PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE

ACADEMIC YEAR: 2023-24

DEPARTMENT OF COMPUTER ENGINEERING

CLASS: B.E. **SEMESTER: I**

SUBJECT: Object oriented modelling and Design 410244(D)

ASSIGNMENT NO.	1
TITLE	**State Diagram**
PROBLEM STATEMENT /DEFINITION	Draw state model for telephone line, with various activities.
OBJECTIVE	• To learn and understand the State model elements . • To create a state diagram for a given system
OUTCOME	• Design the state model elements. • Create the state diagram using UML
S/W PACKAGES AND HARDWARE APPARATUS USED	i3/i5 PC with 8 GB RAM 64-bit Ubuntu OS, Modelio 3.0
REFERENCES	Textbooks: T1. Michael Blaha, James Rumbaugh, ―Object-Oriented Modeling and Design with UML‖, 2 nd Edition, Pearson Education, 2005. Reference books: 1. Grady Booch et al, ―Object-Oriented Analysis and Design with Applications‖, 3rd Edition, Pearson Education, 2007 2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado, UML 2 Toolkit‖, WileyDreamtech India, 2004
STEPS	Refer to theory and concepts
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Design diagrams 8. Description of diagram 10. Conclusion

Prerequisites: Object oriented programming concepts

Concept Related Theory:

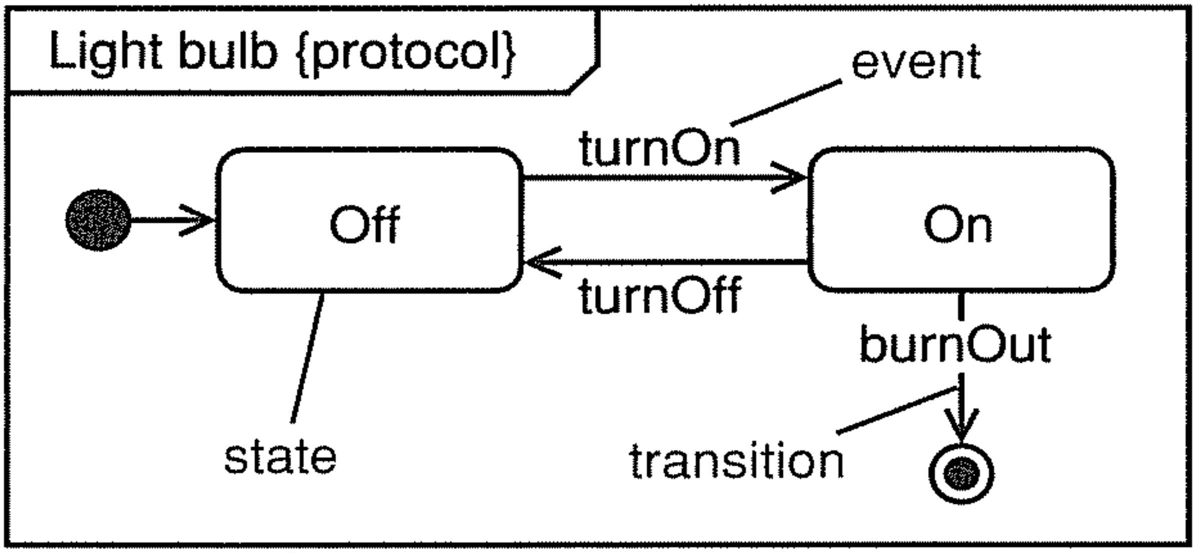
State machine

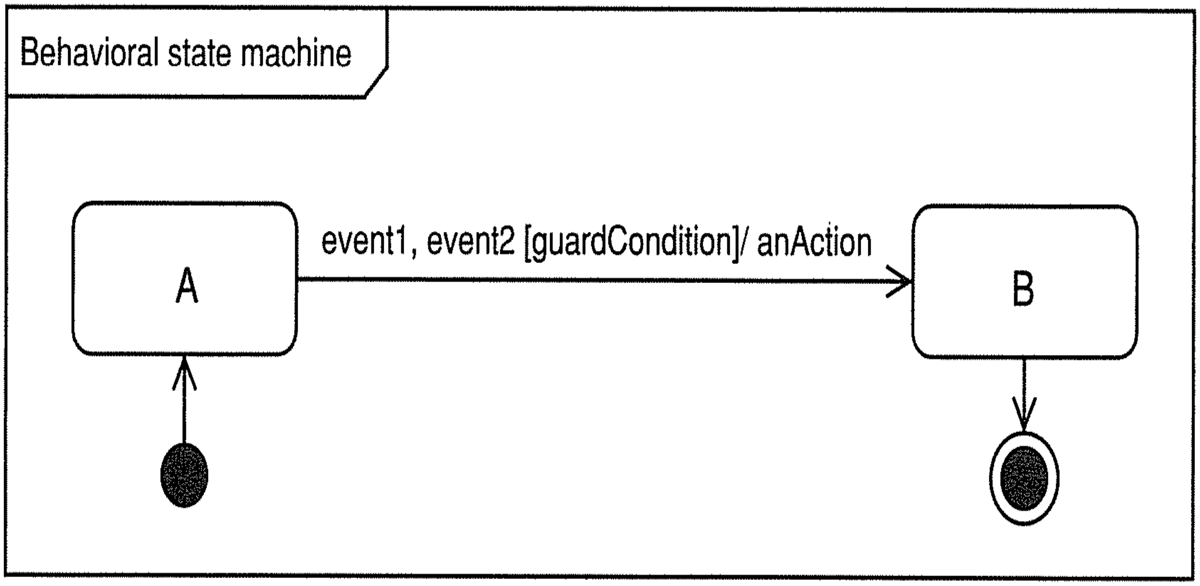
- Specifies the sequences of states of an object during its life time in response to events, with its responses to those events

