10

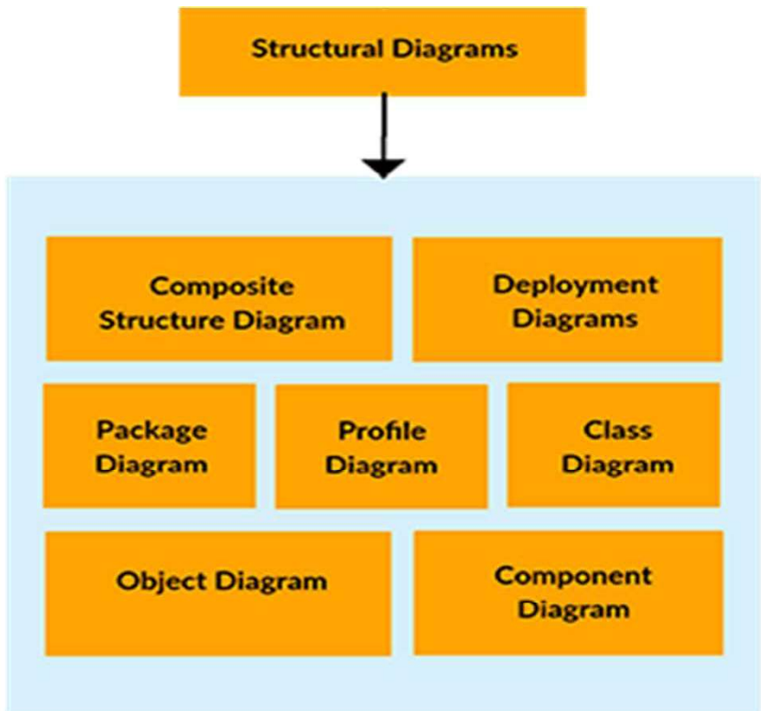
UML Structured and Behavioral Diagrams

Objectives

After completing this lesson, you should be able to do the following:

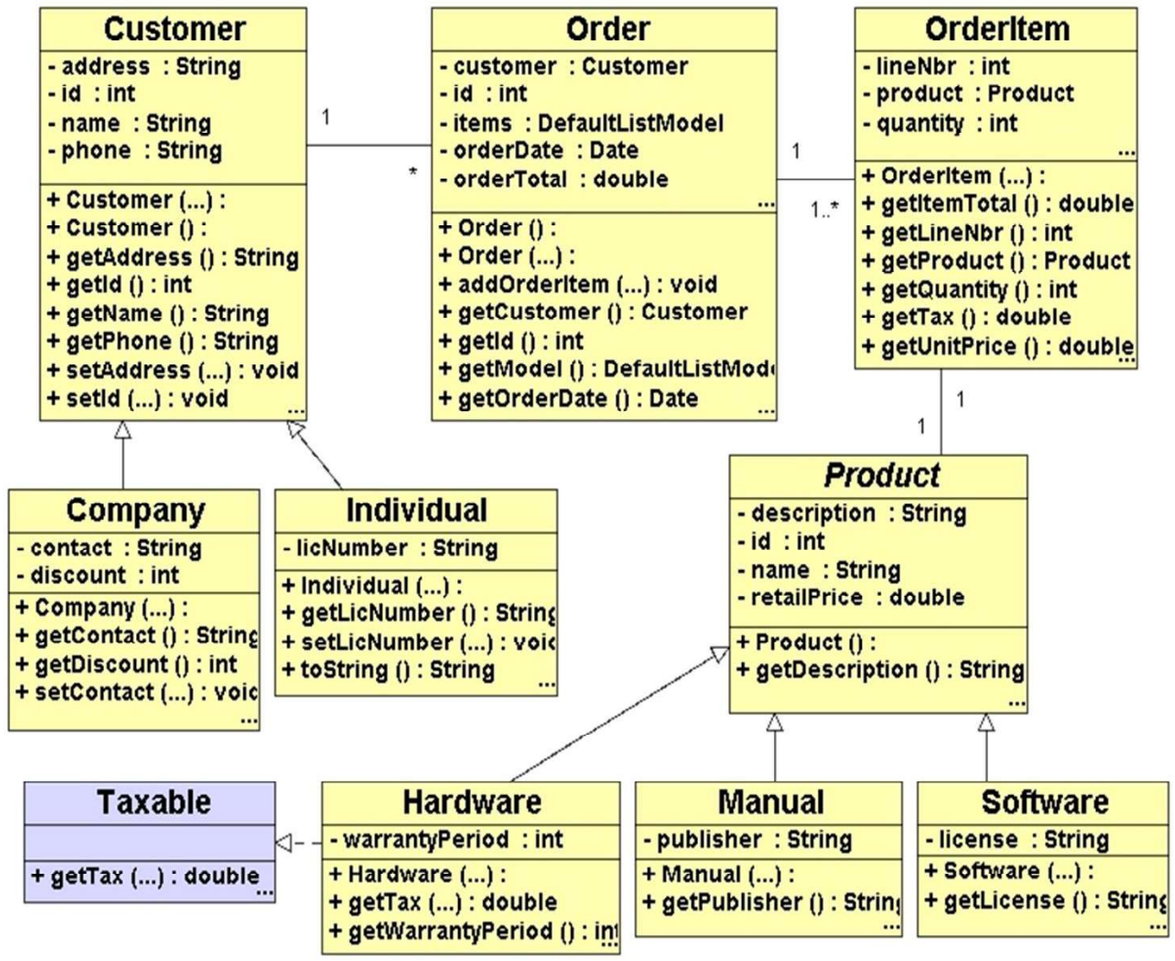
- Explore different types of Structural Diagrams
- Explore different types of Behavioral Diagrams

UML structural diagrams are categorized as follows:

