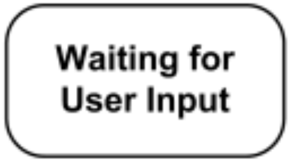
What is State Transition Diagram?

State Transition Diagram

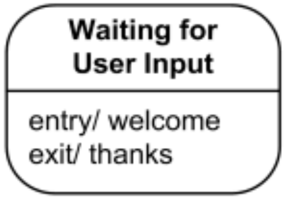
- Models the dynamic flow of control from state to state of a particular entity within a system.
- Shows the behaviour of classes in response to external stimuli.
  - It can be the behaviour of an object, component, agent, or the entire system
  - Describes the behaviour of a single object in response to a series of events in a system. Sometimes it's also known as a Harel state-chart or a state machine.

Symbols and Notations

- States:
  - Represent situations during the life of an object.
  - You can use a simple rectangle. Some tools provide a state in the following formats:



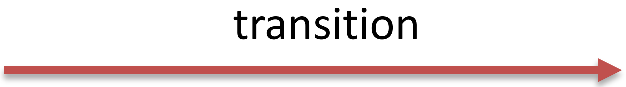
Simple state



Simple that also shows actions and guards

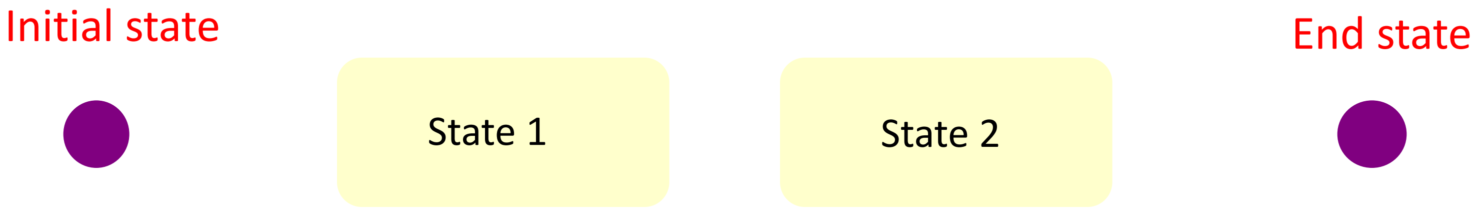
Transition

- A solid arrow represents the path between different states of an object.
- **Must** be labeled with an event/action that triggered.



Special States

- The initial state is the state entered when an object is created.
  - An initial state is mandatory.
  - Only one initial state is permitted.
  - The initial state is represented as a solid circle.
- A final state indicates the end of life for an object.
  - A final state is optional.
  - More than one final state may exist.
  - A final state is indicated by a bull's eye.