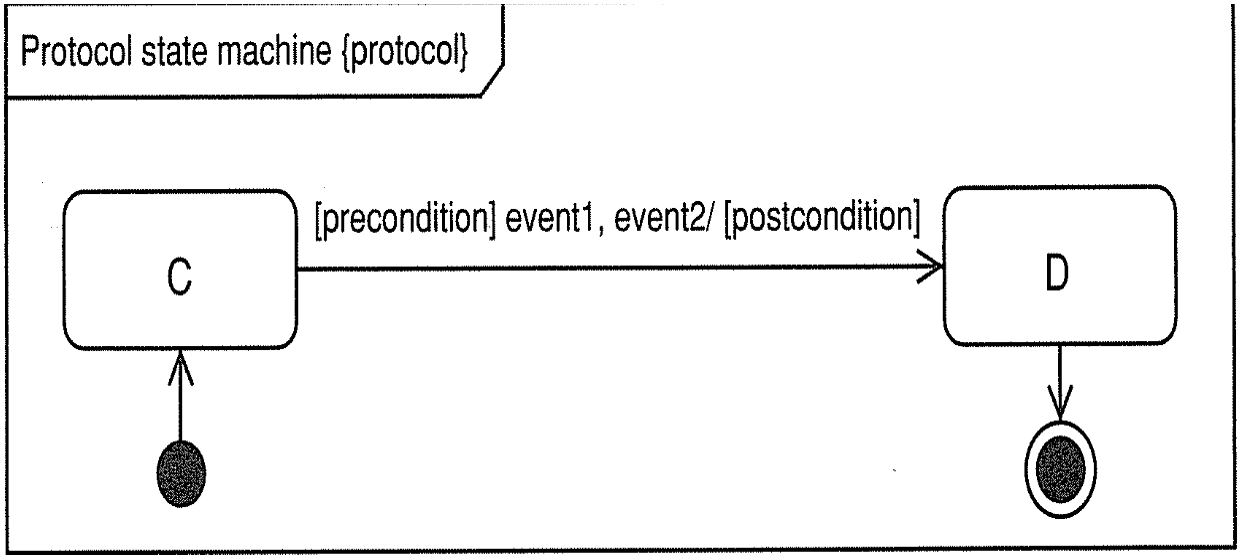
- State of an object refers to the time during which it satisfies some condition/perform activity/wait for an event
- Dynamics of execution can be given by
    - Activity diagram
    - State machine diagram (State transition diagram), state chart, and state transition table
- A finite state machine (state machine) is a conceptual machine with a finite number of states.
- Reactive object
    - Responds to external events
    - Generate and respond to internal events
    - Has lifecycle
    - Depends on past behavior

UML models behavior of instances of classifier like use cases, classes, subsystems, systems

- Behavioural State machine
    - Uses states, events and transitions to give behaviour of classifier
- Protocol State machine
    - Uses states, events and transitions to give behaviour of protocol
    - Conditions of operation call
    - Result of call
    - Ordering of call
    - No actions are specified in state
    - Only models appearance of behaviour of classifier
- Every state machine has a initial state
- Initial pseudo state marks the beginning of state machine
- One or more final states


Light bulb {protocol}
event
turnOn
Off
On
turnOff
burnOut
state
transition


Behavioral state machine
event1, event2 [guardCondition]/ anAction
A
B


Protocol state machine {protocol}
C
[precondition] event1, event2/ [postcondition]
D