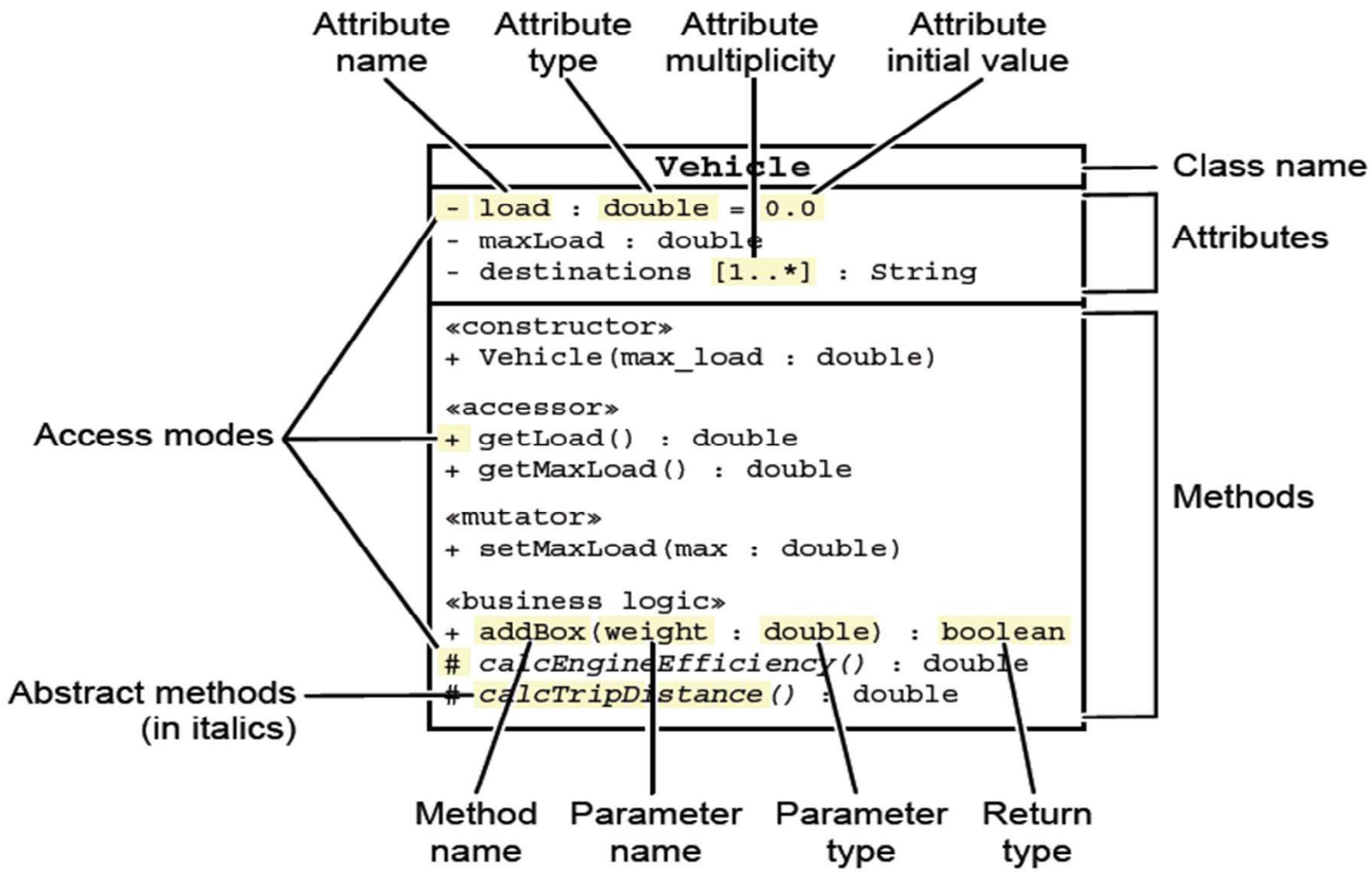


Class Diagram

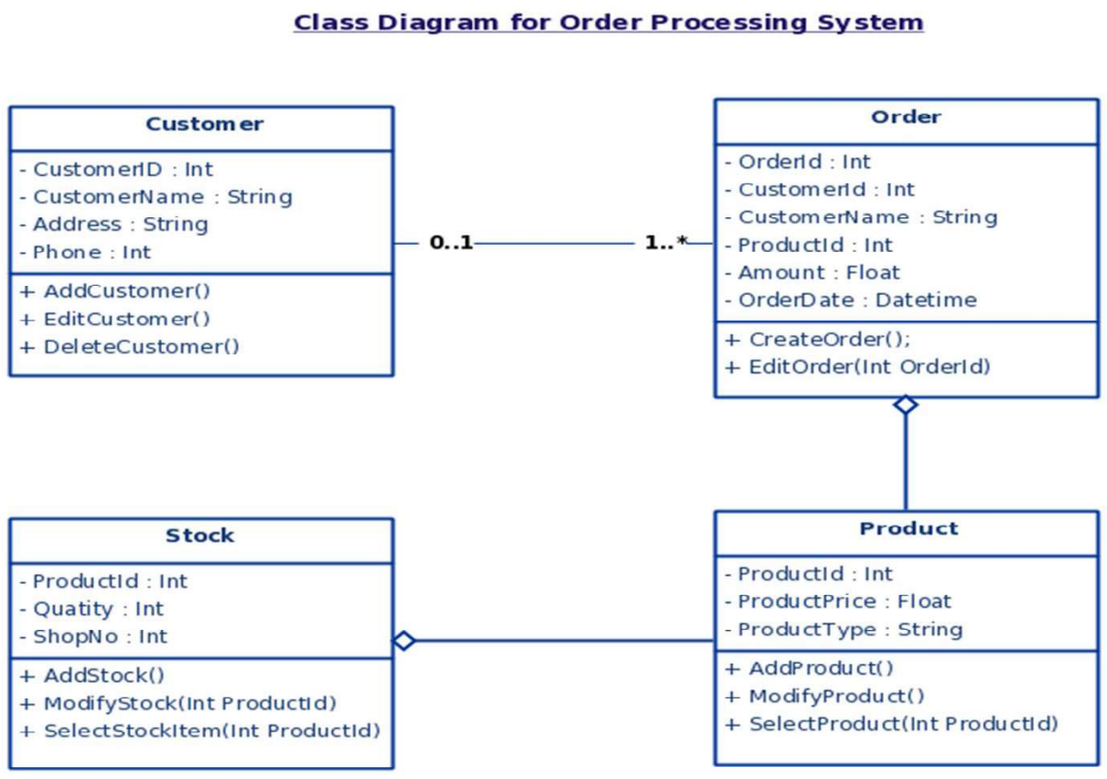
- A class diagram models the static view of a system. It comprises of the classes, interfaces, and collaborations of a system; and the relationships between them.



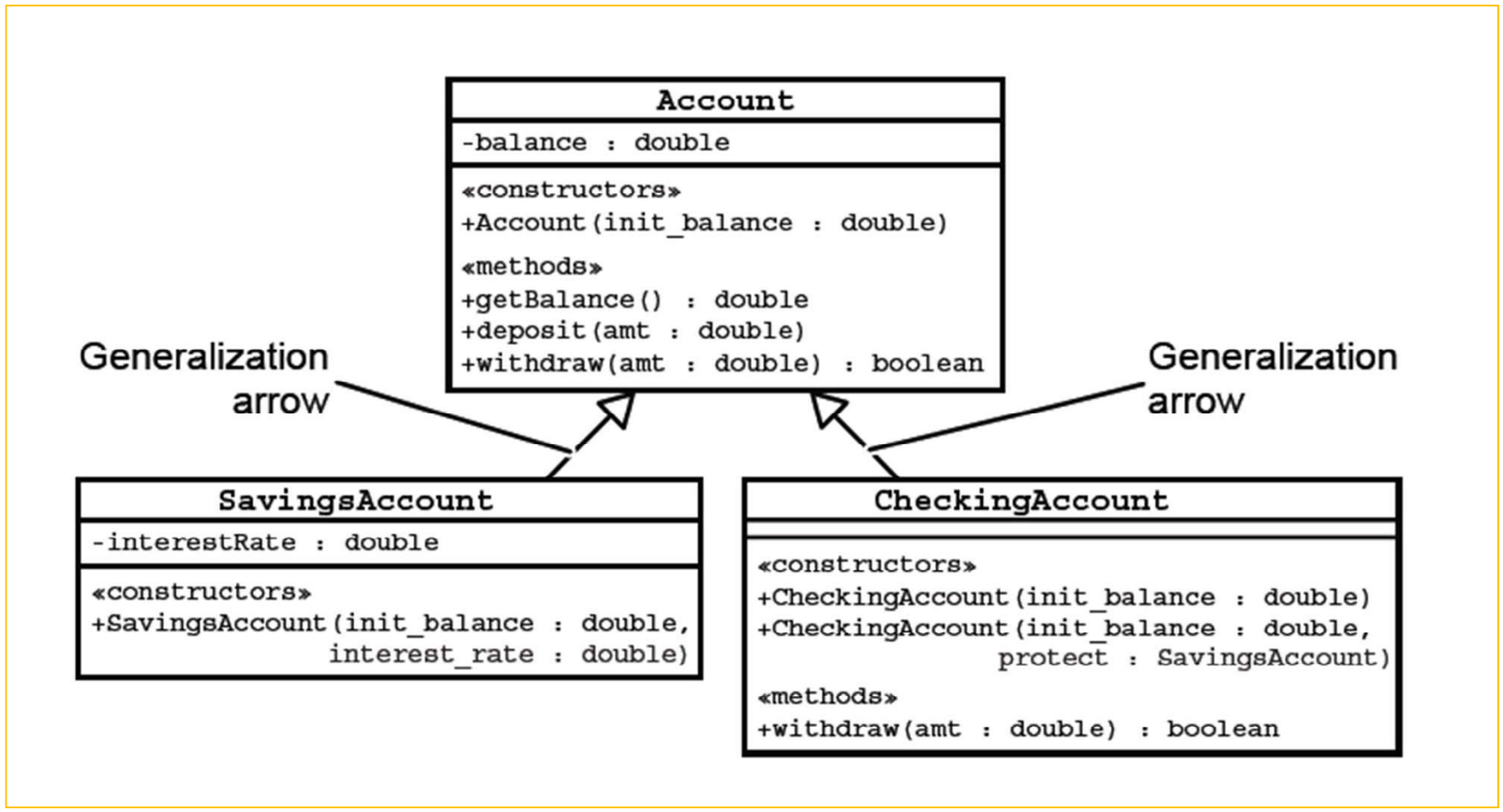
Example


Attribute name
Attribute type
Attribute multiplicity
Attribute initial value
Vehicle
Class name
- load : double = 0.0
- maxLoad : double
- destinations [1..*] : String
Attributes
«constructor»
+ Vehicle(max_load : double)
«accessor»
+ getLoad() : double
+ getMaxLoad() : double
«mutator»
+ setMaxLoad(max : double)
«business logic»
+ addBox(weight : double) : boolean
# calcEngineEfficiency() : double
# calcTripDistance() : double
Methods
Access modes
Abstract methods (in italics)
Method name
Parameter name
Parameter type
Return type

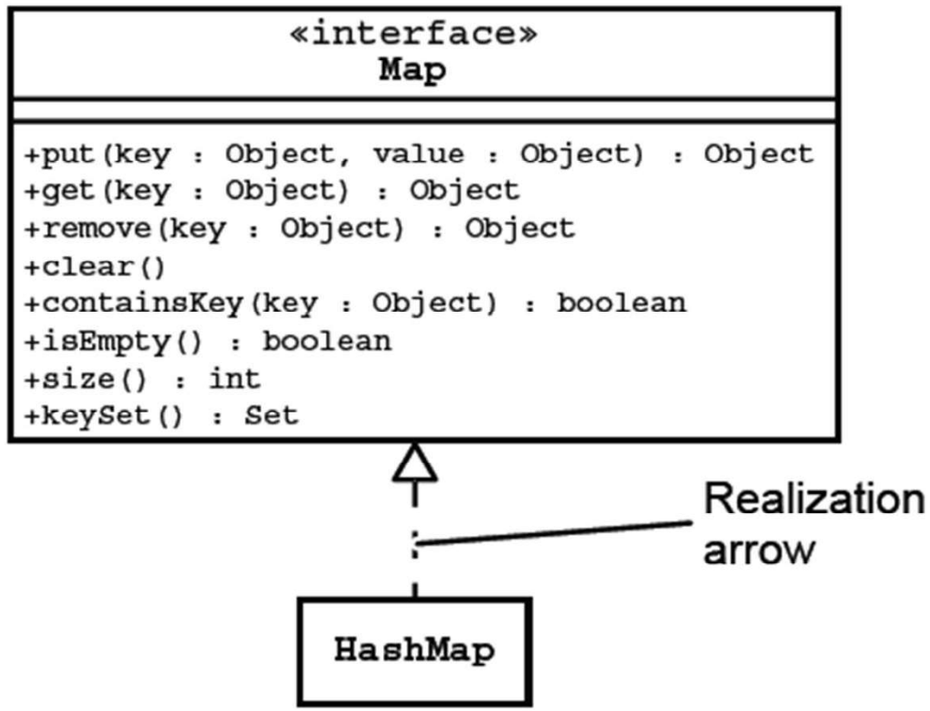
Example :


Class Diagram for Order Processing System
Customer
- CustomerID : Int
- CustomerName : String
- Address : String
- Phone : Int
+ AddCustomer()
+ EditCustomer()
+ DeleteCustomer()
0..1
1..*
Order
- OrderId : Int
- CustomerId : Int
- CustomerName : String
- ProductId : Int
- Amount : Float
- OrderDate : Datetime
+ CreateOrder();
+ EditOrder(Int OrderId)
Stock
- ProductId : Int
- Quatity : Int
- ShopNo : Int
+ AddStock()
+ ModifyStock(Int ProductId)
+ SelectStockItem(Int ProductId)
Product
- ProductId : Int
- ProductPrice : Float
- ProductType : String
+ AddProduct()
+ ModifyProduct()
+ SelectProduct(Int ProductId)