Initial state

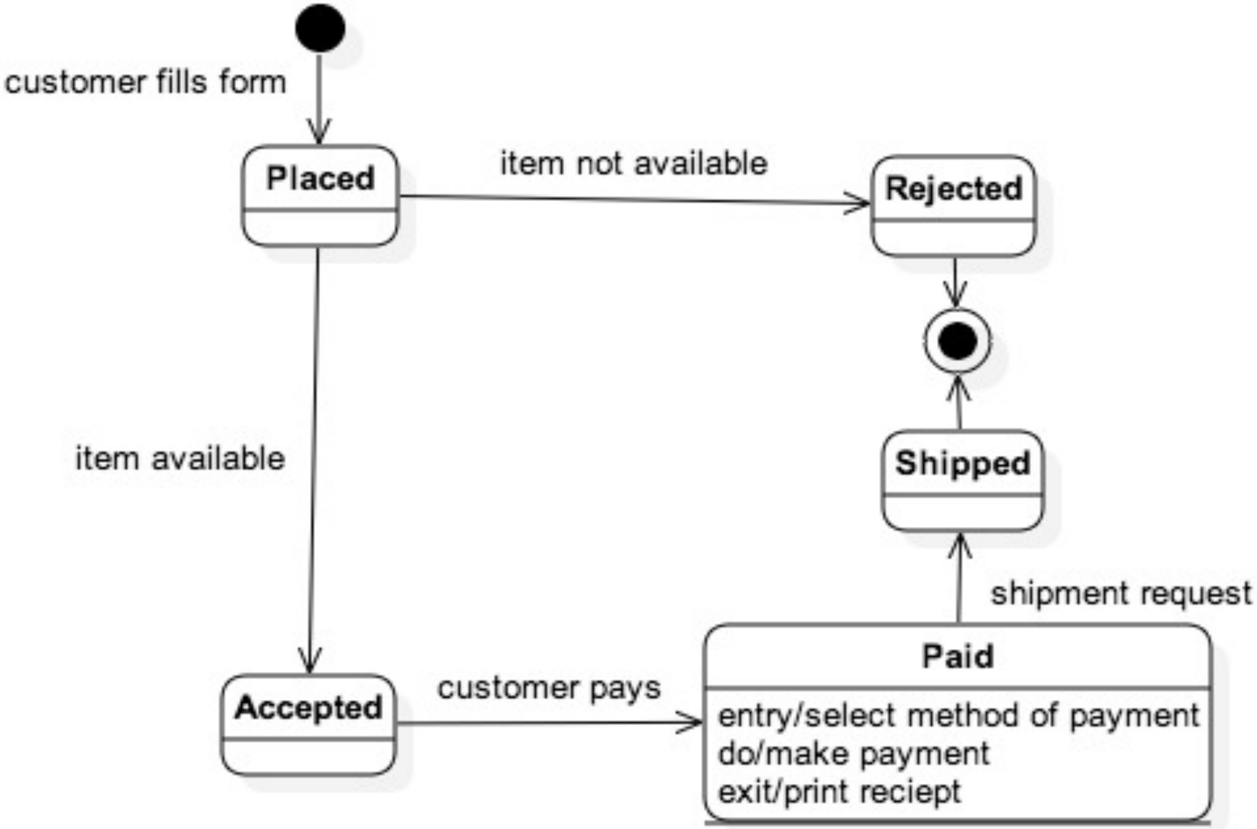
State 1

State 2

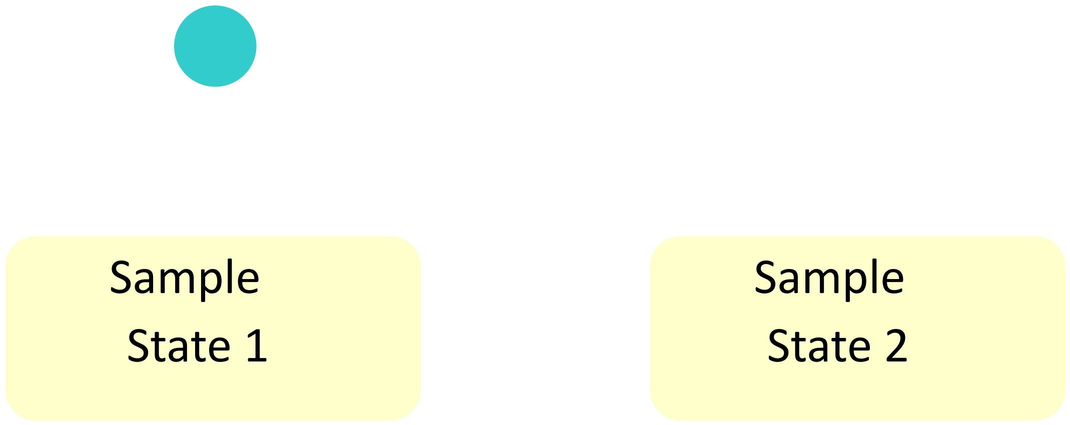
End state

Example

- A state transition diagram for an online shopping module/component:



- Can have transitions back to itself

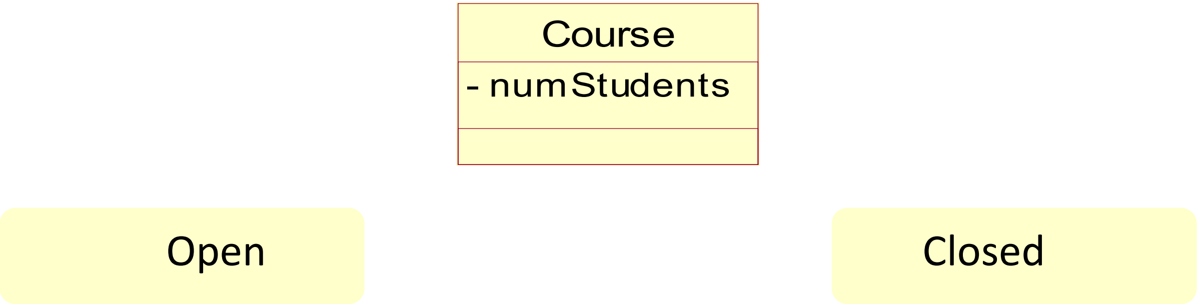


Where to Start to Determine States

- Initially, concentrate on the behavior of the modules of classes, with significant dynamic behavior.
- For a given module or class, look for possible states by
  - Evaluating attribute values.
  - Evaluating operations.
  - Defining the rules for each state.
  - Identifying the valid transactions between states.
- Examine message trace or interaction diagrams that involve the class being modeled.
  - The interval between two operations may be a state.

State and Attributes

- States may be distinguished by the values of certain attributes.
  - Example: If the number of student is less than a limit course is canceled.
- States may be also distinguished “cardinality”. Example professor **may** teach a course.
  - Teaching when a link to a course exists.
  - On sabbatical when no link exists.