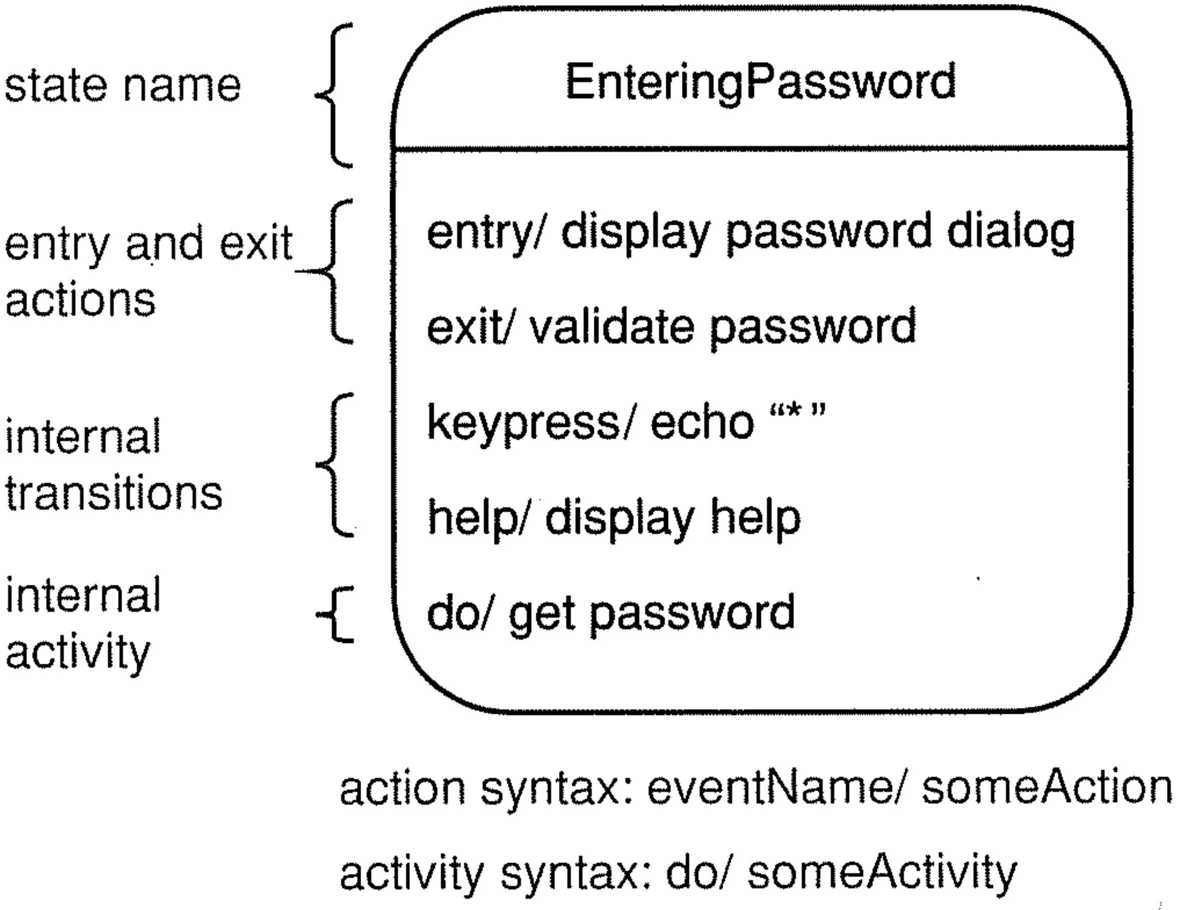
Modeling

- State machine models states of object
- May be instance of class/usecase/entire system
- State diagrams give graphical representation of state machines
- Sate Diagrams include
  - States
  - Events
  - Transition
  - Effects

States

- State of an object refers to the time during which it satisfies some condition/perform activity/wait for an event
- Example: Process states
  - Idle
  - Run
  - Wait
  - End
- Name: Unique name which is a textual string

- Entry/Exit Actions(Effects)
  - Actions executed on entering or exiting the state

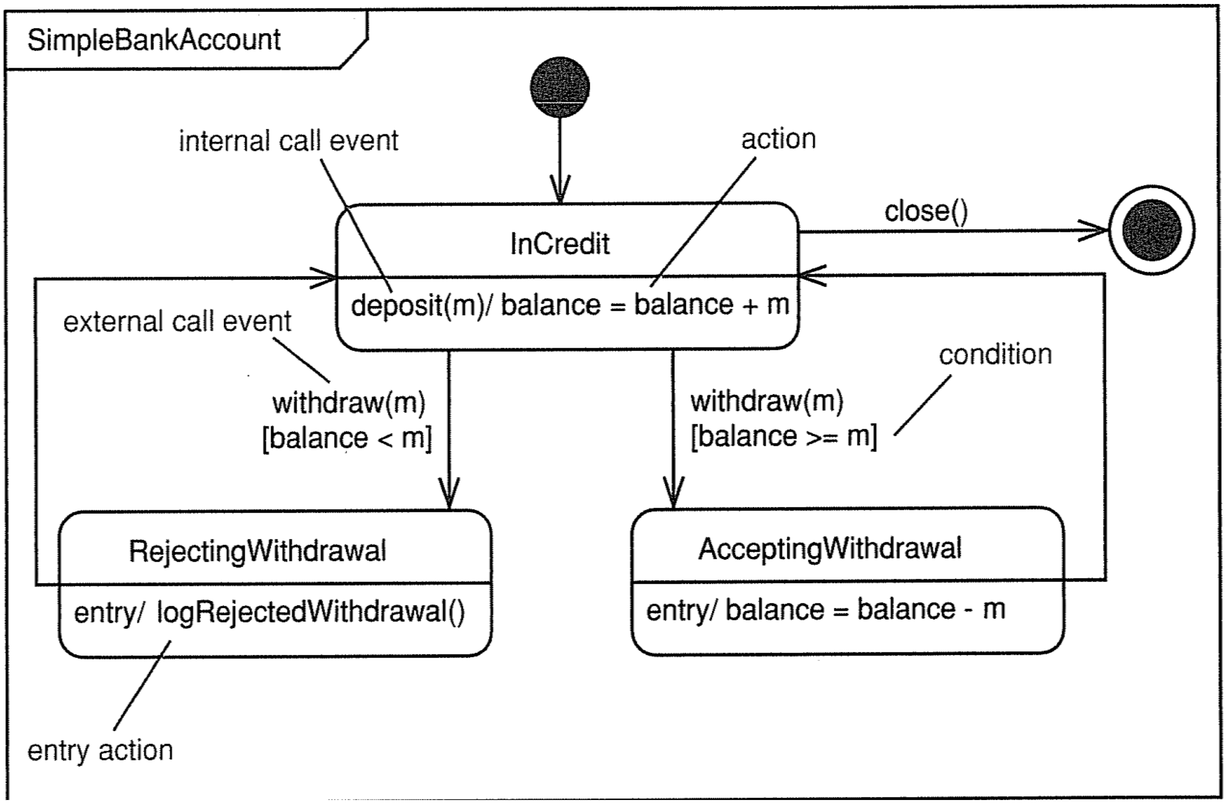


- Internal Transitions
  - No change in state -- keypress
- Substates
  - Nested structure of a state with sequential or orthogonal(concurrent) sub states
- Defered Events
  - Events that are not handled in that state but postponed and queued for handling by other state
- Initial and Final States
  - Special states or pseudo states
  - Initial state gives default starting place
  - The initial state of a state machine is the state that is entered when the state machine is

activated.

- A transition from start to normal state can not be a trigger event
- Final State gives the end of state machine or enclosing state
- **Transitions**
- Relationship between two objects when the change in state occurs in response to an event
- It has
- Source State: State affected by the transition
- Event : Makes the transition to fire
- Guard Condition: A Boolean expression that is evaluated when the transition is triggered and transition occurs only if it is true.
- Effect : Action on object
- Target State: State that is active after completion of transition
- **Event Trigger**
- Event is an occurrence which has value in time and space
- Occurrence of stimulus that triggers transition
- Event can be (Card Inserted, Pin Entered)
- Signals
- Calls
- The passing of time
- Change of state
- Completion transition is represented by transition without event trigger(Source completes behavior)
- Call Event
- When an operation is called
- Can be internal or external
- **Guard Condition**
- Condition to be true for the transition to change state