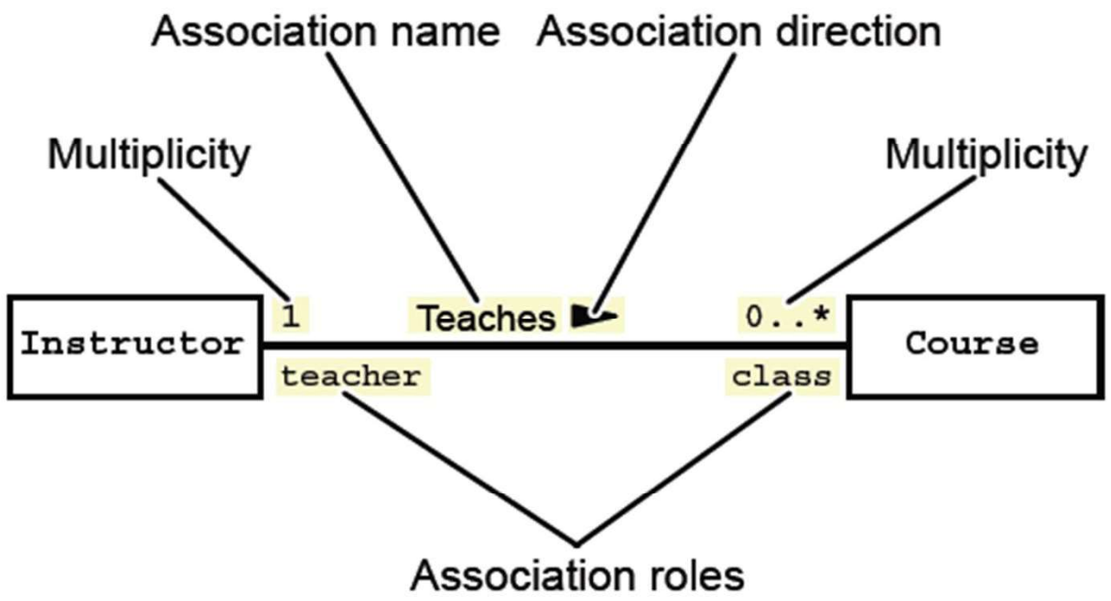
Inheritance



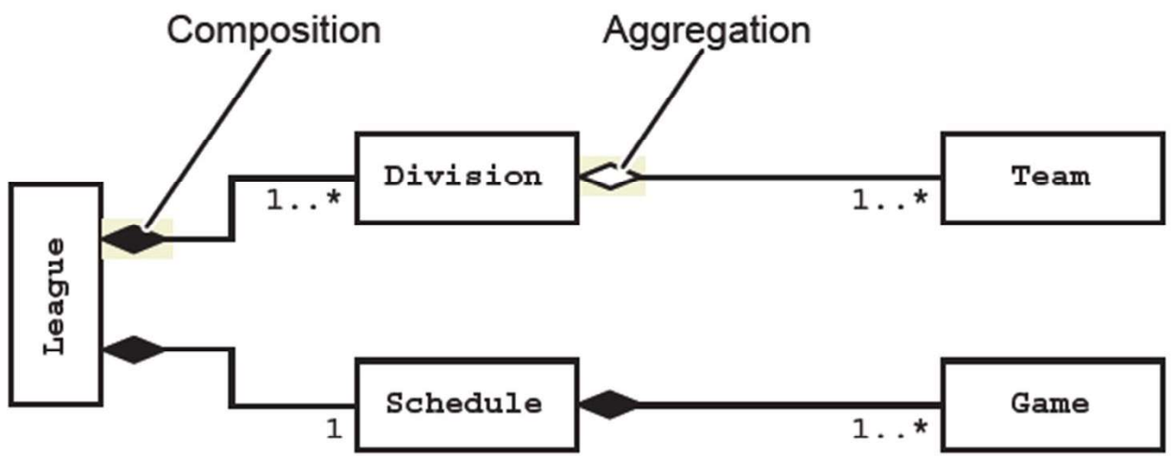
Interface Implementation



Association, Roles, and Multiplicity

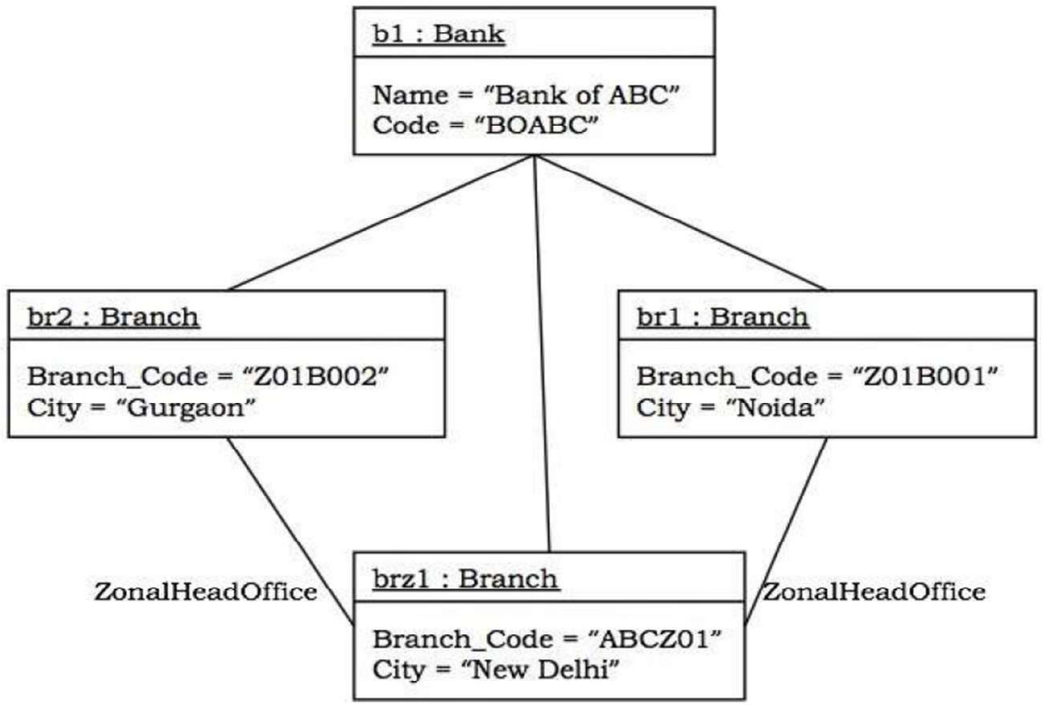


Aggregation and Composition

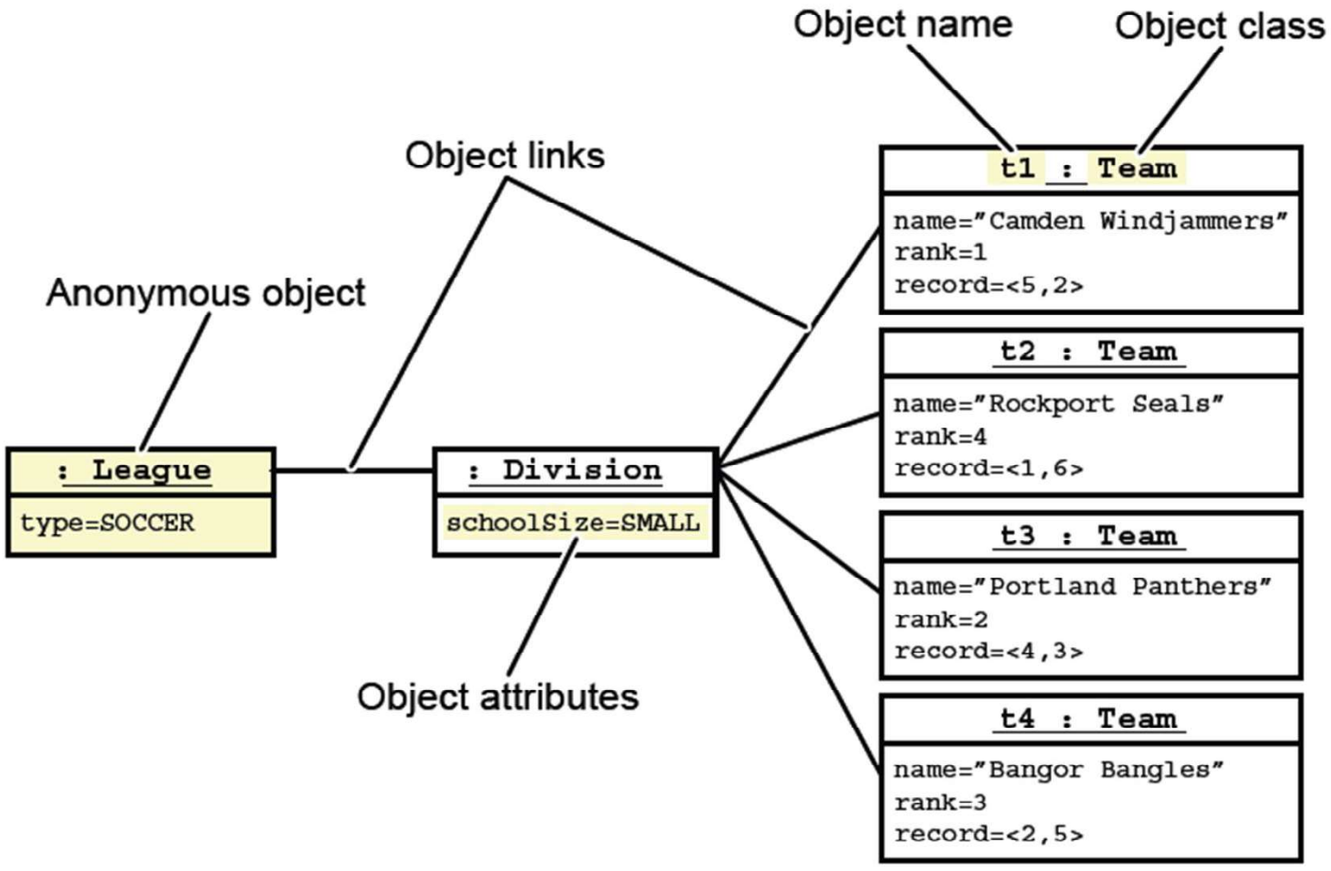


Object Diagram

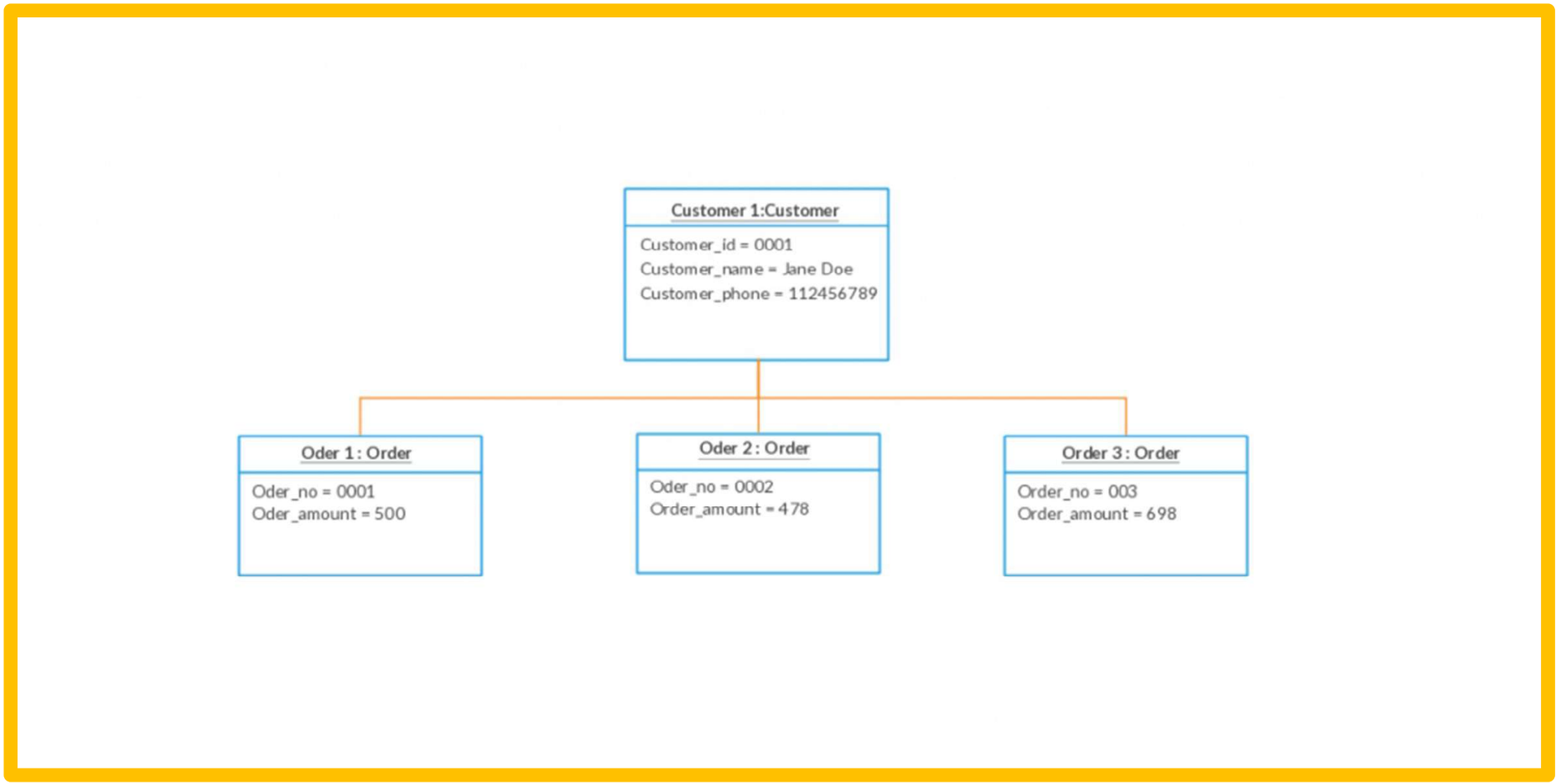
- An object diagram models a group of objects and their links at a point of time. It shows the instances of the things in a class diagram. Object diagram is the static part of an interaction diagram.



Example


Object name
Object class
Object links
Anonymous object
: League
type=SOCCER
: Division
schoolSize=SMALL
Object attributes
t1 : Team
name="Camden Windjammers"
rank=1
record=<5,2>