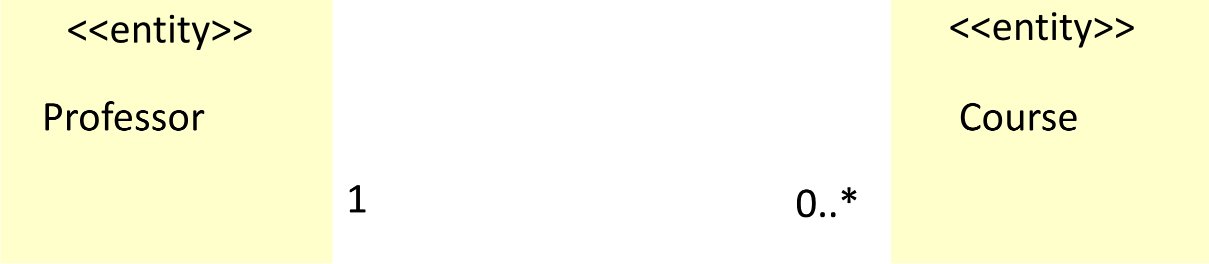


Course

- numStudents

Open

Closed



<<entity>>

Professor

1

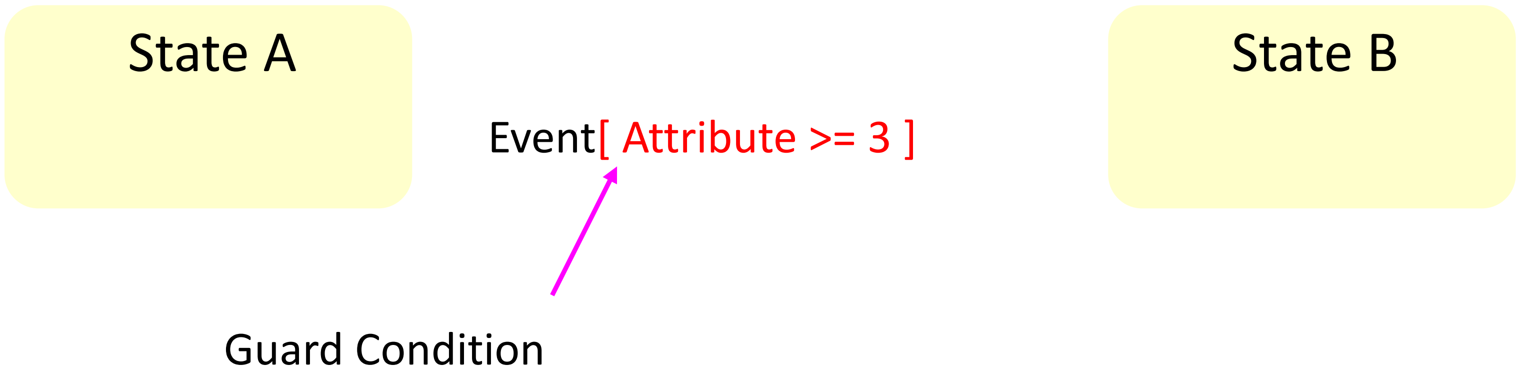
0..*

<<entity>>

Course

What Are Events and Guard Conditions?

- A Boolean expression that is evaluated when the transition is triggered by the reception of the event trigger. If the expression evaluates True, the transition is eligible to fire. If the expression evaluates to False, the transition does not fire.



State A

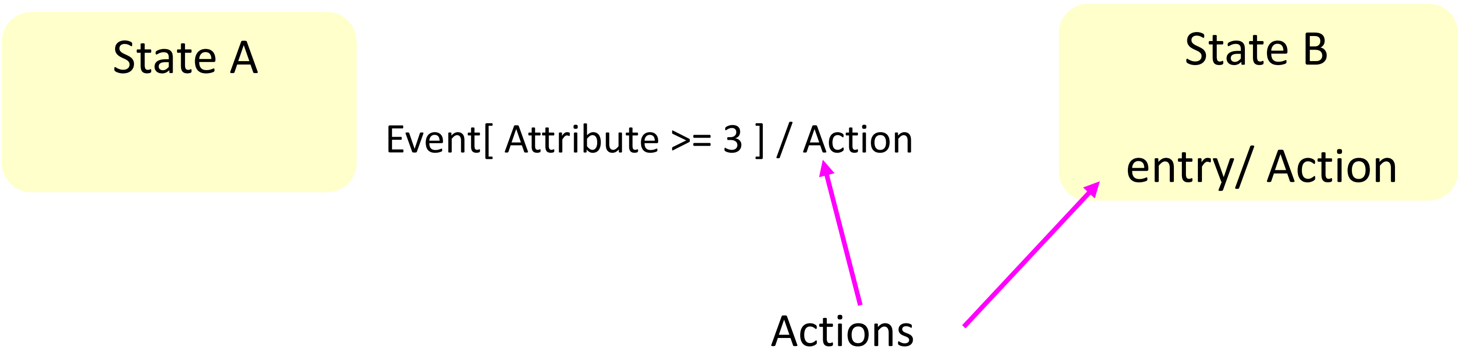
State B

Event[Attribute >= 3]

Guard Condition

What Are Actions?

- Executable atomic computations that results in a change in state of the model or the return of a value
  - Associated with a transition
  - Take an insignificant amount of time to complete
  - Non-interruptible



State A

Event[Attribute >= 3] / Action

State B

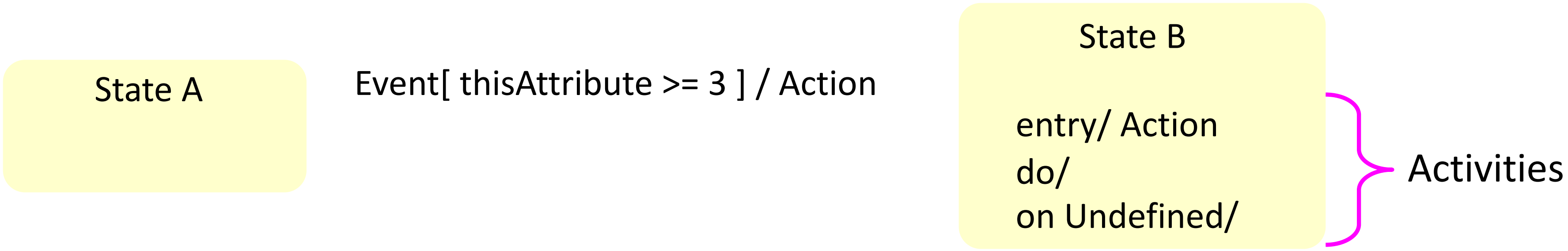
entry/ Action

Actions

Ongoing Activities within State?