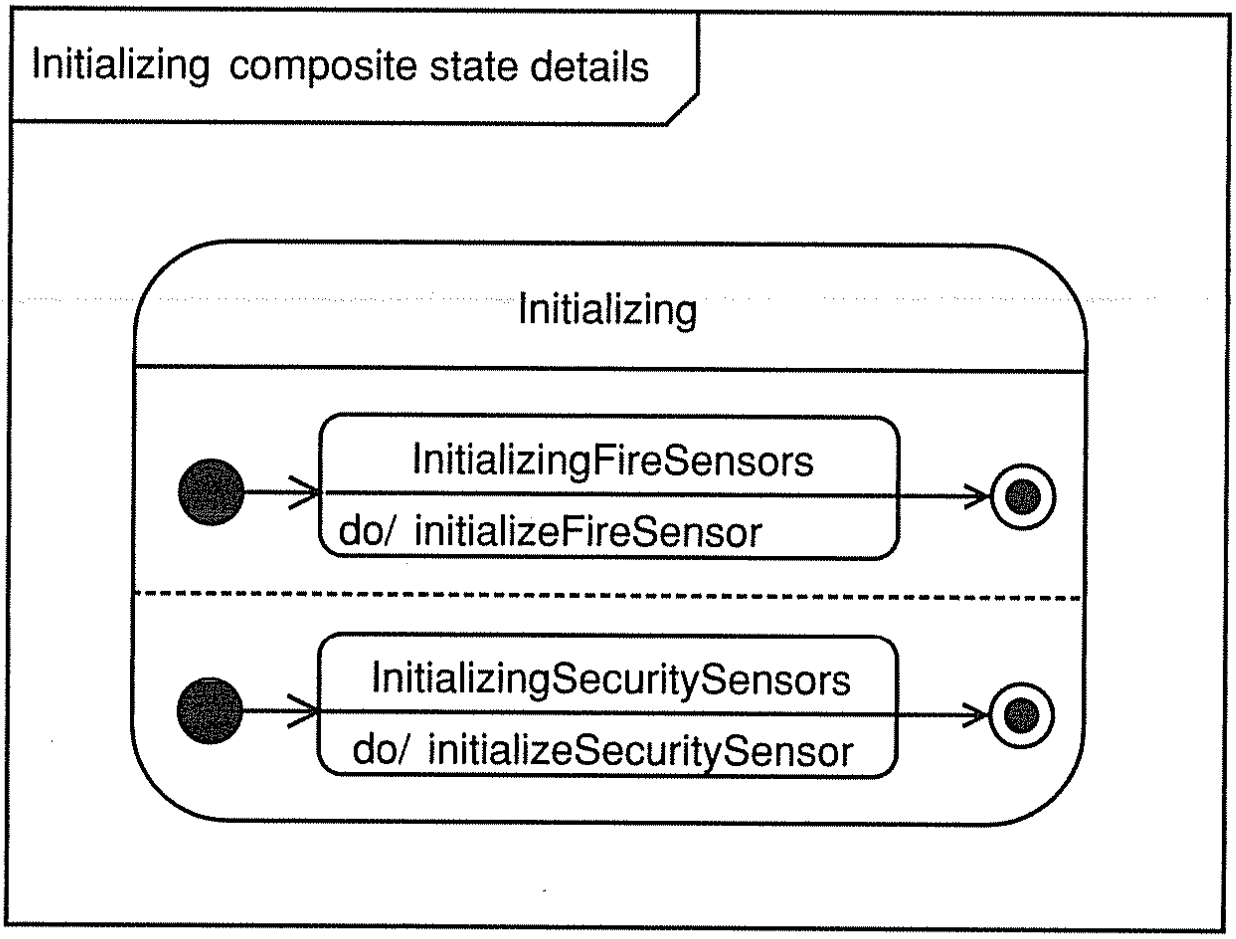
- ◦ Evaluated after the trigger event for its transition occurs
- ◦ Effect is the behaviour that is executed when transition occurs.
- ◦ Effect may be a operation call, sending a signal, computation, object creation/destruction
- ◦



- ◦

Orthogonal states

- Represent the state machines that execute in parallel
- A composite state is divided in to regions
- AN object will be in a stste from each of the orthogonal regions
- Fork and join are used to pass control to any state in orthogonal regions and join control from any state in orthogonal regions


Initializing composite state details
Initializing
InitializingFireSensors
do/ initializeFireSensor
InitializingSecuritySensors
do/ initializeSecuritySensor

Review Questions:

Q1. What are elements of state machine diagram?

Q2. What are orthogonal states?

Q3.Are there multiple end exist?

Q4. How transition is shown with activity?

Q5. When state diagrams are modelled in software?

PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE

ACADEMIC YEAR: 2023-24

DEPARTMENT OF COMPUTER ENGINEERING

CLASS: B.E. **SEMESTER: I**

SUBJECT: Object oriented modelling and Design 410244(D)

ASSIGNMENT NO.	2
TITLE	**Class Diagram**
PROBLEM STATEMENT /DEFINITION	Draw basic class diagrams to identify and describe key concepts like classes, types in your system and their relationships
OBJECTIVE	• To learn and understand the class model elements . • To create a class diagram for a given system
OUTCOME	• Design the class model elements. • Create the class diagram using UML
S/W PACKAGES AND HARDWARE APPARATUS USED	i3/i5 PC with 8 GB RAM 64-bit Ubuntu OS, Modelio 3.0
REFERENCES	Textbooks: T1. Michael Blaha, James Rumbaugh, ―Object-Oriented Modeling and Design with UML‖, 2 nd Edition, Pearson Education, 2005. Reference books: 1. Grady Booch et al, ―Object-Oriented Analysis and Design with Applications‖, 3rd Edition, Pearson Education, 2007 2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado, UML 2 Toolkit‖, WileyDreamtech India, 2004
STEPS	Refer to theory and concepts
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Design diagrams 8. Description of diagram 10. Conclusion

Prerequisites: Object oriented programming concepts

Concept Related Theory:

CLASS

- Building blocks of object oriented system

- Description of set of objects that share same attributes, operations, relationships and semantics
- Represent s/w, h/w & conceptual things