t2 : Team
name="Rockport Seals"
rank=4
record=<1,6>
t3 : Team
name="Portland Panthers"
rank=2
record=<4,3>
t4 : Team
name="Bangor Bangles"
rank=3
record=<2,5>

Example

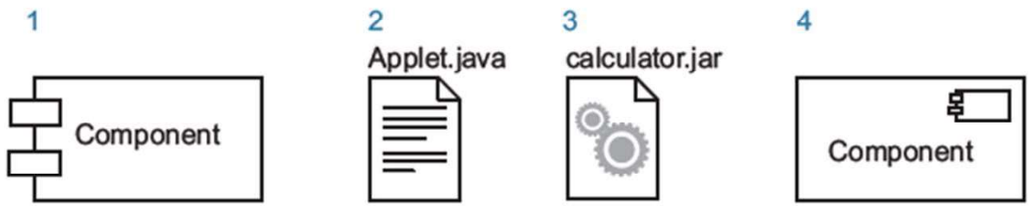

Customer 1:Customer
Customer_id = 0001
Customer_name = Jane Doe
Customer_phone = 112456789
Oder 1 : Order
Oder_no = 0001
Oder_amount = 500
Oder 2 : Order
Oder_no = 0002
Order_amount = 478
Order 3 : Order
Order_no = 003
Order_amount = 698

Component Diagram

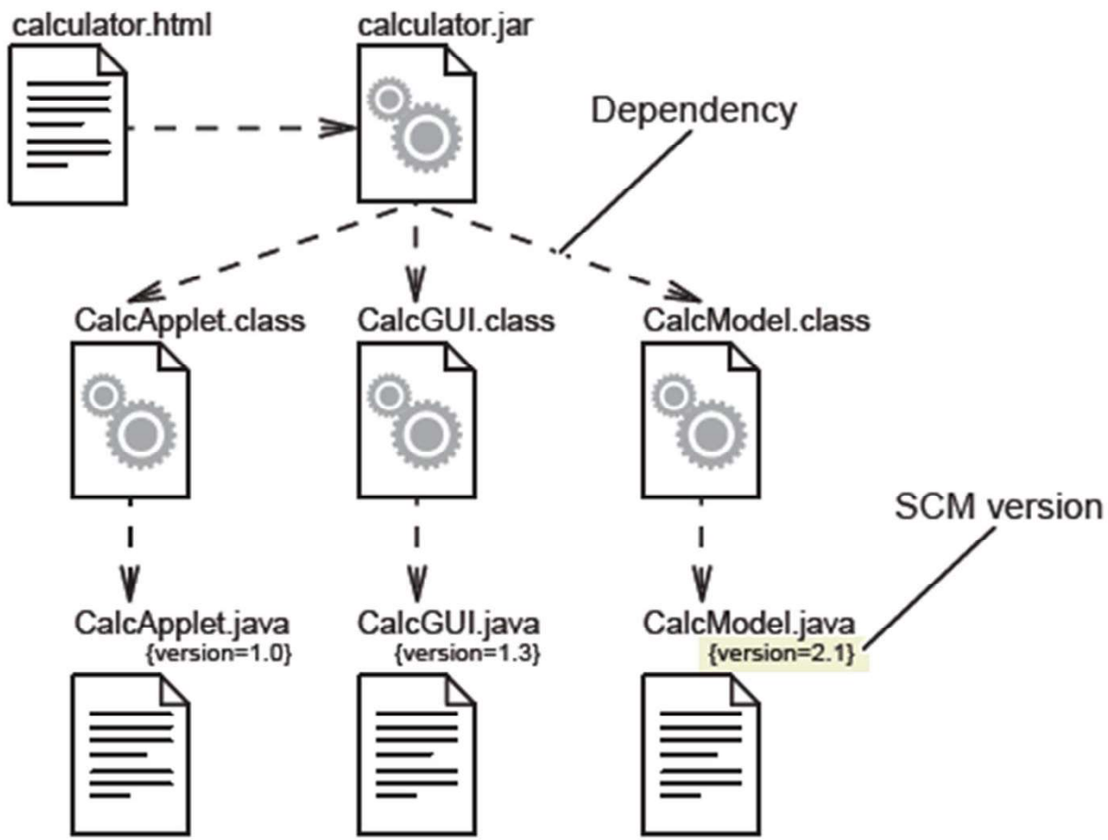
- Component diagrams show the organization and dependencies among a group of components.
- Component diagrams comprise of:
  1. Components
  2. Interfaces
  3. Relationships
  4. Packages and Subsystems *optional*

