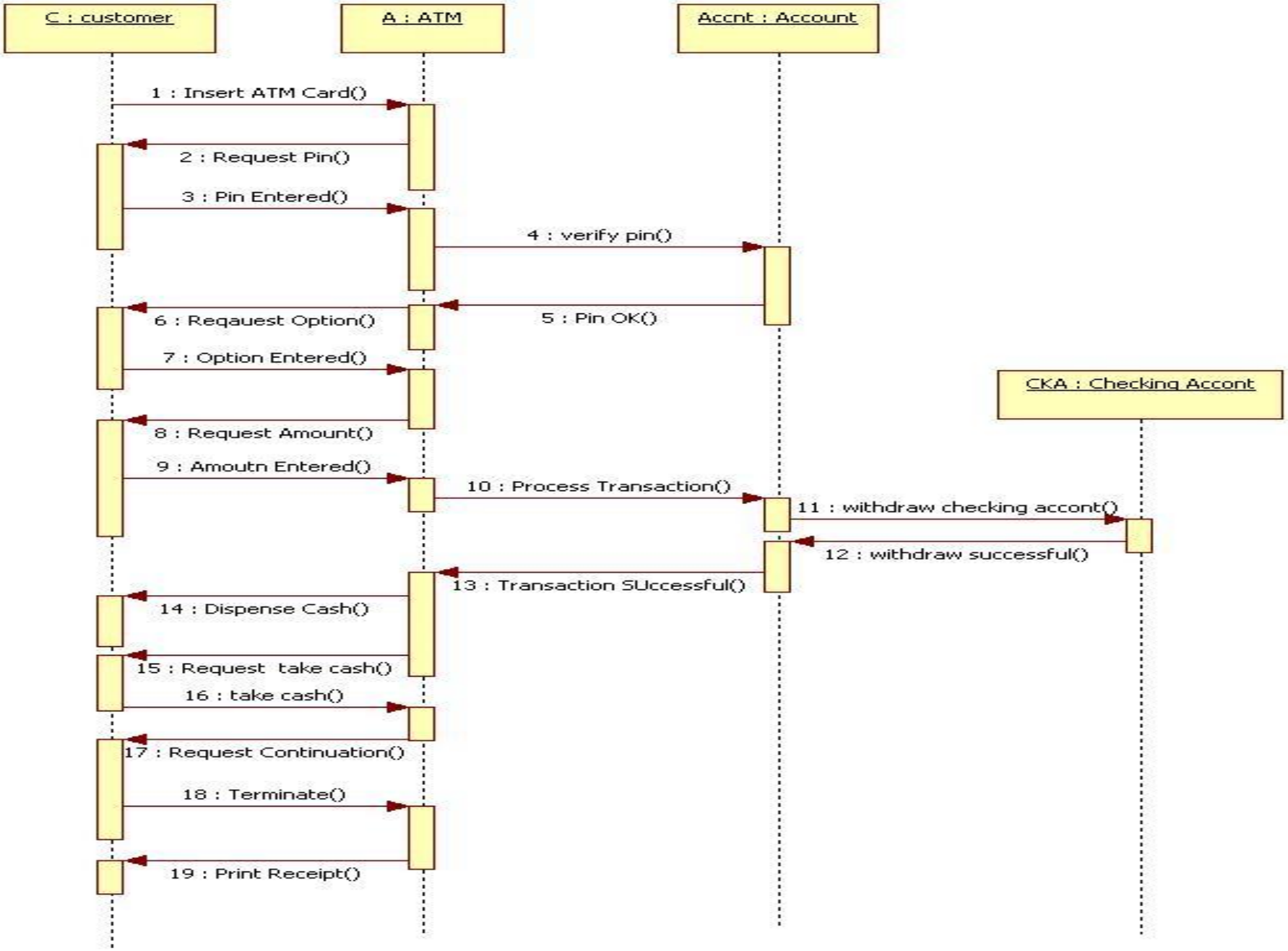
Sequence Diagrams(cont..)

- X-axis is objects
  - Object that initiates interaction is left most
  - Object to the right are increasingly more subordinate
- Y-axis is time
  - Messages sent and received are ordered by time
- Object life lines represent the existence over a period of time
- Activation (double line) is the execution of the procedure.

Properties of Sequence Diagrams

- Initiator is leftmost object (boundary object)
- Next is typically a control object
- Then comes entity objects

Sequence Diagram of ATM(Withdraw)


C : customer
A : ATM
Accnt : Account
CKA : Checking Accont
1 : Insert ATM Card()
2 : Request Pin()
3 : Pin Entered()
4 : verify pin()
5 : Pin OK()
6 : Reqauest Option()
7 : Option Entered()
8 : Request Amount()
9 : Amoutn Entered()
10 : Process Transaction()
11 : withdraw checking accont()
12 : withdraw successful()
13 : Transaction SUccessful()
14 : Dispense Cash()
15 : Request take cash()
16 : take cash()
17 : Request Continuation()
18 : Terminate()
19 : Print Receipt()