- Name – unique in its package
- Attributes – give the property of class

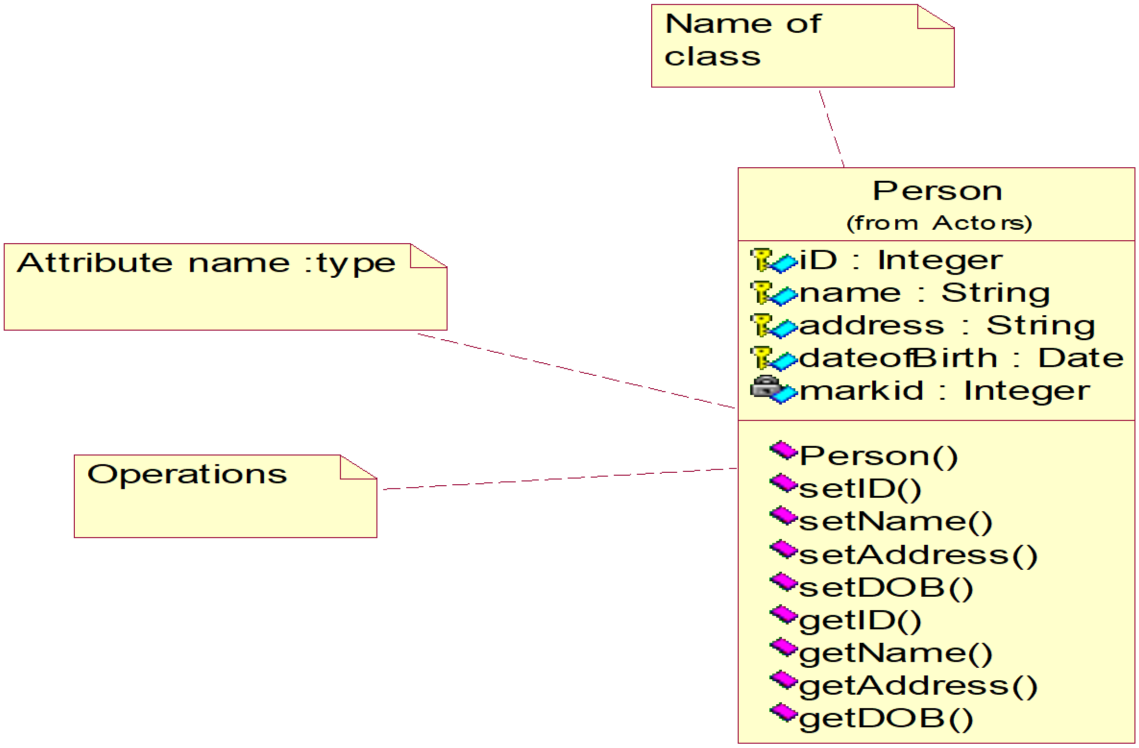
<visibility> <name> : <type> = <default value>

Visibility

+ public

- private

# protected



- Operations – implement the service of the class

give the behaviour of the class

<visibility> <name> (<parameter list>): <return type> {property string}

- Parameter list

<direction> : <name> : <type> = <default value>

direction

in/out/inout

Relationships

- Dependency
  - Using relationships
  - Change in depended class will affect depending class
  - Class in signature of operation of a class
  - Stereotypes are used
- Generalization
  - Relationship between a general and specific kind
  - Parent and child relationships
  - Association
  - Structural relationships
  - Navigates from one object to another
  - Named association
  - Role names are associated with relation
  - Multiplicity gives the no of objects connected across association -- 1, 0 .. 1, 0..*, 1 .. *, 0..1,3,6..*
- Aggregation
  - Whole/part relationship
  - Special kind of association
- Composition
  - Part is always associated with a single whole

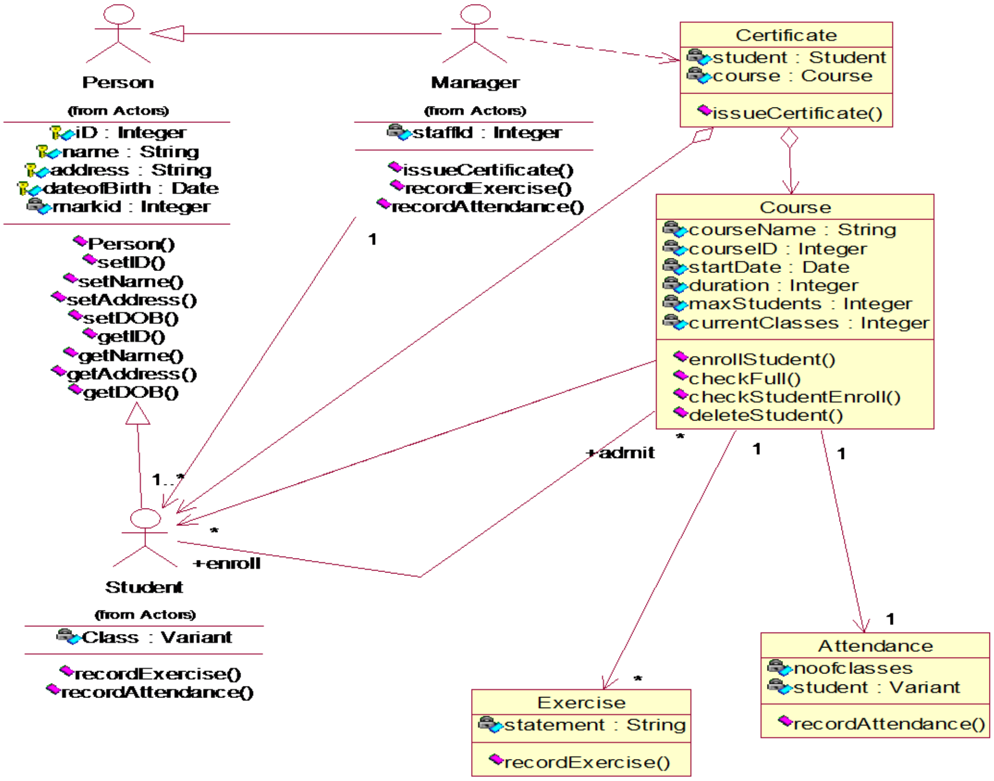
Class Diagrams

- Shows the set of classes , interfaces, & collaborations and their relationships
- Static design view of the system

- Help in understanding the responsibilities of each class
- Finding classes
    - Give the structure of the each class and semantics
    - Noun verb analysis
    - CRC analysis
    - Finding RUP Analysis classes Entity, Boundary and Control Classes
    - Physical objects: people, devices
    - Documents: Roll List, Table of accounts
    - Interfaces : keyboard, screen particularly in embedded system
    - Conceptual things: BankAccount, File

Class Modeling

- Identify classes from many methods
- Combine the sets of classes by removing duplicates
- Identify relation using collaboration among classes
- Refine classes



Review Questions:

Q1. What are elements of class diagram?