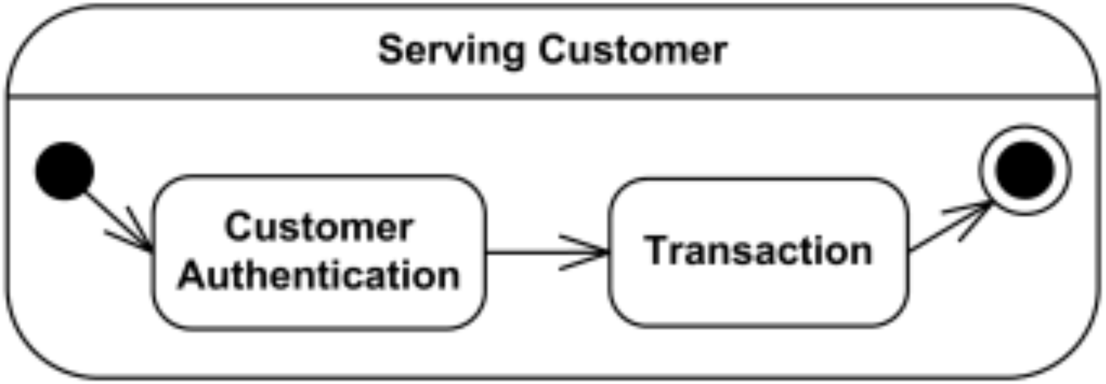
Activities are ongoing and non-atomic executions within a state machine

- An operation that takes time to complete
- Starts when the state is entered
- Can run to completion or can be interrupted by an outgoing transition



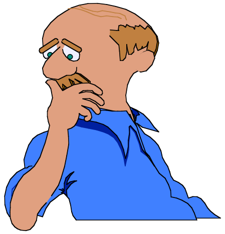
Composite States

- Generally, **composite state** is defined as state that has sub-states (nested states).

