Lecture 10

UML Views

Objectives

- At the end of this lecture, the student should :
- Know the purpose of the 5 views that UML models provide
- Be able to distinguish between the different types of UML diagrams

Overview of UML

- UML is a language for
  - Visualising
  - Specifying
  - Constructing
  - Documenting

  the artifacts of a software-intensive system

Visualising

- Graphical nature of UML allows you visualise a system before it is implemented
- Can communicate to a wider audience – better than using a textual description or a programming language

Specifying

- Used to specify “what” is required of a system
- Used to specify “how” a system must be implemented
- Captures all the important requirements, analysis, design and implementation decisions required

Constructing

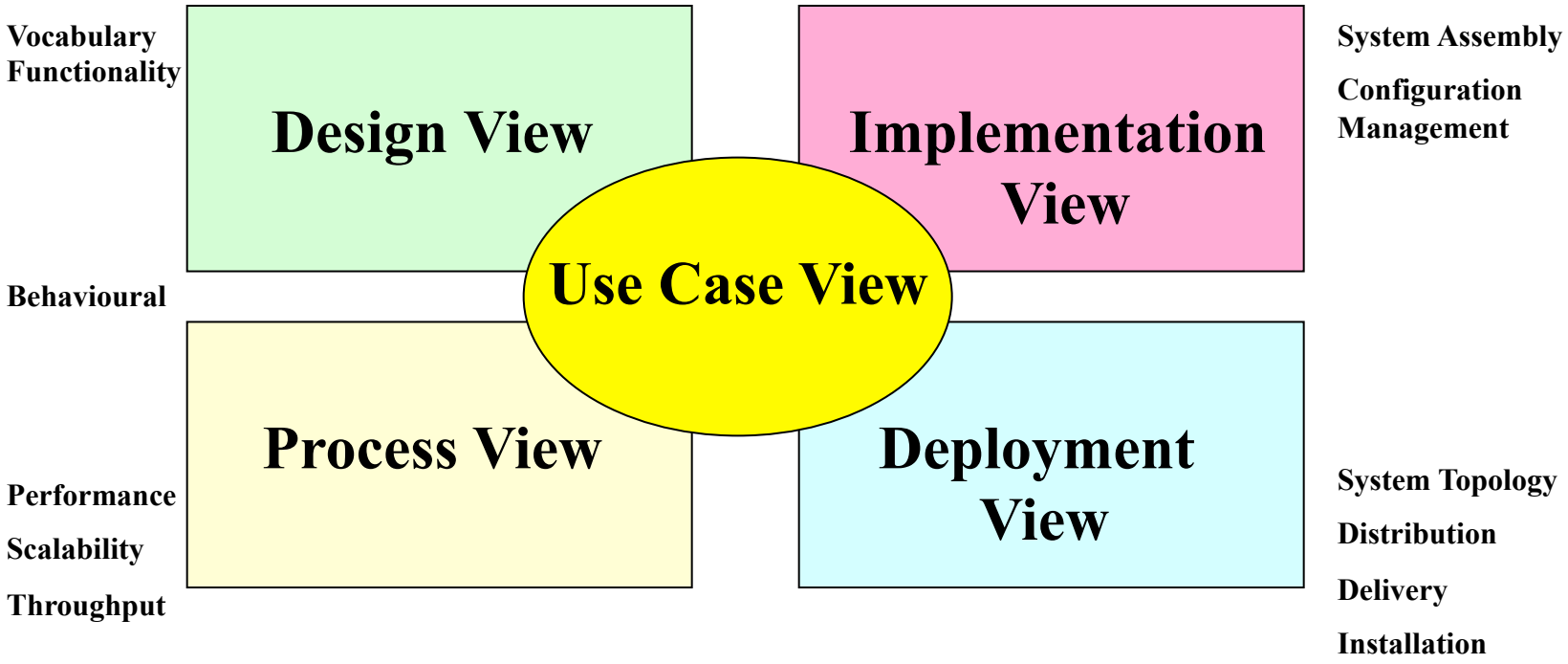
- UML can guide the implementation of a system
- Can use UML Case Tools to generate OO source code

Documenting

- Can document deliverables using UML e.g. requirements doc., design docs, test plans etc. all important for controlling and managing a project

UML Views --- 5 Views

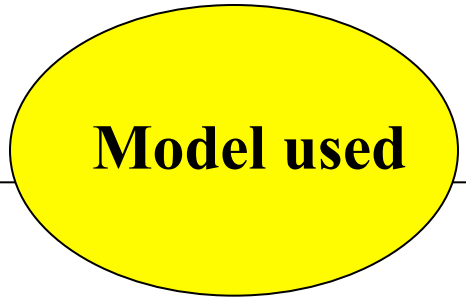
UML splits the modelling of systems up into five main views as shown in the diagram below.



Use Case View

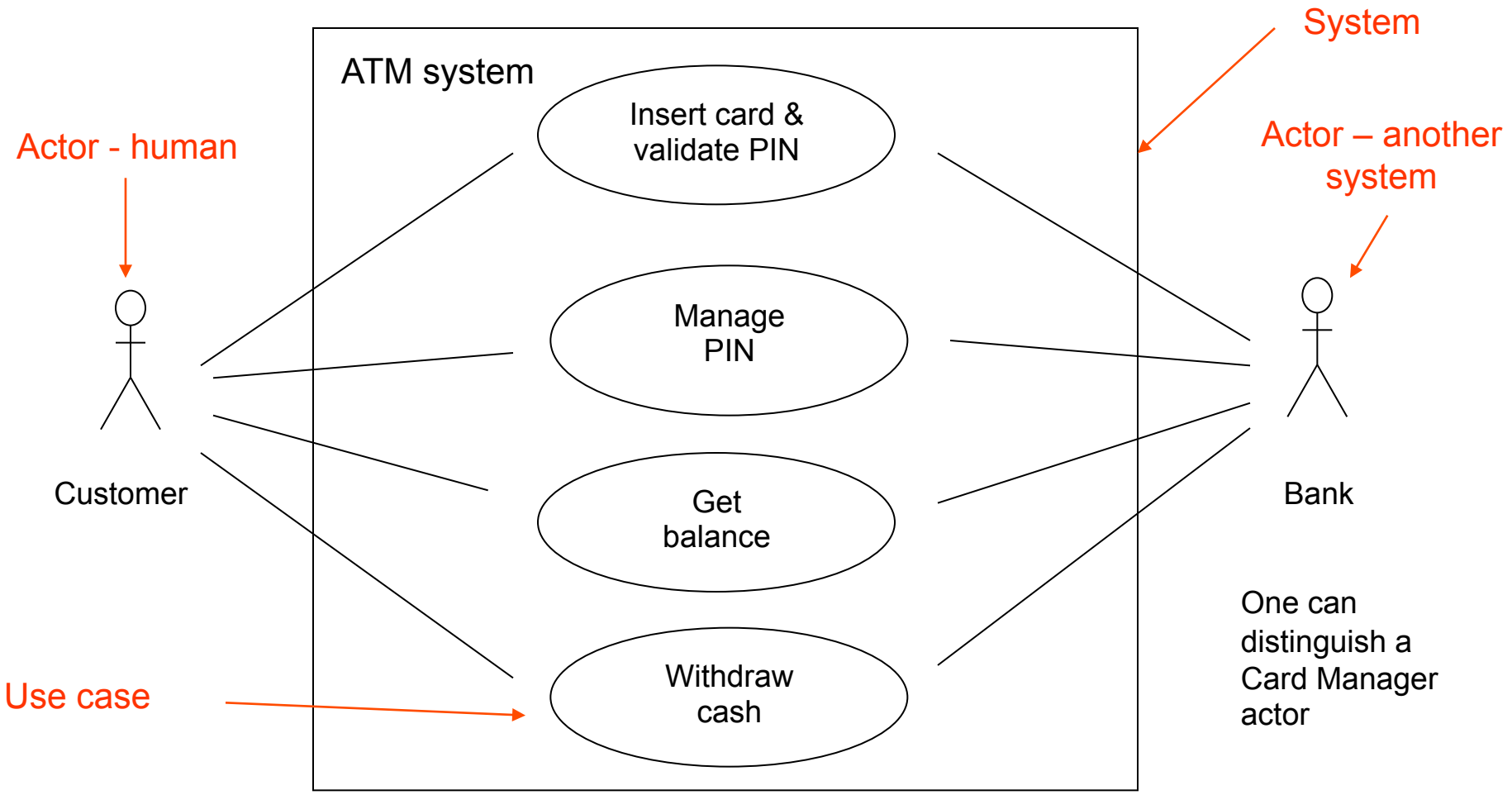
- Looks at a problem and solution from the perspective of those individuals whose problem the solution addresses … end users
- Focuses on **WHAT** the system does without showing **HOW** the system achieves it.
- Of particular interest to end users, analysts, testers.