Q2. What are static and behaviour diagrams?

Q3. Which relationships are used in class diagram?

Q4. How class diagram relate to other diagrams

Q5. What are the extra features in class diagram?

PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE

ACADEMIC YEAR: 2023-24

DEPARTMENT OF COMPUTER ENGINEERING

CLASS: B.E. **SEMESTER: I**

SUBJECT: Object oriented modelling and Design 410244(D)

ASSIGNMENT NO.	3
TITLE	**Use case Diagram**
PROBLEM STATEMENT /DEFINITION	Draw one or more Use Case diagrams for capturing and representing requirements of the system. Use case diagrams must include template showing description and steps of the Use Case for various scenarios.
OBJECTIVE	• To learn and understand the use case model elements . • To create a use case diagram for a given system
OUTCOME	• Design the use case model elements. • Create the use case diagram using UML
S/W PACKAGES AND HARDWARE APPARATUS USED	i3/i5 PC with 8 GB RAM 64-bit Ubuntu OS, Modelio 3.0
REFERENCES	Textbooks: T1. Michael Blaha, James Rumbaugh, ―Object-Oriented Modeling and Design with UML‖, 2 nd Edition, Pearson Education, 2005. Reference books: 1. Grady Booch et al, ―Object-Oriented Analysis and Design with Applications‖, 3rd Edition, Pearson Education, 2007 2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado, UML 2 Toolkit‖, WileyDreamtech India, 2004
STEPS	Refer to theory and concepts
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Design diagrams 8. Description of diagram 10. Conclusion

Prerequisites: Object oriented programming concepts

Concept Related Theory:

Use Case Modeling

- Functional requirements of the system are defined in terms of use cases and actors.
- The relationship between actors, use cases and between use cases are analysed and modelled. This gives the static model.
- The system behaviour and interaction of actor with the use case scenario is analysed for each use case.
- The use case behaviour can be modelled as description or as activity.

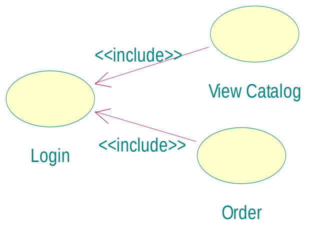
Use cases

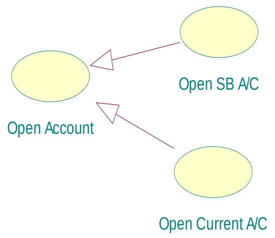
- Use case is a description of set of sequence of actions to yield an observable result of value to an Actor
- A scenario is a sequence of steps describing an interaction between user and the system
- Use case is a set of scenarios tied together by a common user goal
- Use case specifies the behaviour of the system or part of the system

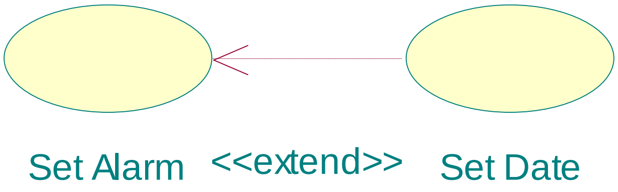
Use case Specification

- No standard for specification
- Steps describing scenario
- Alternative scenarios
- Preconditions
- Post conditions
- Activity/interaction diagrams

Organizing Use Cases





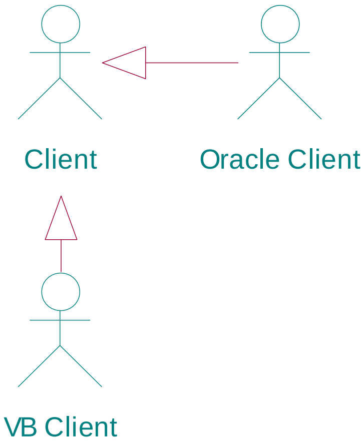


Actors

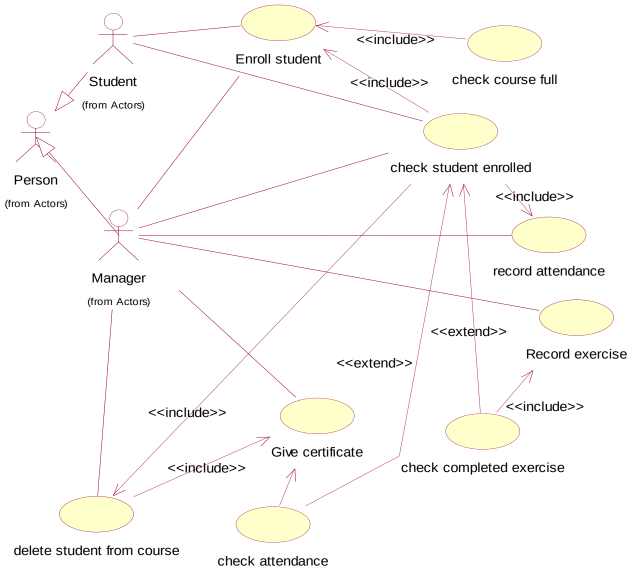
- A role that a user plays when interacting with the usecases
- Many-Many relationship between actor and usecase
- Human being or any external system that needs some information from the current system
- Specialized using generalization relationships
- Connected to usecases by association relationship

Identification of Actors

- **Actor can be users of the system**
- **The external system which may access information from the current system**
- **The Roles in the system**
- **The one who gets a value from the USECASE**



Use case diagram for Course Management system



Review questions:

Q1. What are elements of use case diagram?