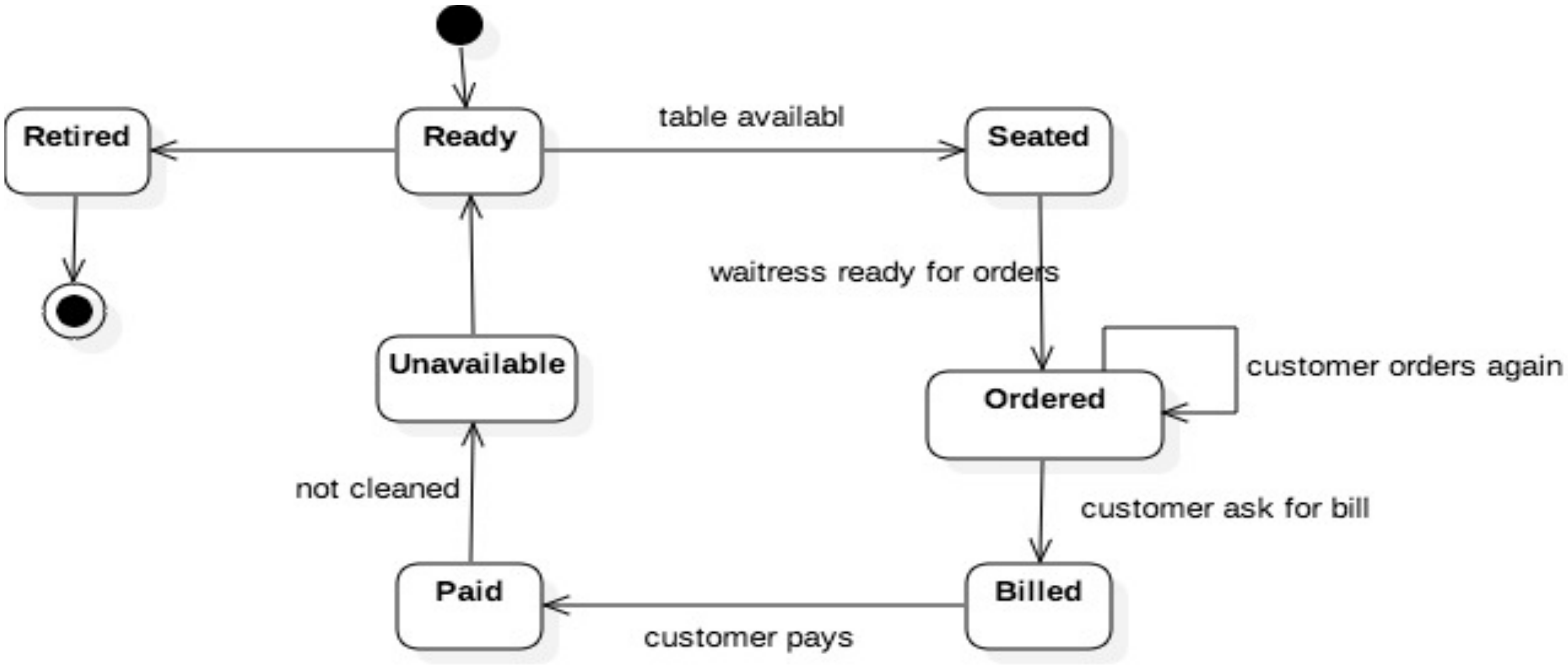


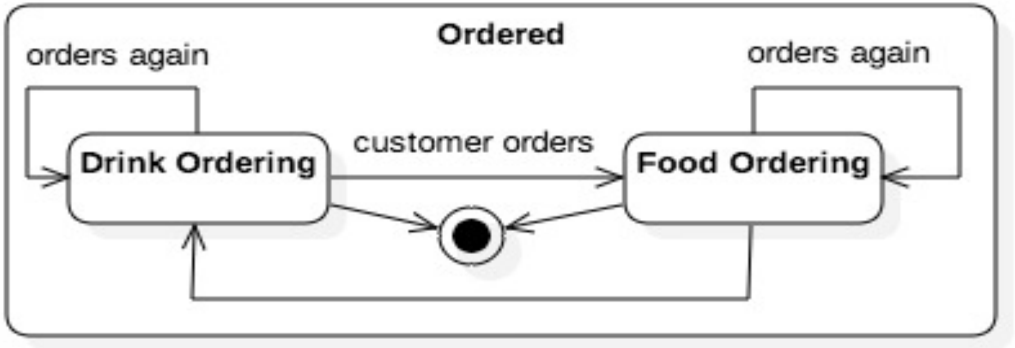
Class Discussion – Problem I

- Let's assume:
  - you are working as a software engineer to design an “Automated family restaurant” application, and your task is to build a STD model for a class called “Table”.



Solution


Retired
Ready
table availabl
Seated
waitress ready for orders
Unavailable
Ordered
customer orders again
not cleaned
customer ask for bill
Paid
Billed
customer pays