Use Case View



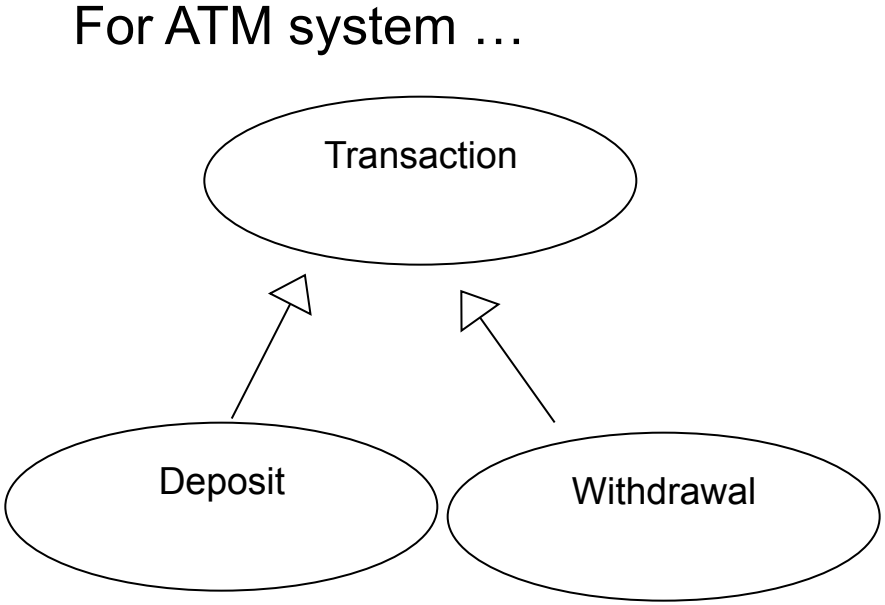
- Use Case Diagram
- Shows the functionality required by the system and the interaction of actors (i.e. users etc.) with the system

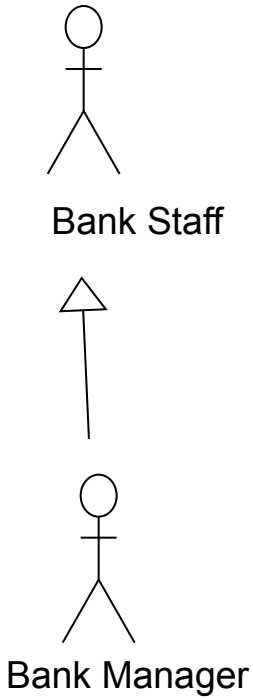
Use Case – ATM System



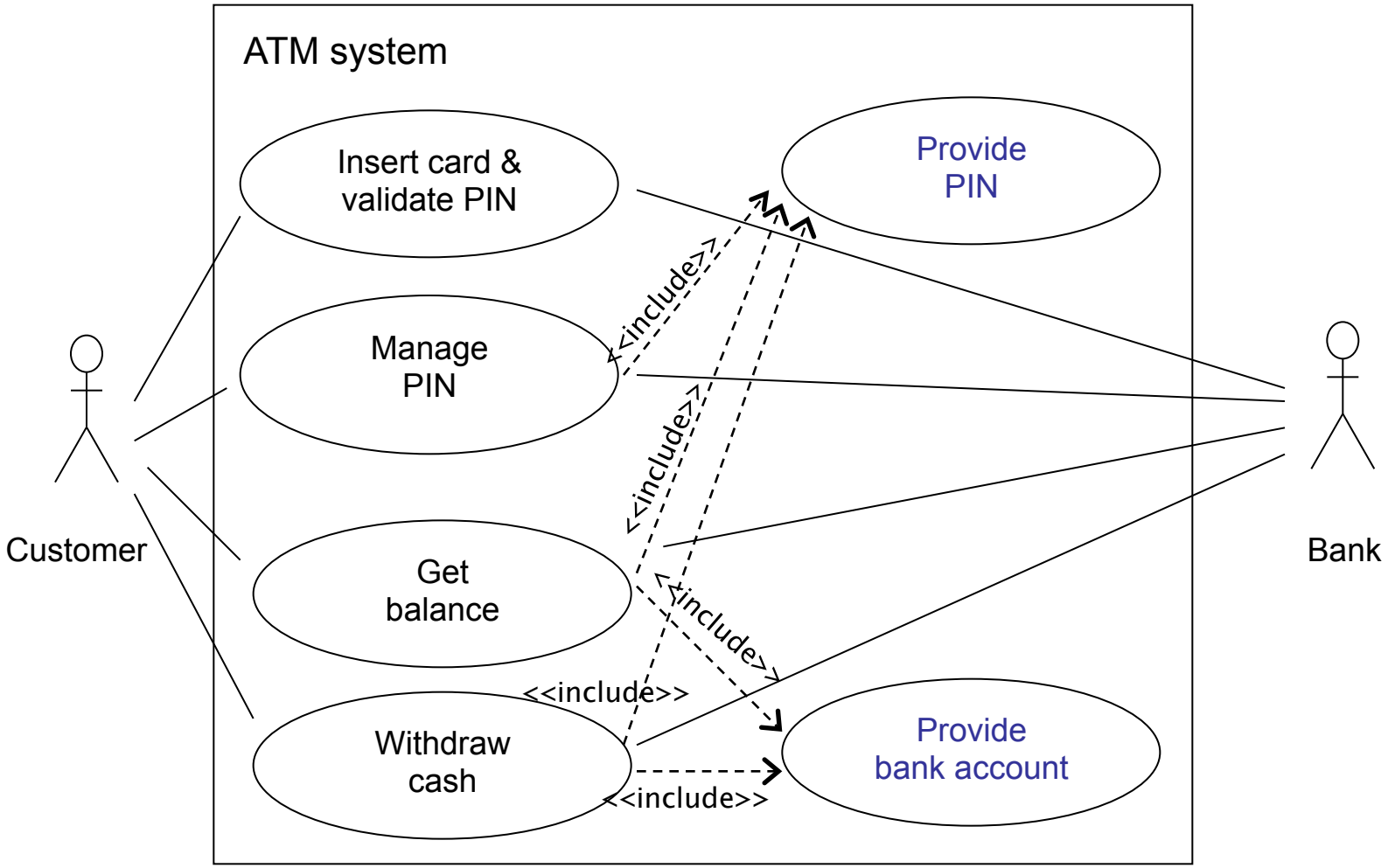
Use Case - Generalisations

- A generalisation relationship may exist between use cases as well as actors.
- X is a generalisation of Y if Y-is-a-X.
- Direction of generalisation is from specific to general.
- For ATM system …