Component diagrams are used for

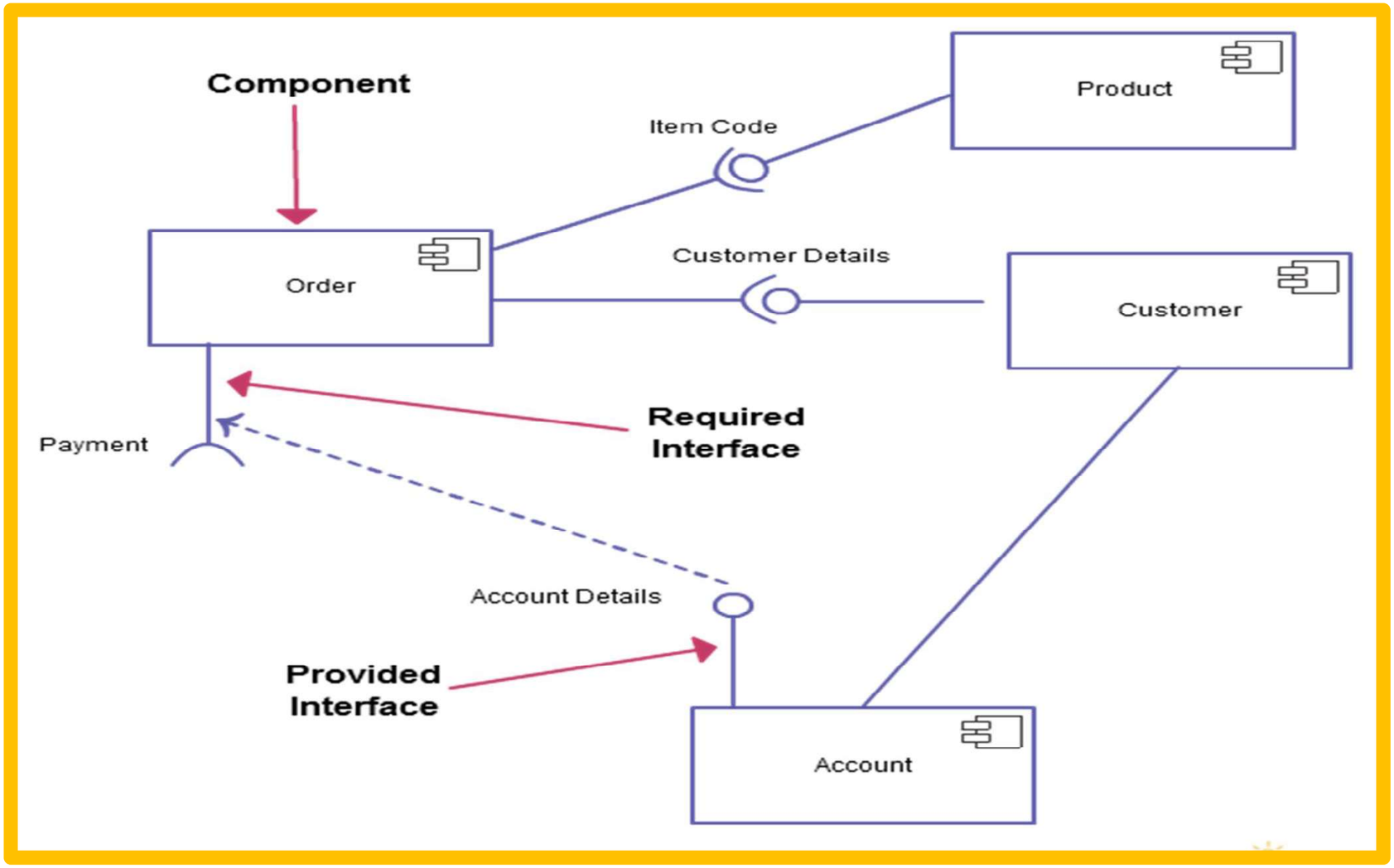
- Constructing systems through forward and reverse engineering.
- Modeling configuration management of source code files while developing a system using an object-oriented programming language.
- Representing schemas in modeling databases.
- Modeling behaviors of dynamic systems.



Example


calculator.html
calculator.jar
Dependency
CalcApplet.class
CalcGUI.class
CalcModel.class
SCM version
CalcApplet.java
{version=1.0}
CalcGUI.java
{version=1.3}
CalcModel.java
{version=2.1}

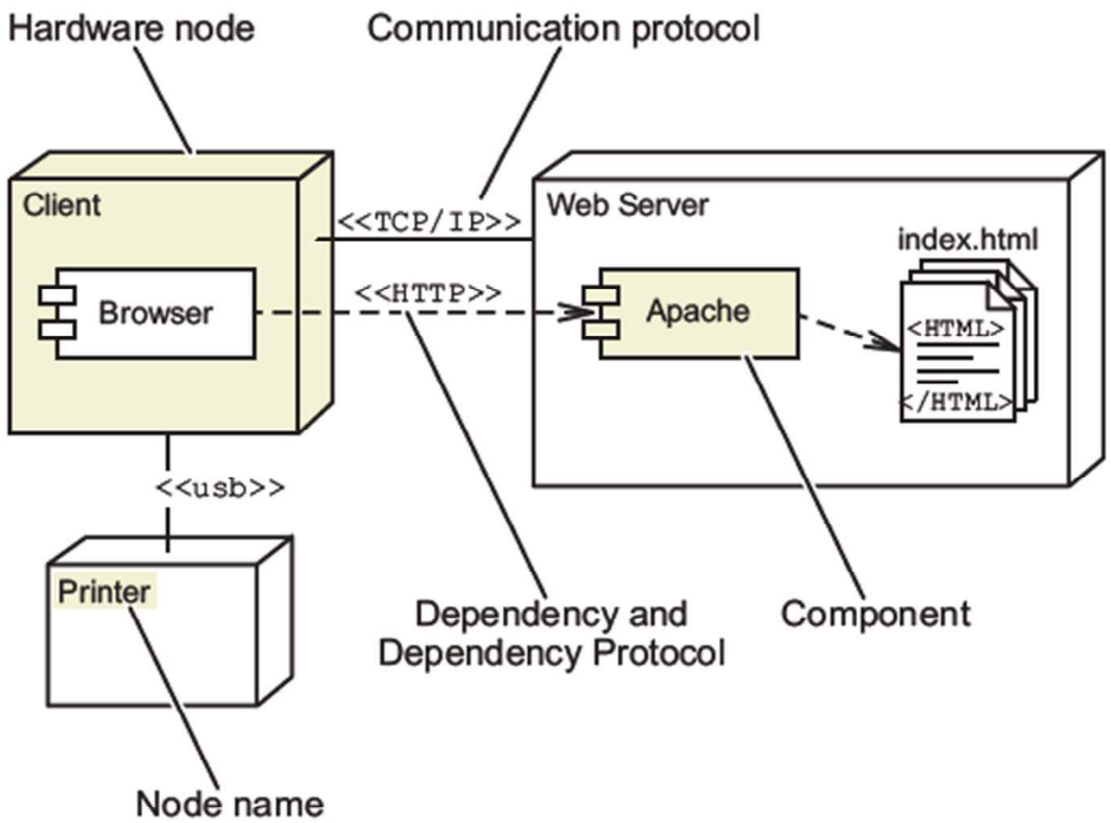
Example


Component
Product
Item Code
Order
Customer Details
Customer
Payment
Required
Interface
Account Details
Provided
Interface
Account

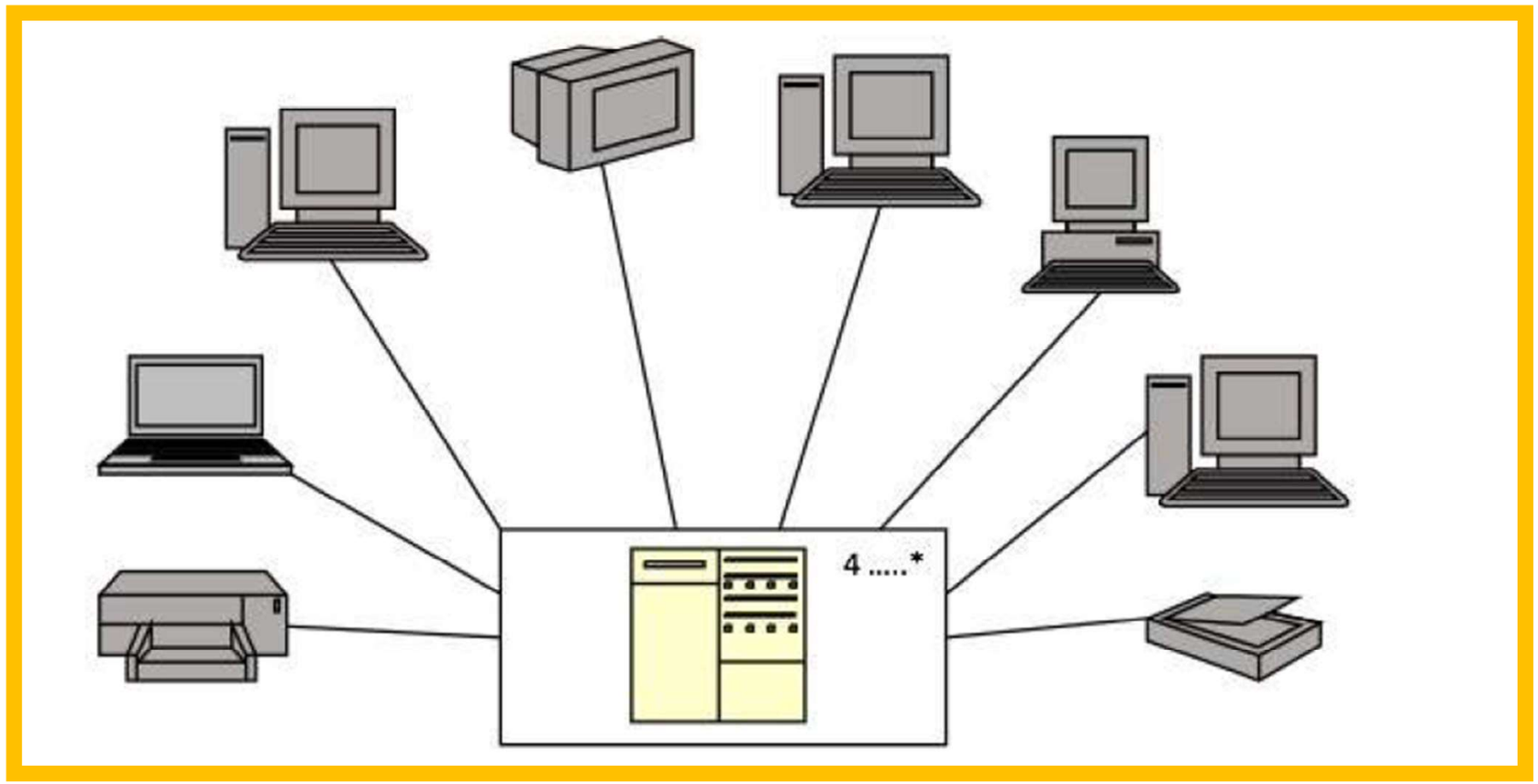
Deployment Diagram

- A deployment diagram puts emphasis on the configuration of runtime processing nodes and their components that live on them.
- They are commonly comprised of nodes and dependencies, or associations between the nodes
- Deployment diagrams are used to:
  1. Model devices in embedded systems that typically comprise of software-intensive collection of hardware.
  2. Represent the topologies of client/server systems.
  3. Model fully distributed systems.