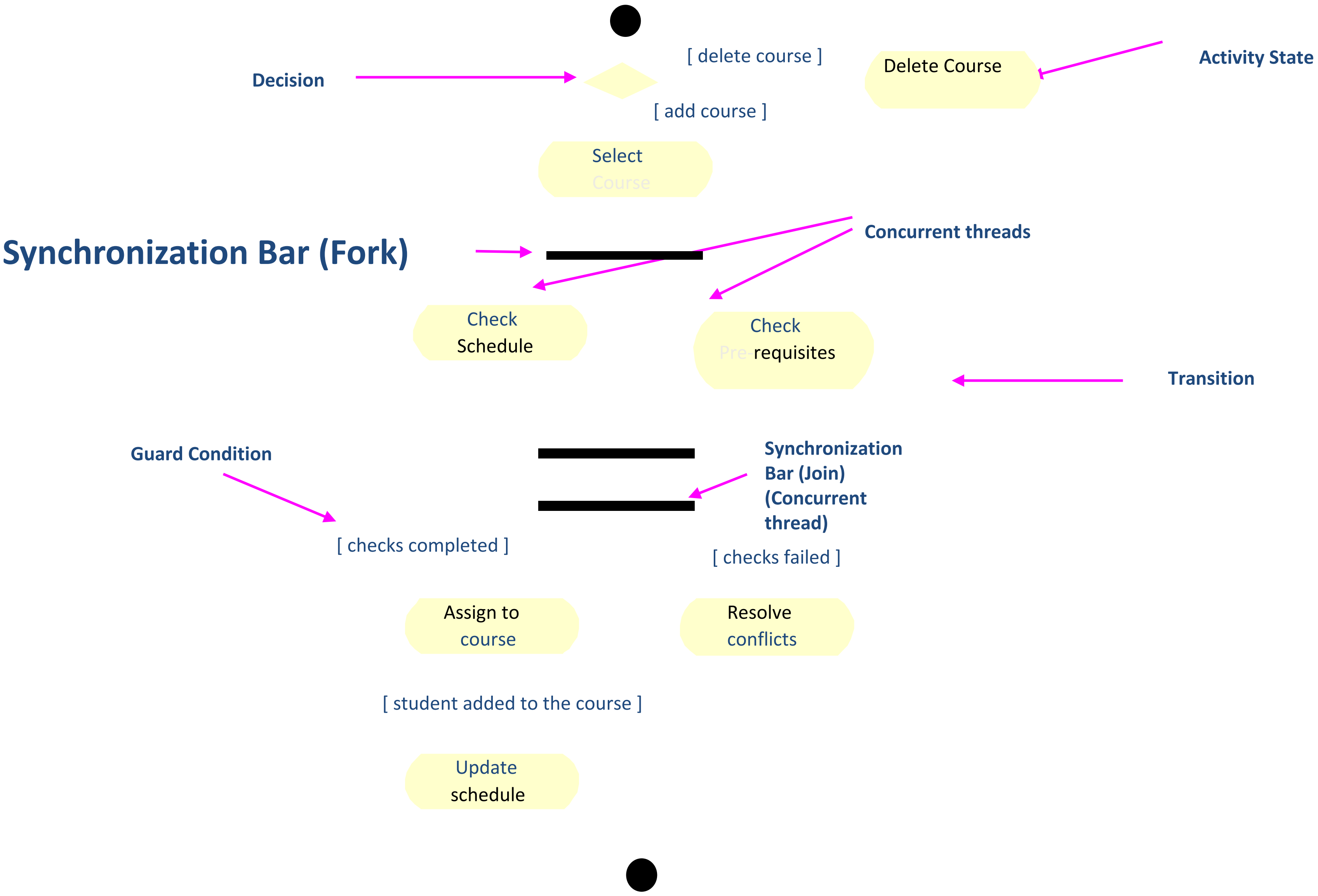

Ordered
orders again
orders again
Drink Ordering
customer orders
Food Ordering

What is Activity Diagram?

Activity Diagram

- Activity diagram also describe the dynamic aspects of the system.
- Activity diagram similar to a flowchart illustrates the flow from one activity to another activity.
  - The activity can be described as an operation of the system.
  - The control flow is drawn from one operation to another.
- Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc
  - The flow can be:
    - sequential, branched, or concurrent.

Example: Activity Diagram



[delete course]

Delete Course

Activity State

Decision

[add course]

Select Course

Synchronization Bar (Fork)

Concurrent threads

Check Schedule

Check Pre-requisites

Transition

Guard Condition

Synchronization Bar (Join) (Concurrent thread)

[checks completed]

[checks failed]

Assign to course

Resolve conflicts

[student added to the course]