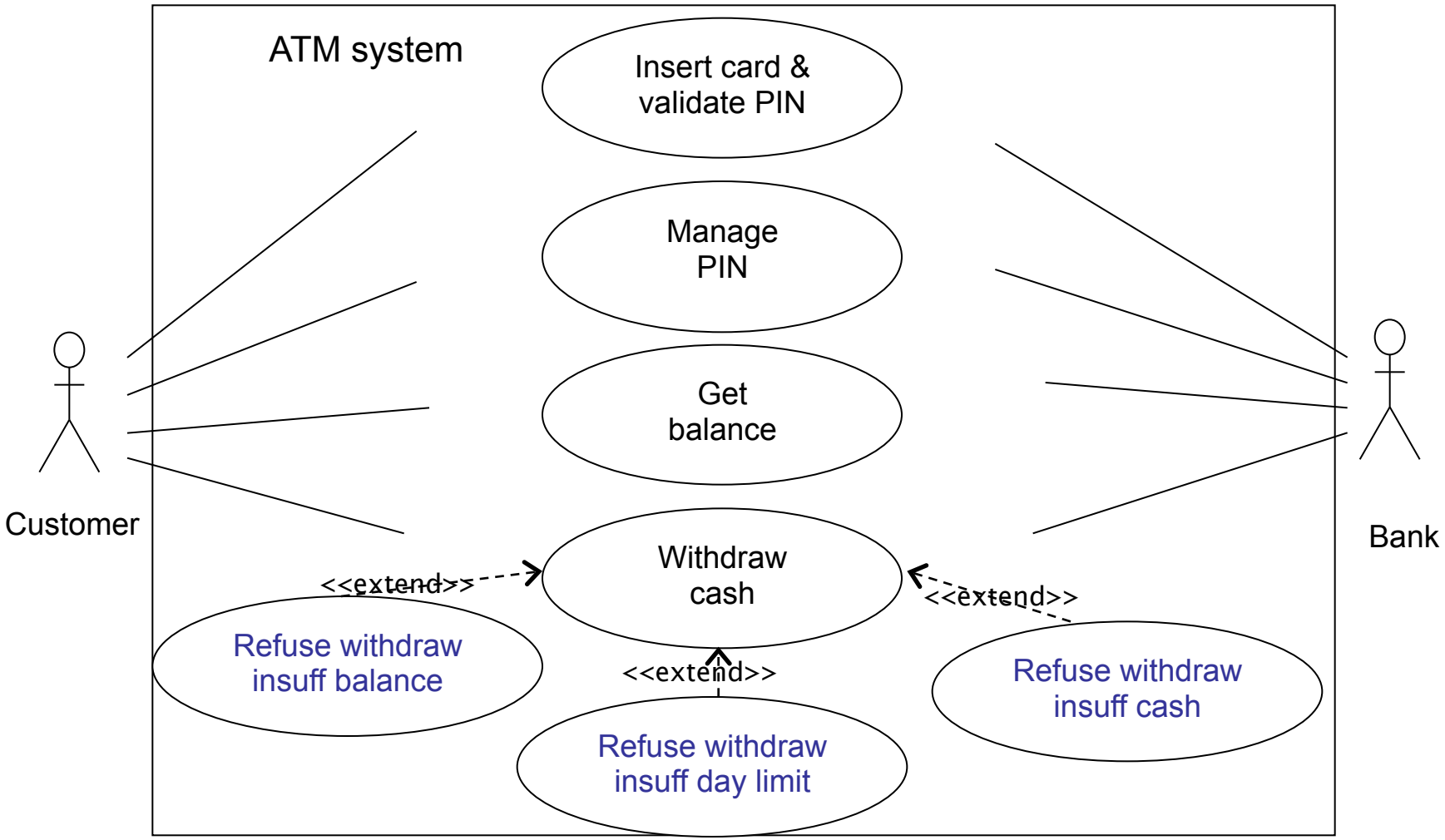




ATM system with <<include>>



ATM system with <<extend>>

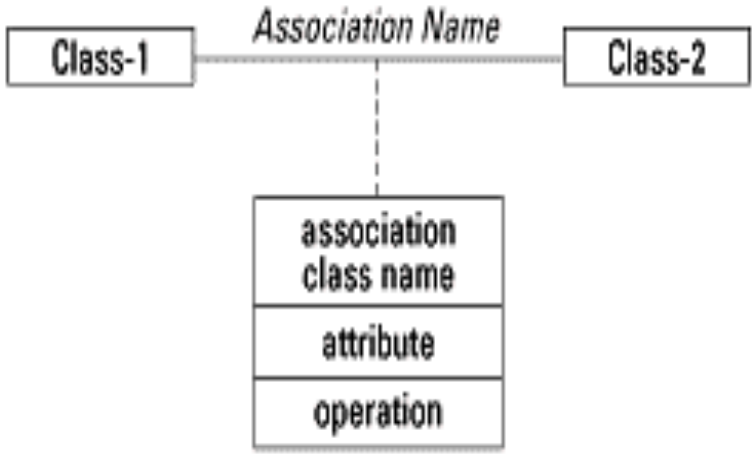


Design View

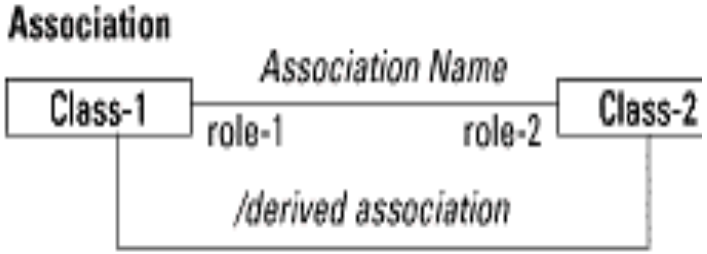
Models used

- Details the vocabulary of a system
  - Includes the **classes, objects, relations, interfaces** etc. of the system
  - Focuses on **HOW** the system does what it is required i.e. how the functional requirements are met.
  - Of particular interest to software developers.