Q2. What are static and behaviour diagrams?

Q3. Which relationships are used in use case diagram?

Q4. How use cases and actors are organized?

Q5. How to describe a use case?

PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE

ACADEMIC YEAR: 2023-24

DEPARTMENT OF COMPUTER ENGINEERING

CLASS: B.E. **SEMESTER: I**

SUBJECT: Object oriented modelling and Design 410244(D)

ASSIGNMENT NO.	4
TITLE	**Activity Diagram**
PROBLEM STATEMENT /DEFINITION	Draw activity diagrams to display either business flows or like flow charts.
OBJECTIVE	• To learn and understand the Activity model elements . • To create a Activity diagram for a given system
OUTCOME	• Design the Activity model elements. • Create the activity diagram using UML
S/W PACKAGES AND HARDWARE APPARATUS USED	i3/i5 PC with 8 GB RAM 64-bit Ubuntu OS, Modelio 3.0
REFERENCES	Textbooks: T1. Michael Blaha, James Rumbaugh, ―Object-Oriented Modeling and Design with UML‖, 2 nd Edition, Pearson Education, 2005. Reference books: 1. Grady Booch et al, ―Object-Oriented Analysis and Design with Applications‖, 3rd Edition, Pearson Education, 2007 2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado, UML 2 Toolkit‖, WileyDreamtech India, 2004
STEPS	Refer to theory and concepts
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Design diagrams 8. Description of diagram 10. Conclusion

Prerequisites: Object oriented programming concepts

Concept Related Theory:

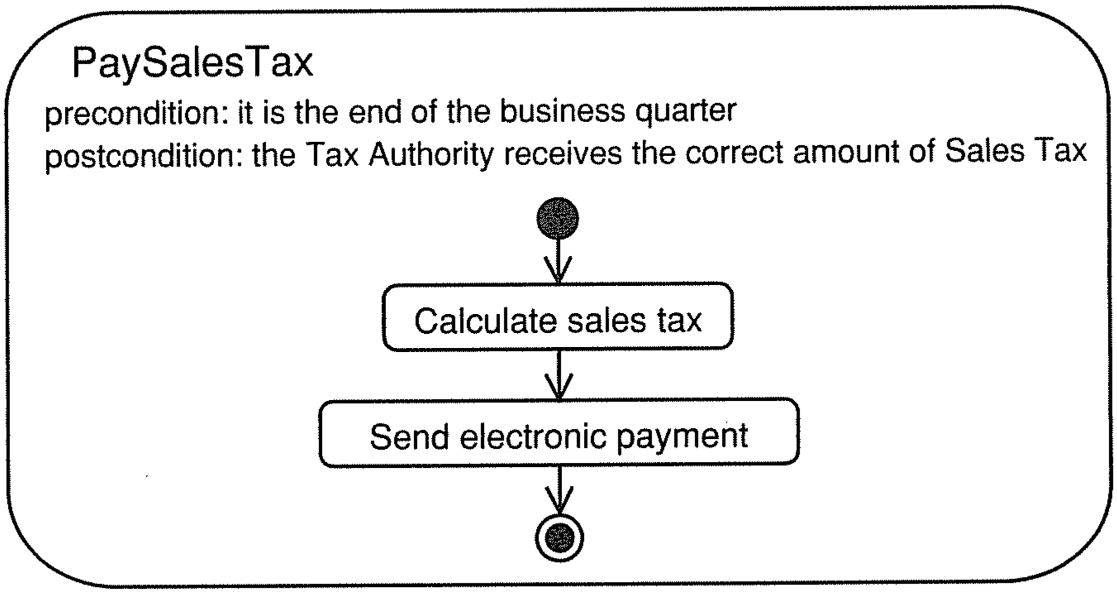
- **Activity diagram shows**

  - Flow of control from one activity to another
  - Modeling
    - Control and concurrency
    - Sequential steps
    - Flow of values among steps
  - Used to model
    - Behaviour of use case
    - Operation
    - Algorithm
    - Business process
  - Activity consists of one or more actions
  - Activities are network of nodes connected by edges
  - Elements of Activity diagram are given below
  - Action Node is an atomic operation
    - Operation call
    - Sending a signal
    - Creation/destruction of object
    - Computation
  - Control nodes
    - Control the flow through activity
  - Object Nodes
    - Represent objects used in the activity

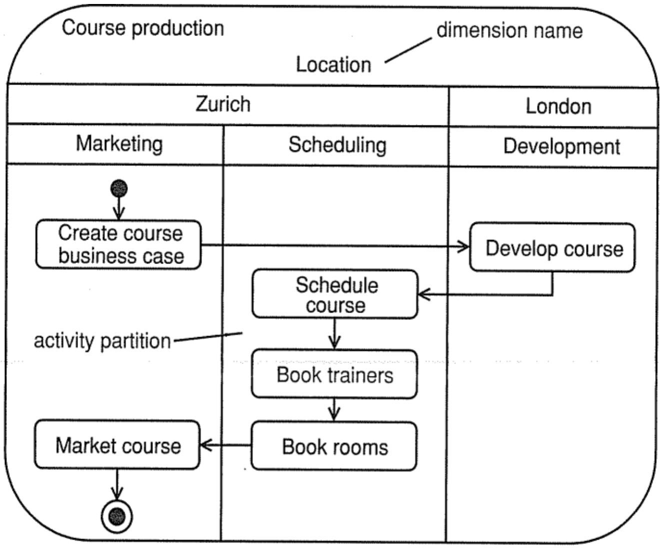
Elements

- Control Flow
  - Flow of control through activity
- Object Flow
  - Flow of object through activity

- Precondition
  - To be true before activity starts
- Post condition
  - To be true after activity starts