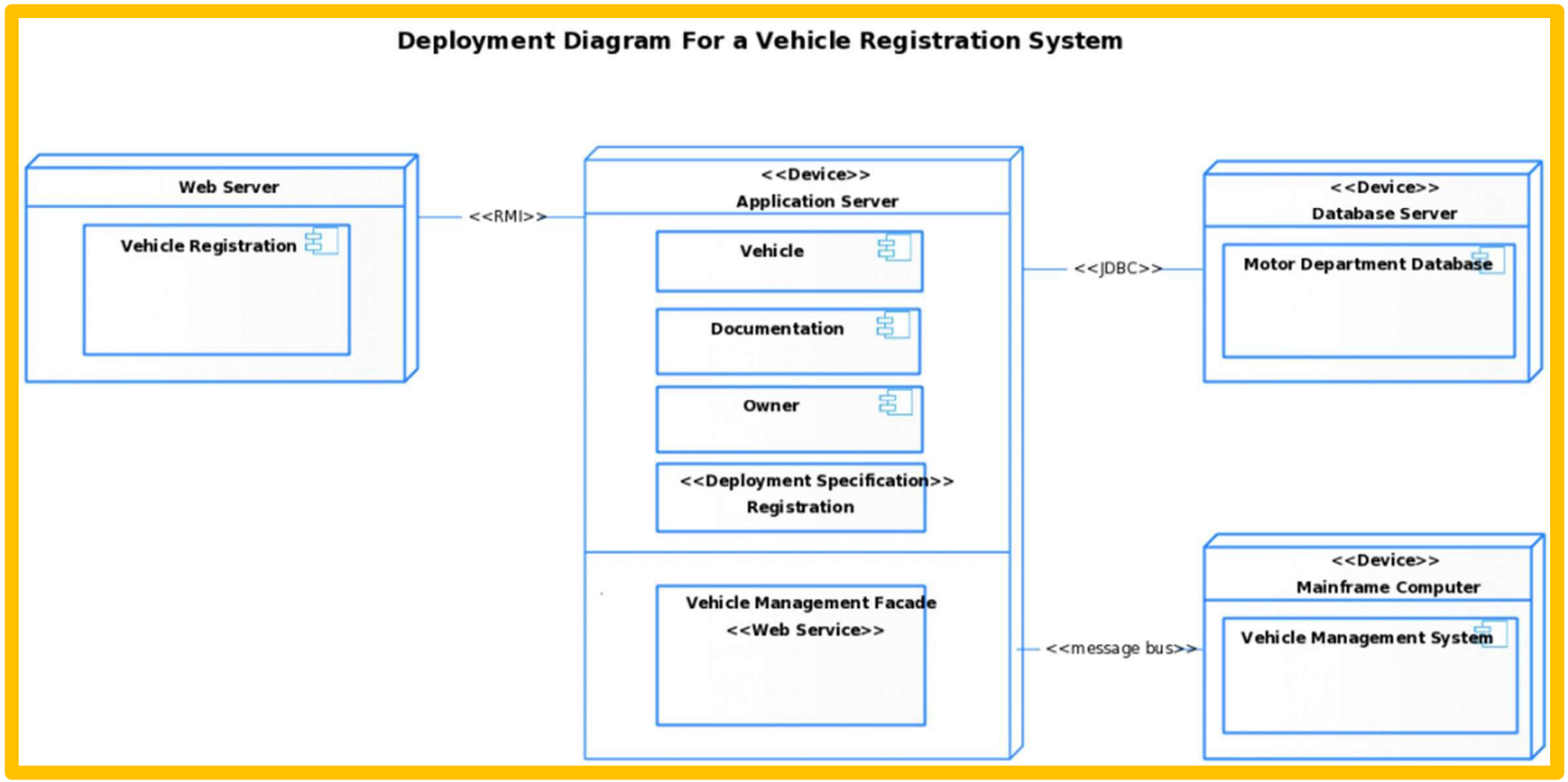
Example


Hardware node
Communication protocol
Client
<<TCP/IP>>
Web Server
index.html
Browser
<<HTTP>>
Apache
<HTML>
</HTML>
<<usb>>
Printer
Dependency and
Dependency Protocol
Component
Node name

Example

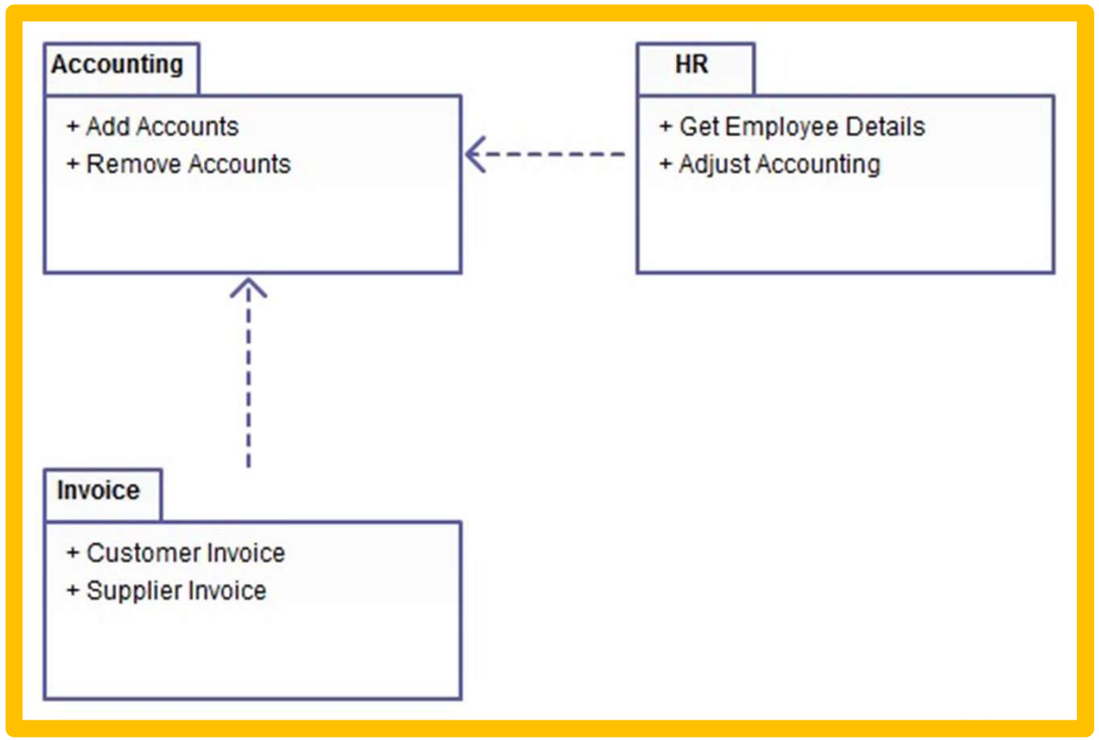


Example


Deployment Diagram For a Vehicle Registration System
Web Server
Vehicle Registration
<<RMI>>
<<Device>>
Application Server
Vehicle
Documentation
Owner
<<Deployment Specification>>
Registration
Vehicle Management Facade
<<Web Service>>
<<JDBC>>
<<Device>>
Database Server
Motor Department Database
<<message bus>>
<<Device>>
Mainframe Computer
Vehicle Management System

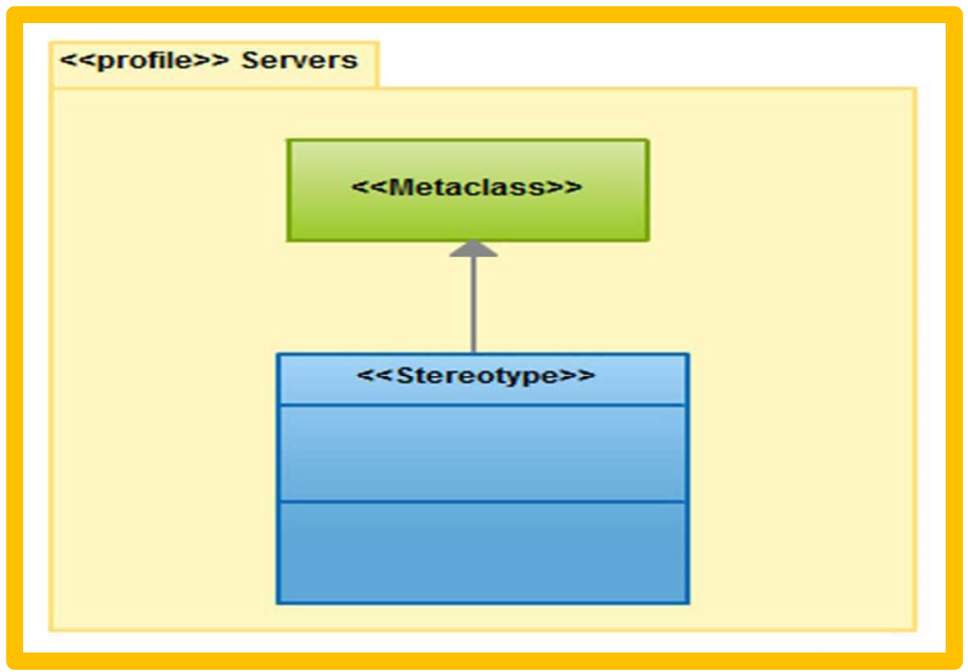
Package Diagram

- As the name suggests, a package diagram shows the dependencies between different packages in a system.