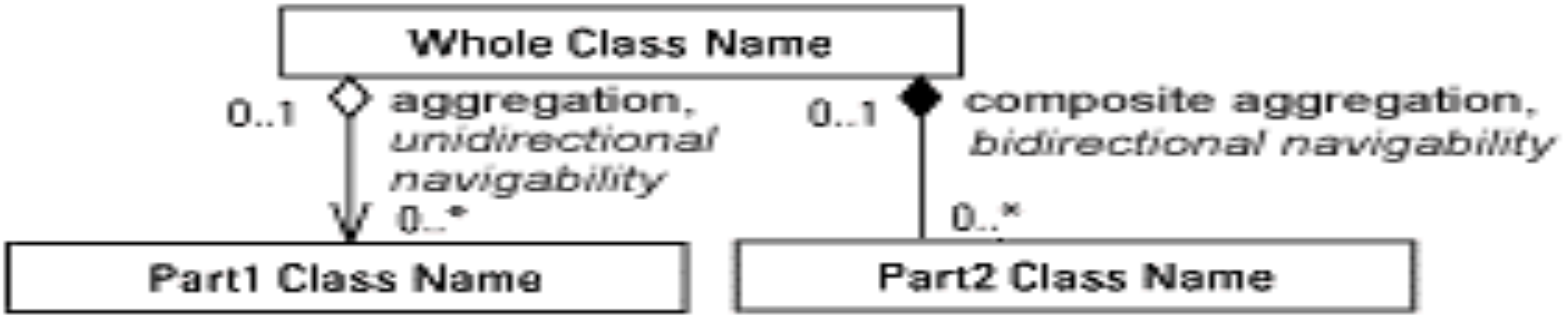
Association classes

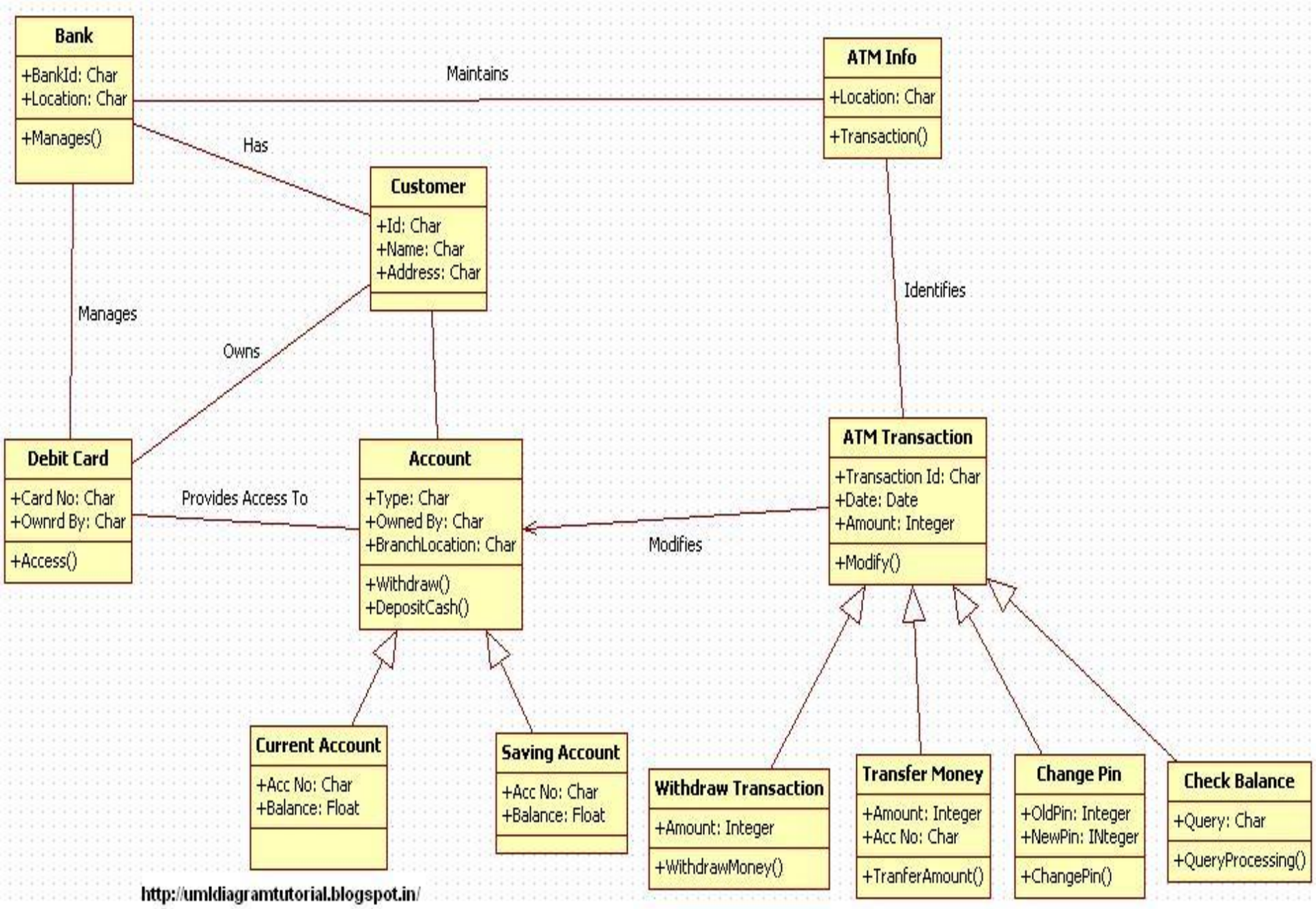


Role names and derived associations



Aggregation, navigability, and multiplicity



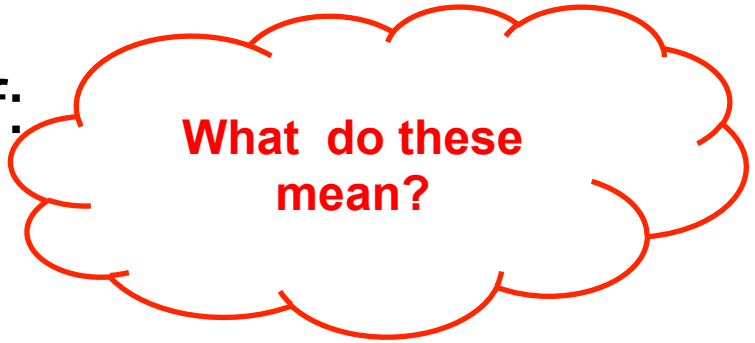


http://umldiagramtutorial.blogspot.in/

Process View

Process View

- Details the dynamic features and methods of the system
- Show the processes and threads used in system
  - Helps in analysis of:
    - Performance
    - Throughput
    - Scalability