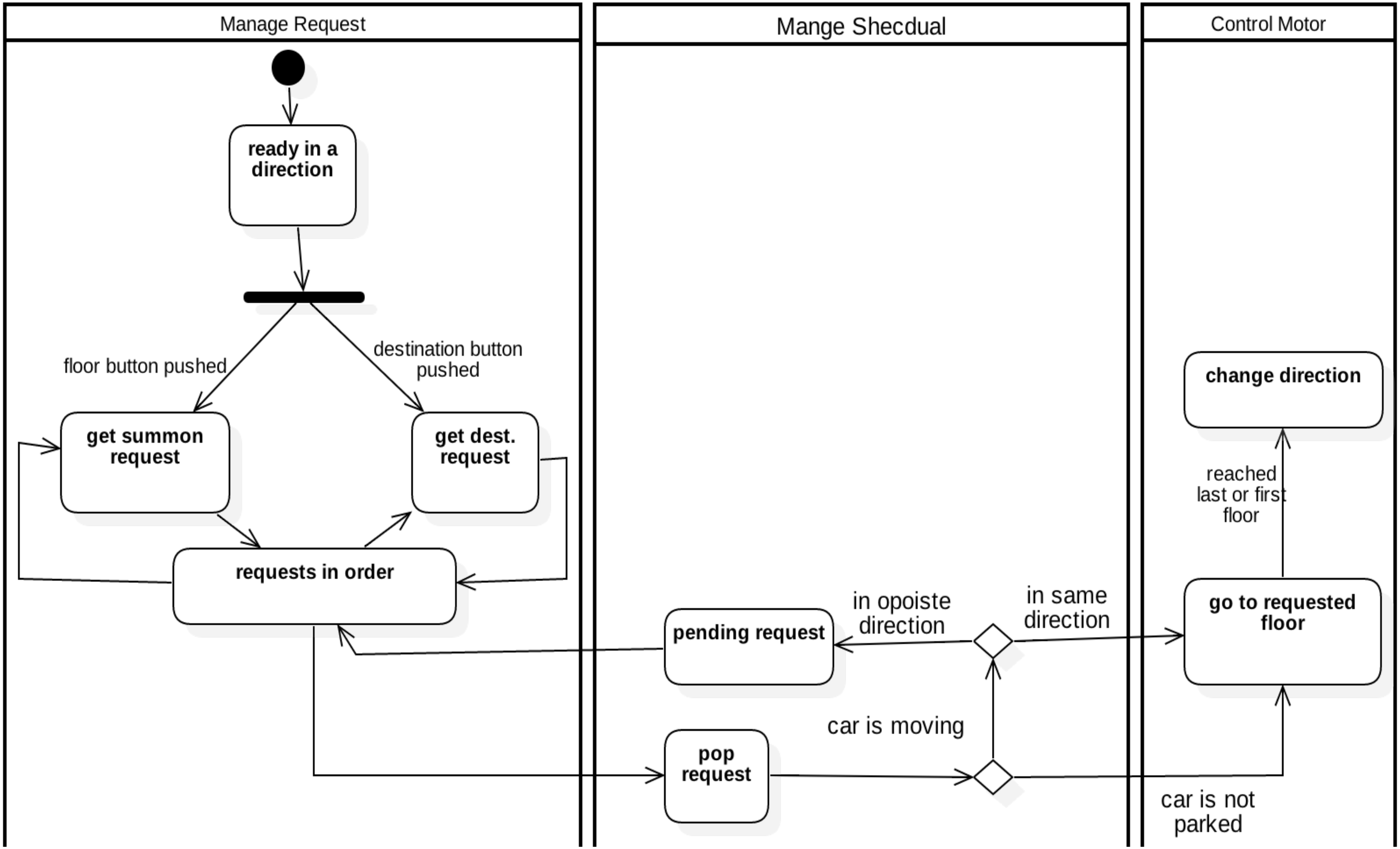
Update schedule

Picture from: Rational Rose, 2001

Example

Let's assume we want to design an Elevator System, and we need to develop an **activity diagram** for the system. This process starts when elevator is ready to move in a direction and based on the requests moves to certain floor.

Example of an Activity Diagram for An Elevator System (using Swim lanes)


Manage Request
Mange Shecdual
Control Motor
ready in a direction
floor button pushed
destination button pushed
get summon request
get dest. request
requests in order
change direction
reached last or first floor
pending request
in opoiste direction
in same direction
go to requested floor
pop request
car is moving
car is not parked