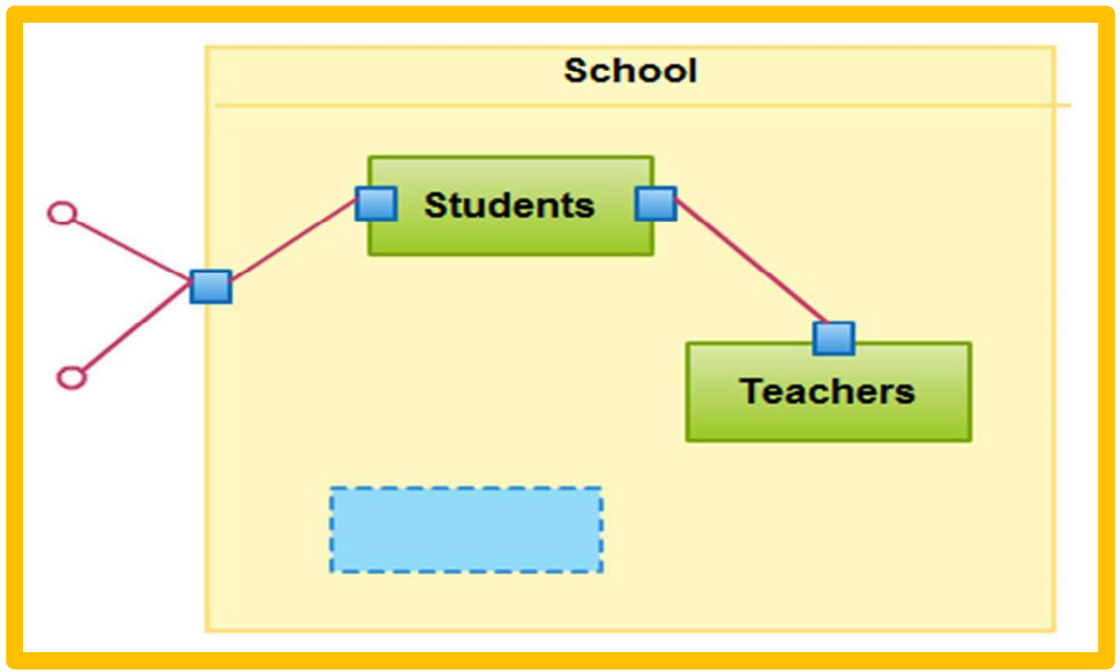
Profile Diagram

- Profile diagram is a new diagram type introduced in UML 2. This is a diagram type that is very rarely used in any specification.



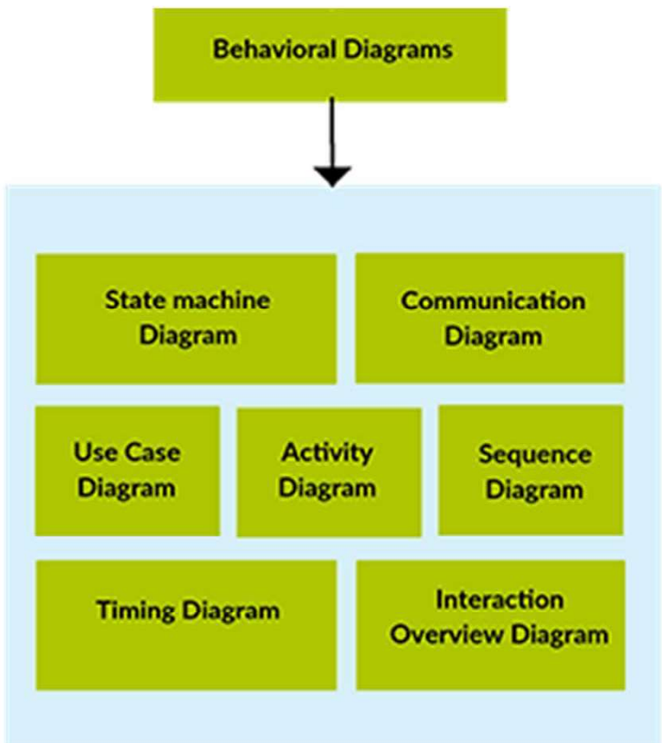
Composite Structure Diagram

- Composite structure diagrams are used to show the internal structure of a class.



Behavioral Diagram

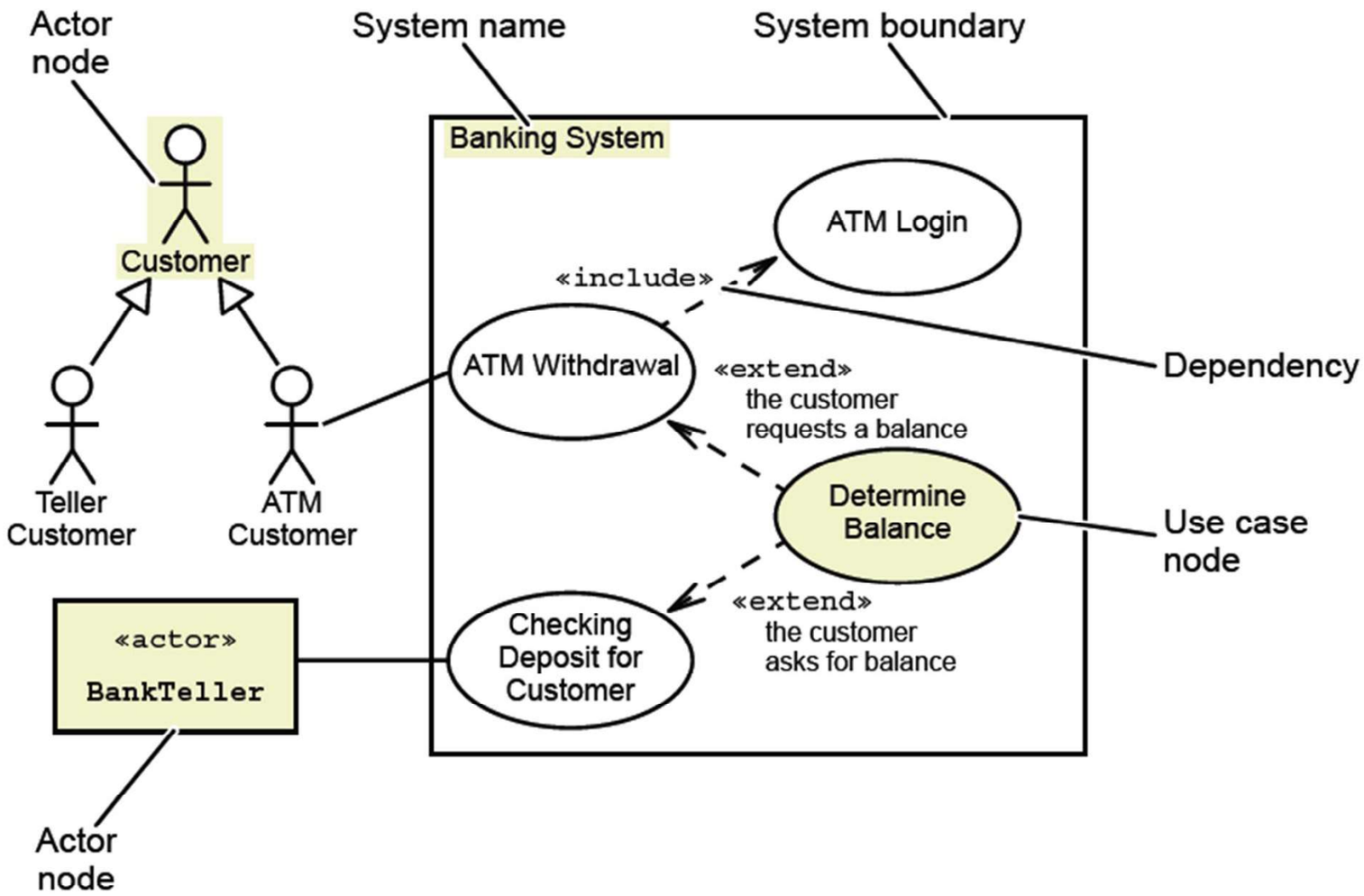
- UML Behavioral Diagram Consists of 7 Diagrams



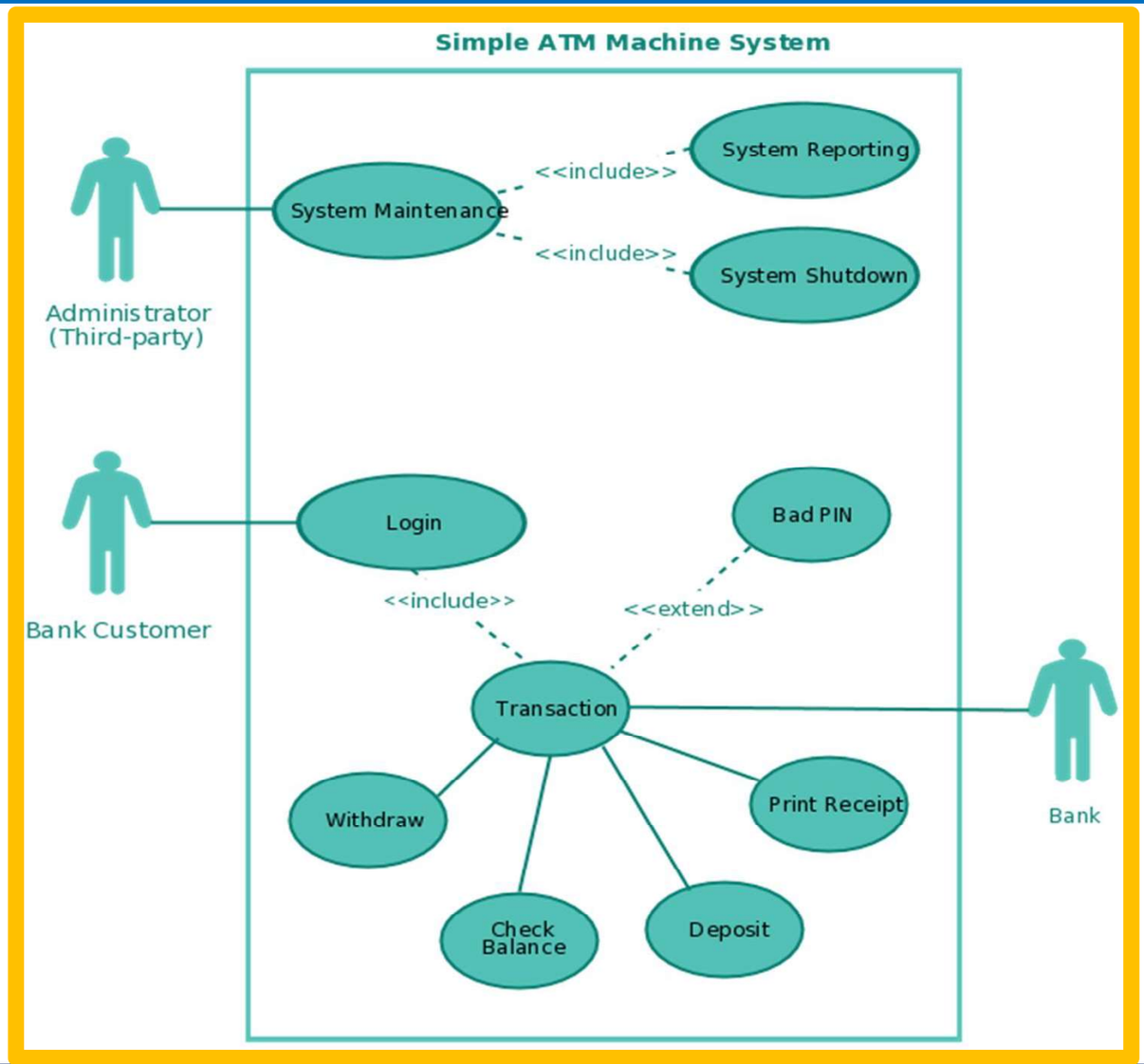
Use Case Diagram

- As the most known diagram type of the behavioral UML diagrams, Use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions are interacted.
- It's a great starting point for any project discussion, because you can easily identify the main actors involved and the main processes of the system.