Process View

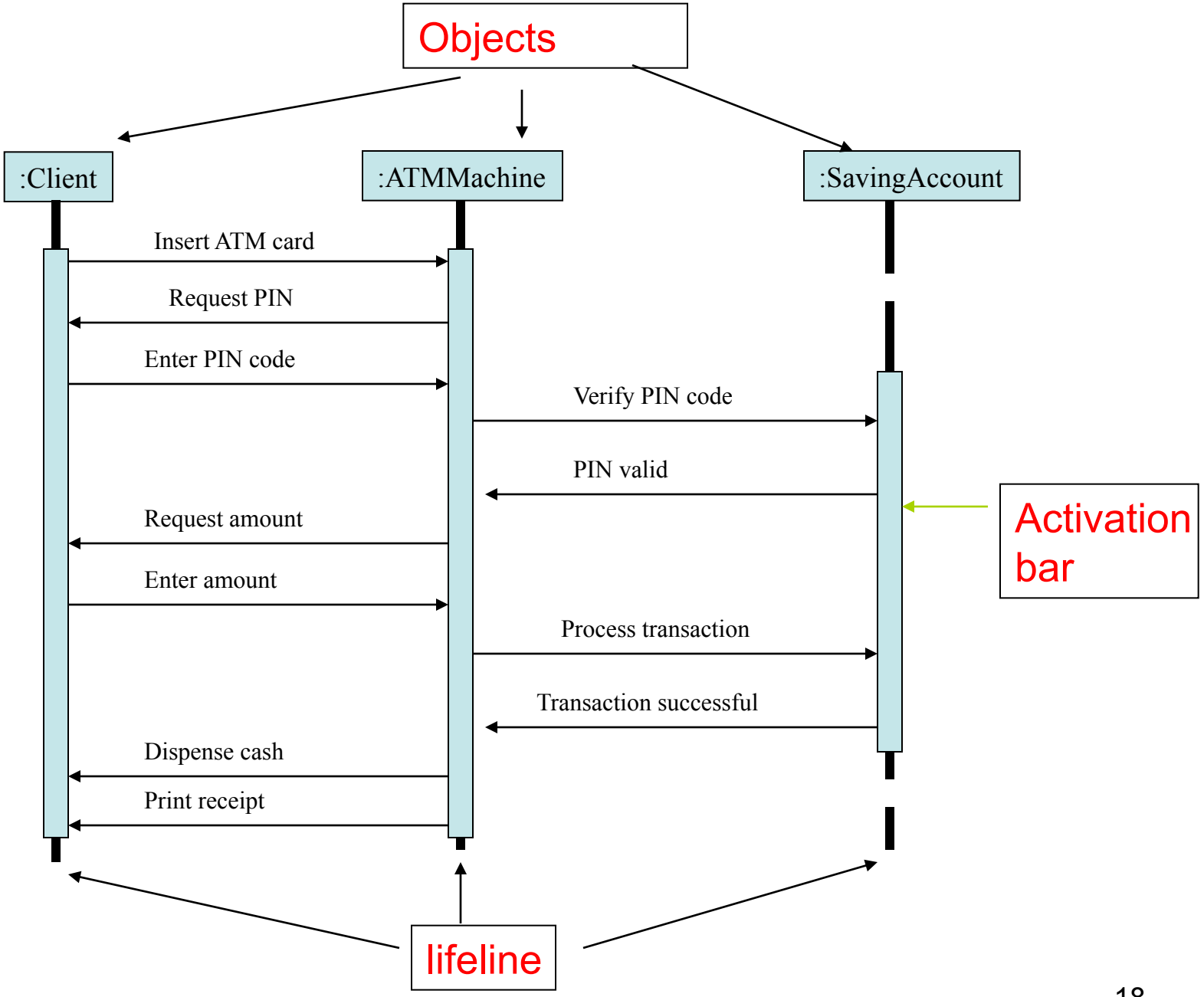
Models used

- Sequence Diagrams
  - - shows timing sequence of objects
  - - interactions between objects
- Activity Diagrams
- - show methods, transitions, flow of processing
- State Chart Diagrams

  - show state of an object during its lifetime as it responds to events, conditions and actions
- Collaboration Diagrams
- - show messages sent between objects. Numbering system used to show sequence

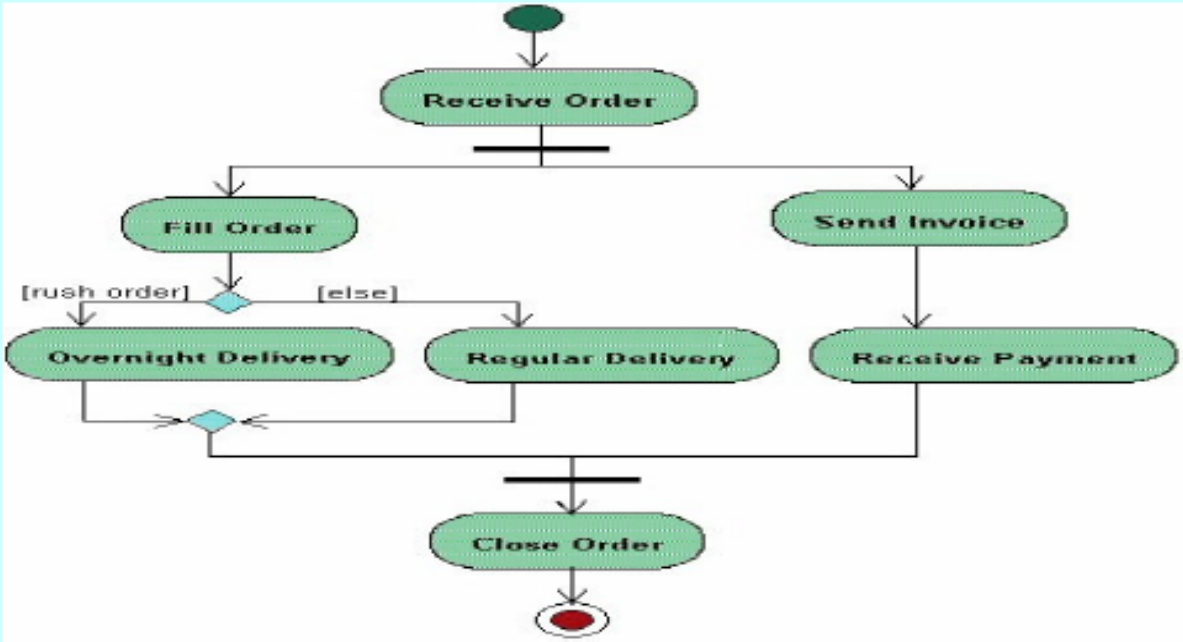
Sequence Diagrams

- Describing the behavior of a system by viewing the interaction between the system and its environment
- Shows the objects participating in the interaction by their life lines and the messages they exchange
- Shows an interaction arranged in a time sequence


Objects
:Client
:ATMMachine
:SavingAccount
Insert ATM card
Request PIN
Enter PIN code
Verify PIN code
PIN valid
Request amount
Enter amount
Process transaction
Transaction successful
Dispense cash
Print receipt
Activation bar
lifeline

Activity Diagram

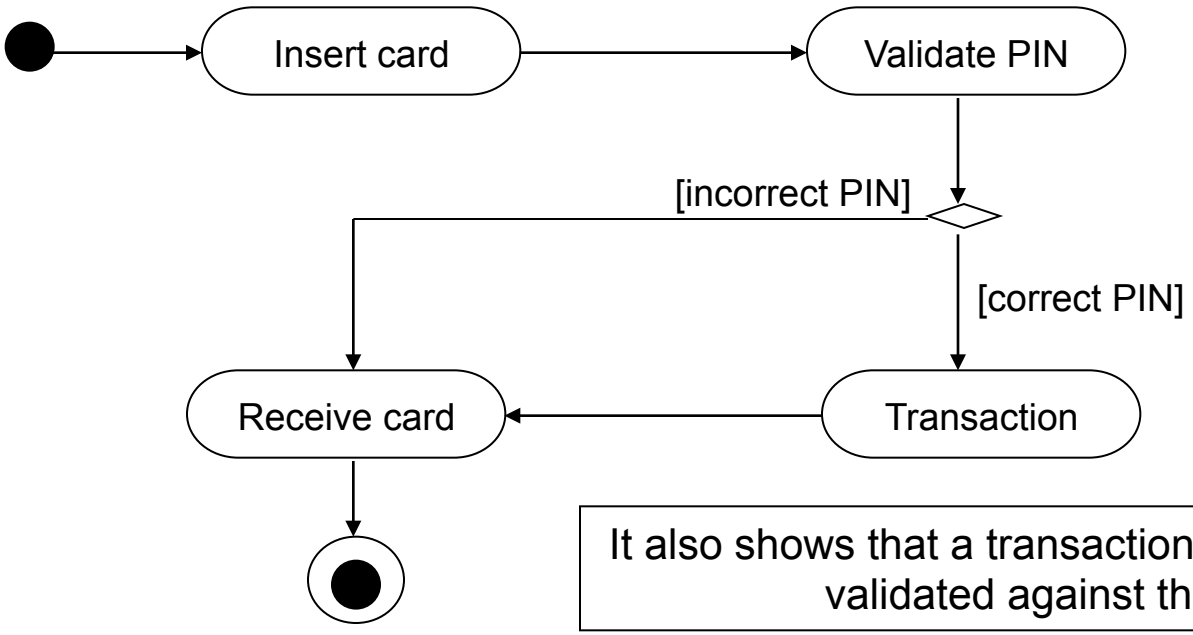
- UML version of a flowchart
- Help to understand the problem