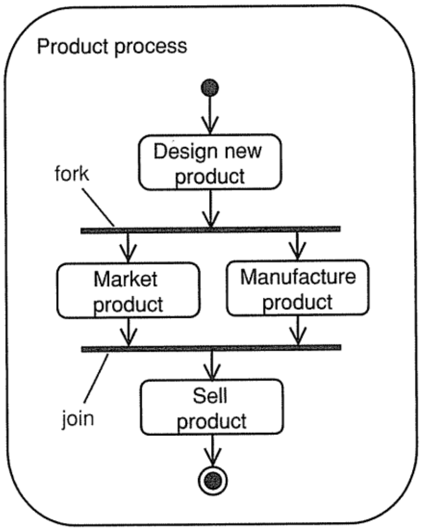

PaySalesTax
precondition: it is the end of the business quarter
postcondition: the Tax Authority receives the correct amount of Sales Tax
Calculate sales tax
Send electronic payment

- **Swimlane or Partition**
  - Divides the activity in to grouping of actions
  - Horizontal or vertical
  - Can be grouped according to responsible for those actions like class, Role, location



Concurrency

- **Fork node splits the flow in to multiple concurrent flows**
- **Fork node has one input and multiple output**
- **Join node has many inputs and one output**
- **It synchronizes multiple flows of control**
- **'AND' operation is performed at node on all inputs edges**



Review questions:

Q1. What are elements of activity diagram?

Q2. What are the advantages of activity diagram?

Q3. Which control elements are used in activity diagram?

Q4. How to show parallel flows in UML?

Q5. What is action and activity?

PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE

ACADEMIC YEAR: 2023-24

DEPARTMENT OF COMPUTER ENGINEERING

CLASS: B.E. **SEMESTER: I**

SUBJECT: Object oriented modelling and Design 410244(D)

ASSIGNMENT NO.	5
TITLE	**Component Diagram**
PROBLEM STATEMENT /DEFINITION	Draw component diagrams assuming that you will build your system reusing existing components along with a few new ones
OBJECTIVE	• To learn and understand the Component model elements . • To create a component diagram for a given system
OUTCOME	• Design the component model elements. • Create the component diagram using UML
S/W PACKAGES AND HARDWARE APPARATUS USED	i3/i5 PC with 8 GB RAM 64-bit Ubuntu OS, Modelio 3.0
REFERENCES	Textbooks: T1. Michael Blaha, James Rumbaugh, ―Object-Oriented Modeling and Design with UML‖, 2 nd Edition, Pearson Education, 2005. Reference books: 1. Grady Booch et al, ―Object-Oriented Analysis and Design with Applications‖, 3rd Edition, Pearson Education, 2007 2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado, UML 2 Toolkit‖, WileyDreamtech India, 2004
STEPS	Refer to theory and concepts
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. 3. Problem definition 4. Learning objective 5. Learning Outcome 6. Concepts related Theory 7. Design diagrams 8. Description of diagram 10. Conclusion

Prerequisites: Object oriented programming concepts

Concept Related Theory:

Component

- A component is a logical , replaceable part of a system that conforms to and provides the realization of a set of interfaces

- Components can be replaced by newer and compatible ones
- An interface is a collection of operations that specify a service that is provided by or requested from a class or component
- Interfaces allow the components to build the implementation of a component using other smaller components

Ports

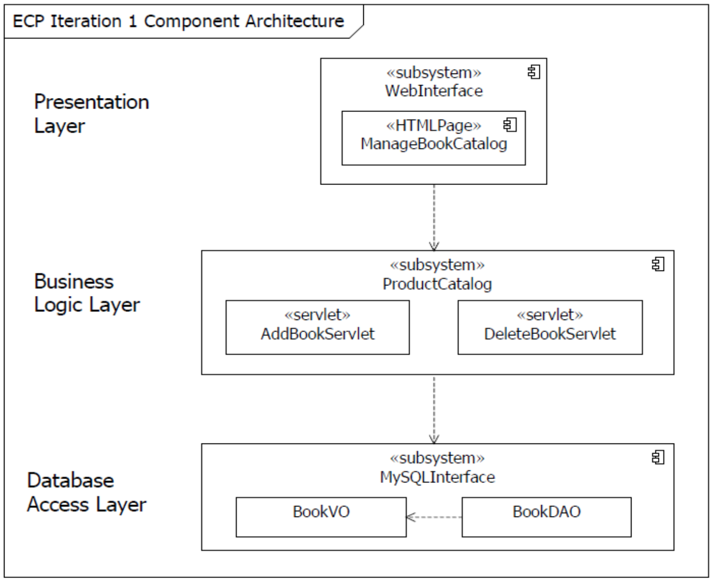
- A port is a specific window into an encapsulated component accepting messages to and from the component conforming to specific interfaces
- Internal structure is the implementation of a component a set of parts that are connected together in a specific way
- Connector is a communication relationship between two parts or ports within the context of a component

Interface: Provided & Required

- An interface that a component realizes is called a provided interface
- An interface that a component uses is called a required interface
- There may be more than one provided and required interface for a component
- The relation between a component and interface is shown with Realization / Dependency or ball and socket notation
- A component is replaceable
    - A component can be replaced by another component that conforms to the same interfaces
- A component is part of the system
    - It is reusable and collaborates with other components
- The internal structure of a component is shown with parts, connectors and ports which is also called as white box view

Component Diagram

- Component diagram shows all the components and their relationships like dependencies, generalization ,realization
- Modeling a structured class
- Modeling a component with API


ECP Iteration 1 Component Architecture
Presentation Layer
Business Logic Layer
Database Access Layer
«subsystem» WebInterface
«HTMLPage» ManageBookCatalog
«subsystem» ProductCatalog
«servlet» AddBookServlet
«servlet» DeleteBookServlet
«subsystem» MySQLInterface
BookVO
BookDAO

Review questions:

Q1. What are elements of component diagram?

Q2. How reusability is achieved in component diagram?

Q3. What are interfaces in component diagram?

Q4. When component diagram is used?

Q5. What are provided and required interfaces?