Activity diagrams - ATM

Insert card & validate PIN:

Inserts card; validates the PIN entered by the user against the central record held on the bank system. Returns an output indicating that the PIN is either correct or incorrect. If the PIN is correct then the session can continue. If the PIN is not correct then the session will terminate, and the card should be returned to the user.

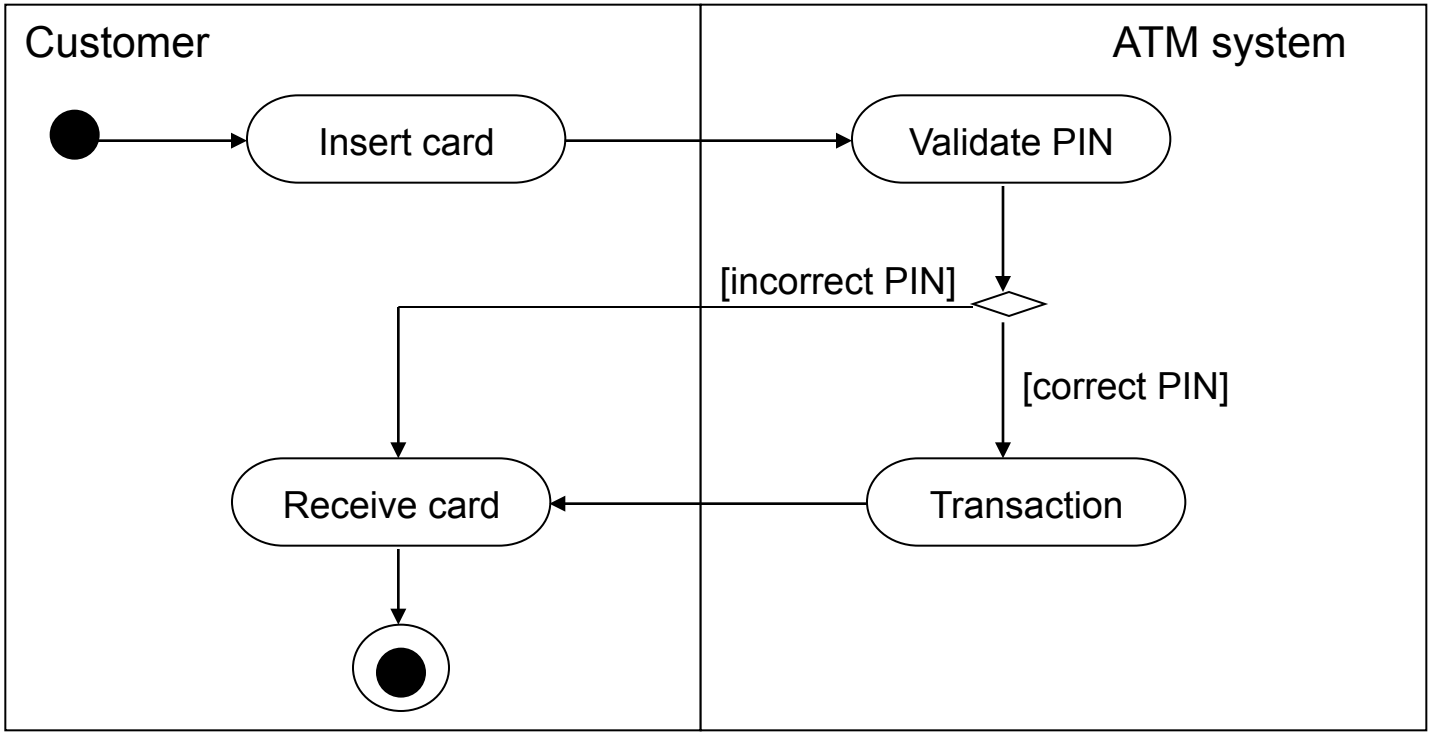


It also shows that a transaction is executed after the PIN is validated against the central record.

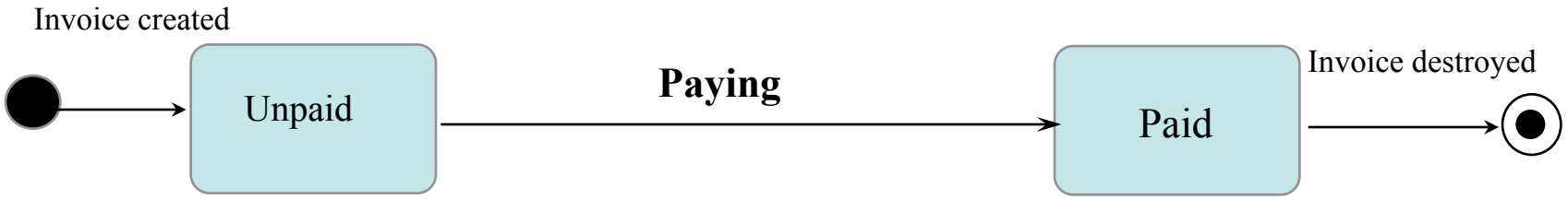
Activity Diagram - swimlanes

When activity diagrams contain several groups of related activities (belong to the same objects, use cases etc) they can be split into partitions known as *swimlanes*.

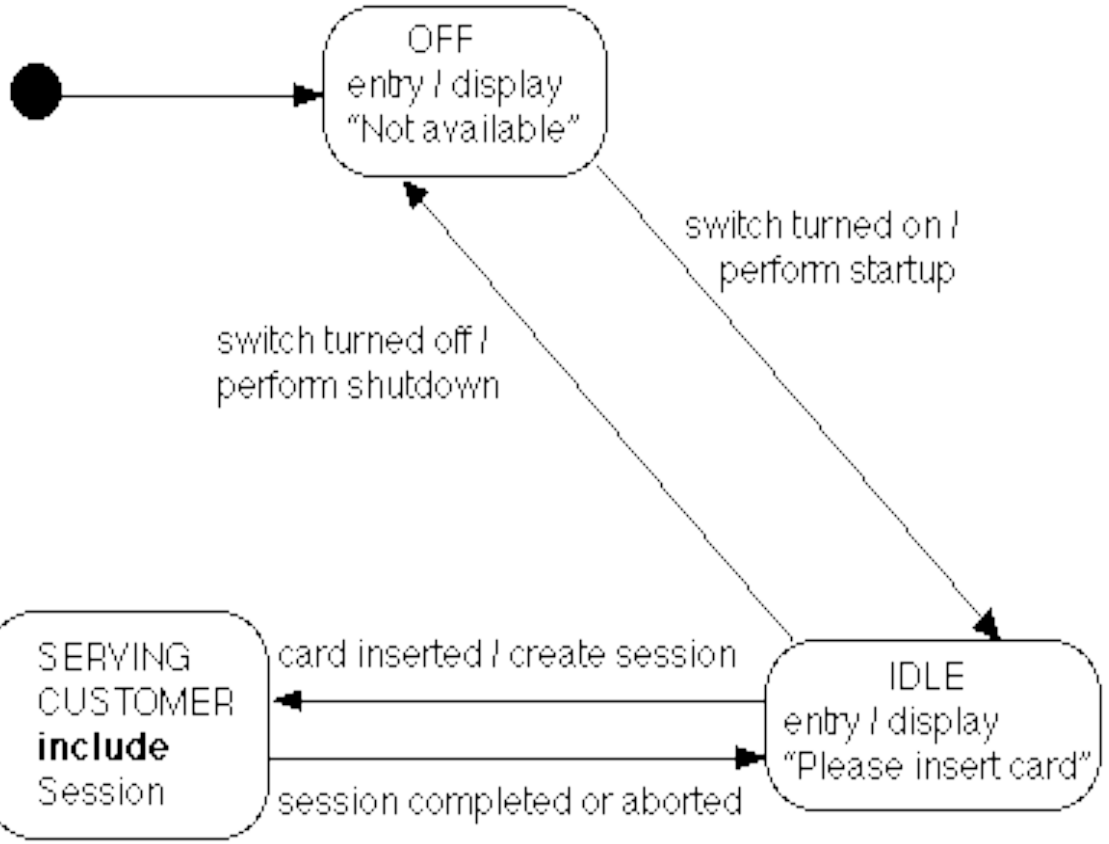
ATM – Insert card & validate PIN:



State Diagram Example

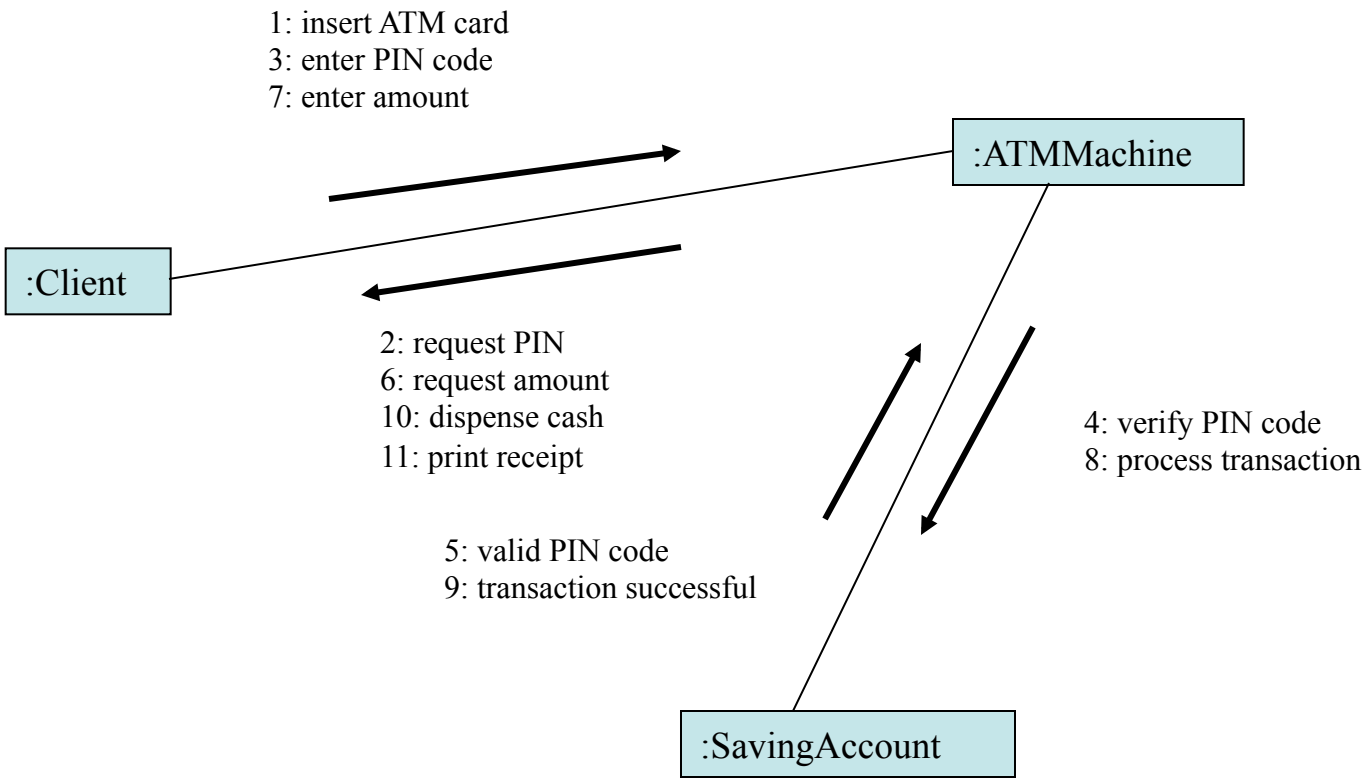


State-Chart for Overall ATM (includes System Startup and System Shutdown Use Cases)



Collaboration Diagrams

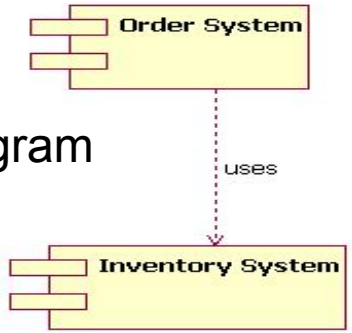
- Represents a collaboration, which is a set of objects related in a particular context, and interaction, which is a set of messages exchanged among the objects within the collaboration to achieve a desired outcome
- Same objects as sequence diagram
- Sequences of messages are shown by numbering


1: insert ATM card
3: enter PIN code
7: enter amount
:ATMMachine
:Client
2: request PIN
6: request amount
10: dispense cash
11: print receipt
4: verify PIN code
8: process transaction
5: valid PIN code
9: transaction successful
:SavingAccount

Implementation View

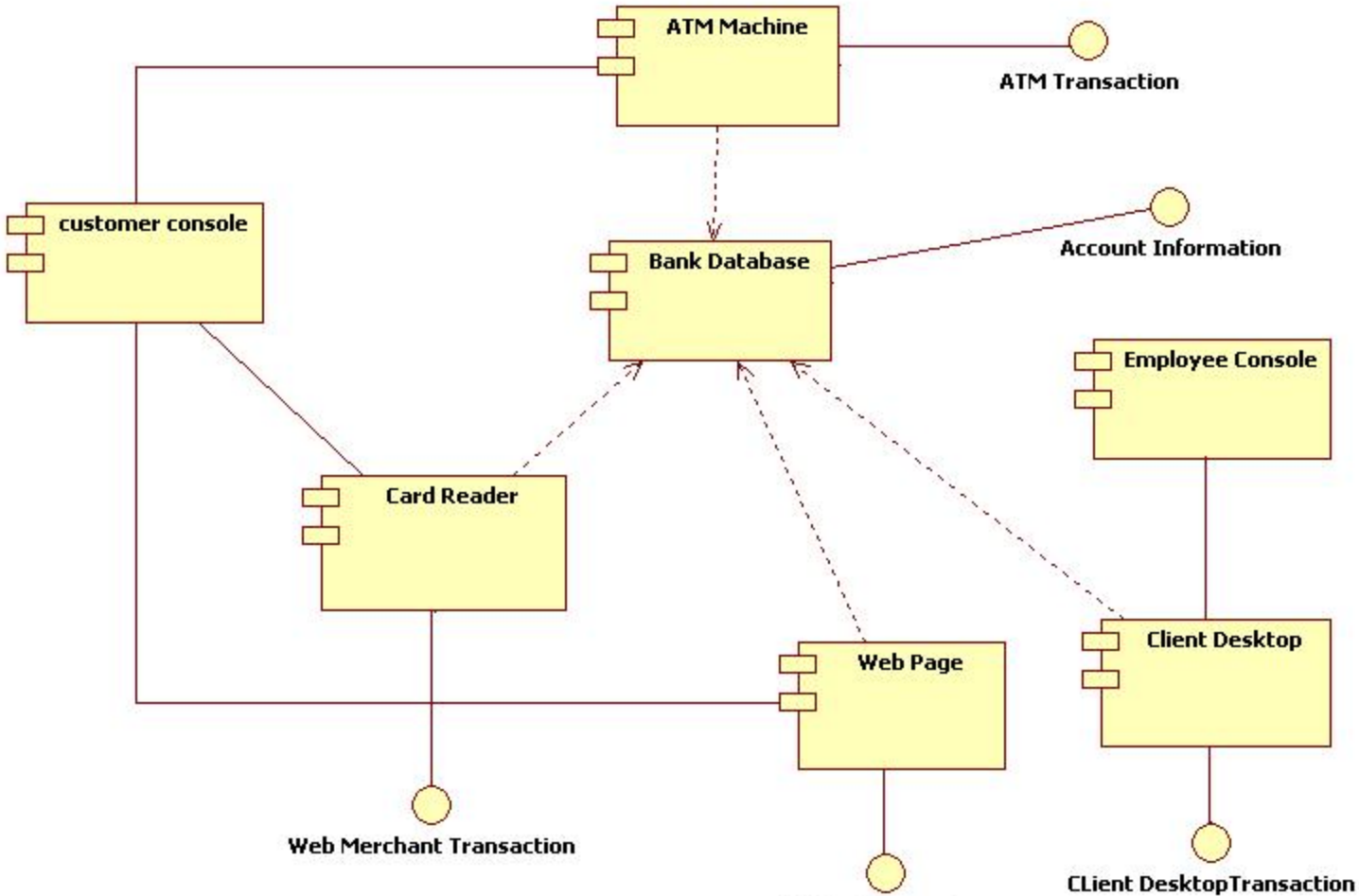
Model used

Component Diagram



- Details components and files used to assemble and release the physical system
- Use component diagrams which show the organisation of source code
- Configuration Management of the system release

Component Diagram – ATM



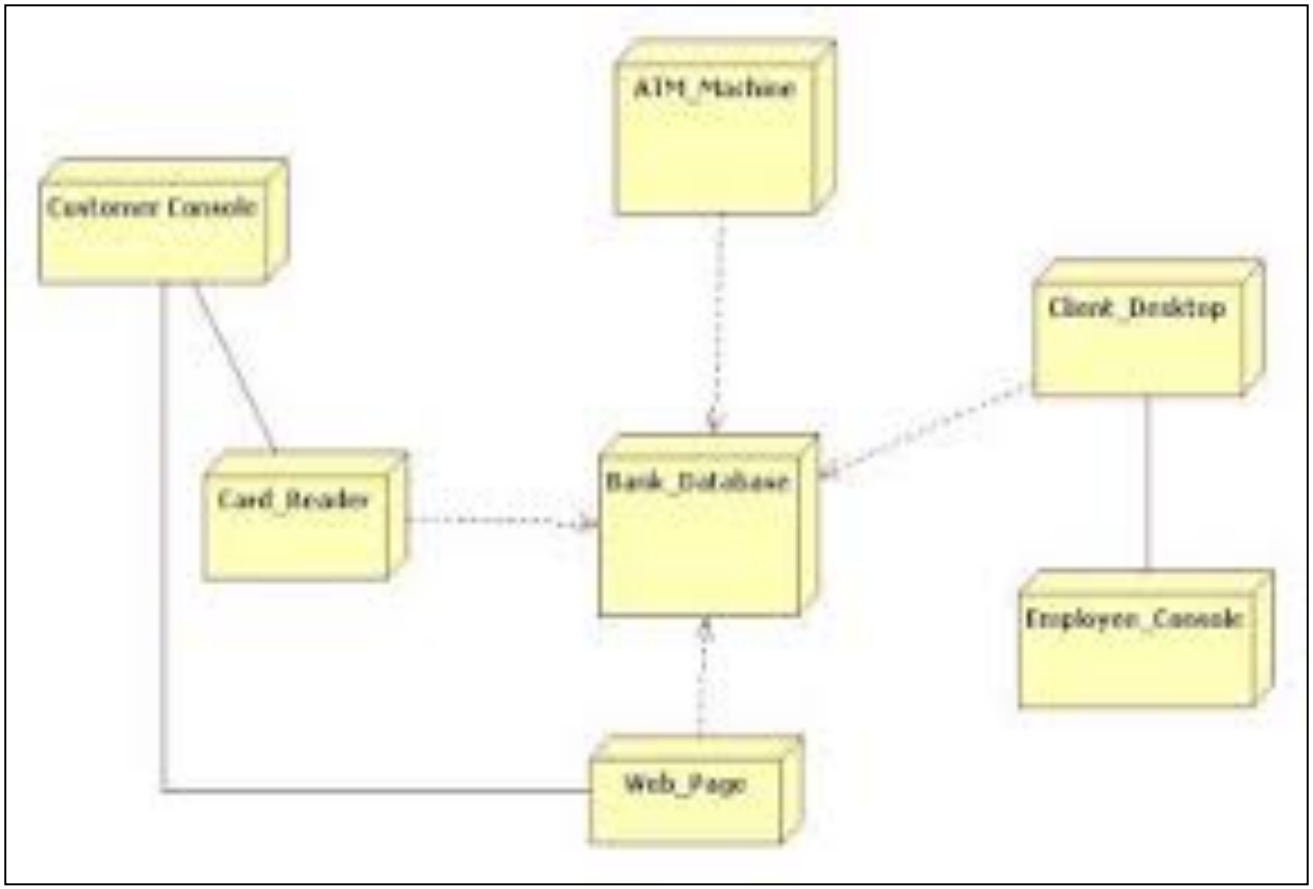
Deployment View

Deployment View

- Topology of system hardware on which system executes

- Concerned with:
- ---- Delivery
- ---- Distribution
- ---- Installation of parts that make up the physical system

Deployment Diagram - ATM



Dynamic V Static Diagrams

There are *about* 9 types of diagrams in UML1.5