Example

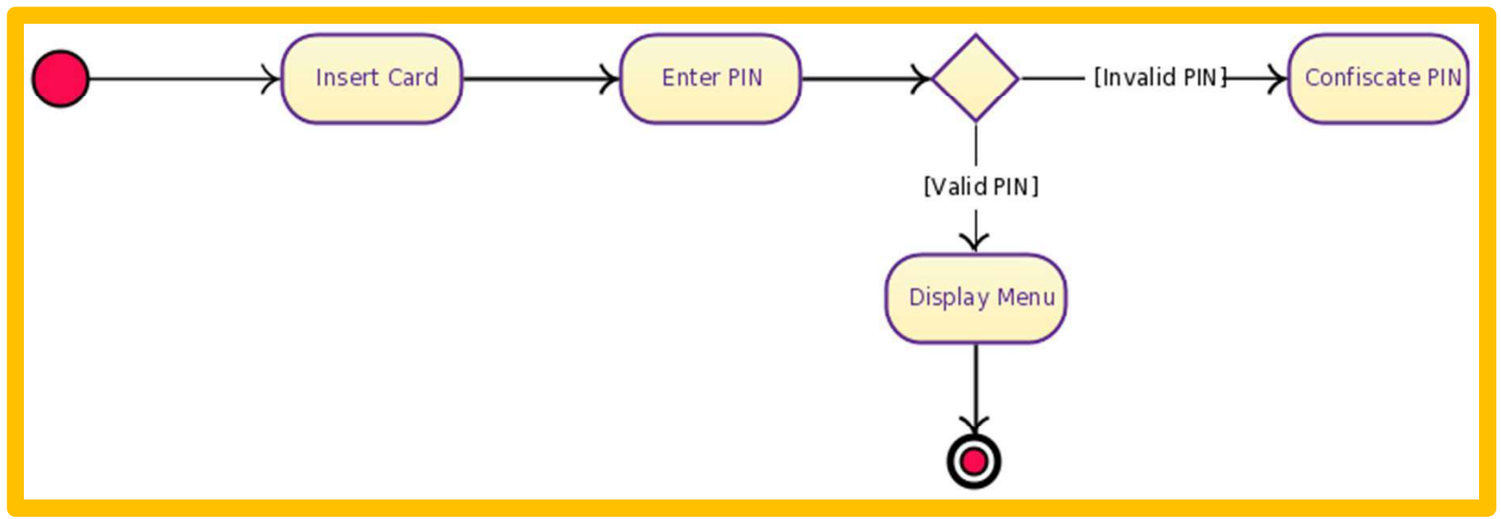

Actor node
System name
System boundary
Banking System
Customer
ATM Login
«include»
ATM Withdrawal
«extend»
the customer requests a balance
Dependency
Teller Customer
ATM Customer
Determine Balance
Use case node
«actor»
BankTeller
Checking Deposit for Customer
«extend»
the customer asks for balance
Actor node

Example

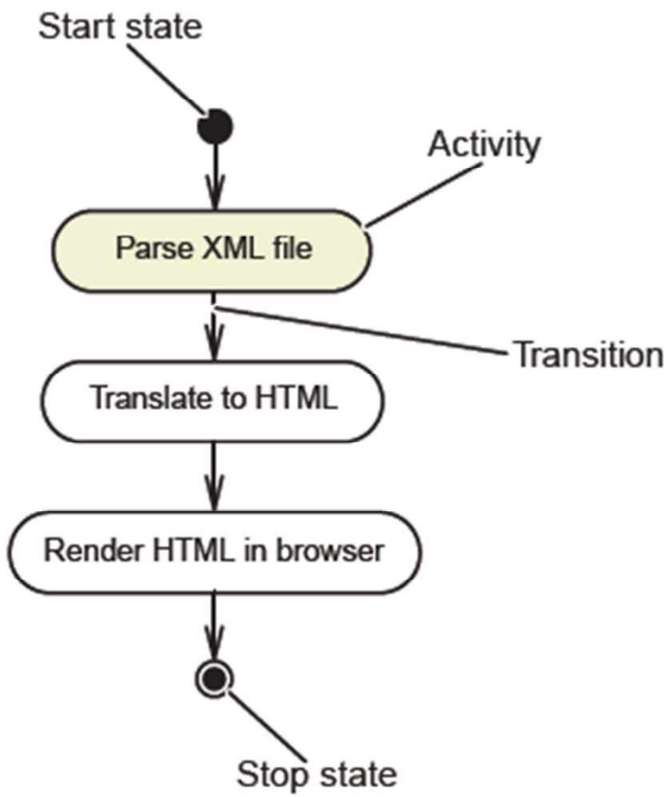


Activity Diagram

- Activity diagrams represent workflows in a graphical way. They can be used to describe business workflow or the operational workflow of any component in a system. Sometimes activity diagrams are used as an alternative to State machine diagrams.

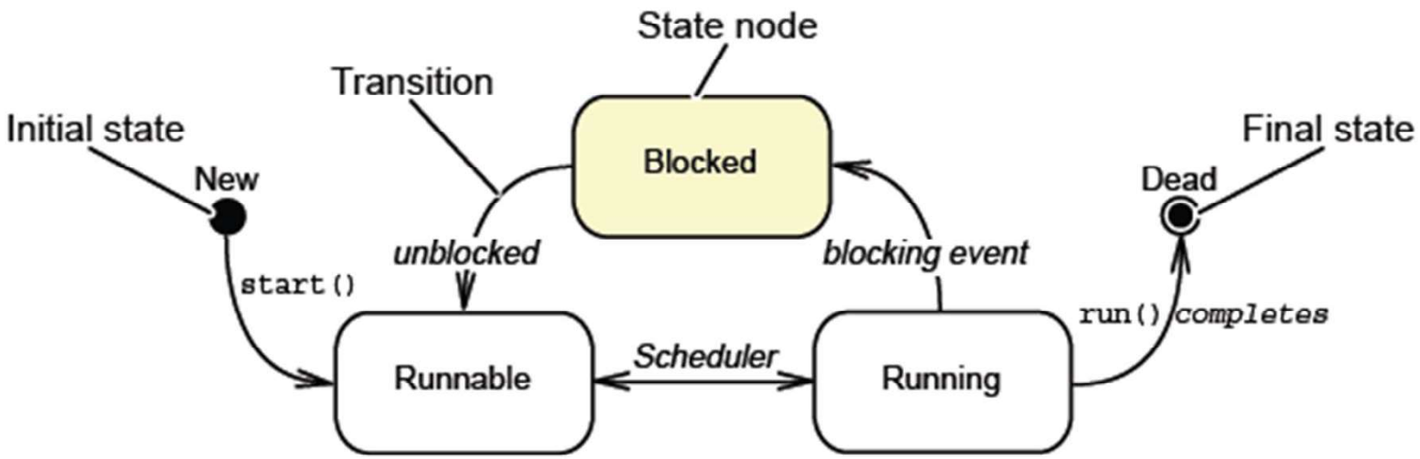


Example

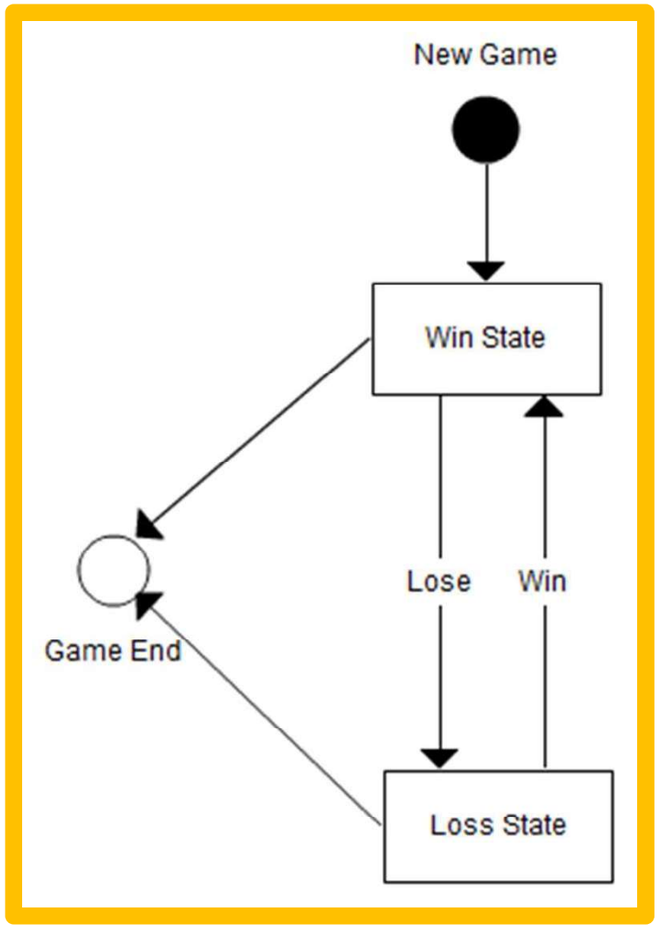

Start state
Activity
Parse XML file
Transition
Translate to HTML
Render HTML in browser
Stop state

State Machine Diagram

- State machine diagrams are similar to activity diagrams, although notations and usage change a bit.
- They are sometime known as state diagrams or state chart diagrams as well.
- These are very useful to describe the behavior of objects that act differently according to the state they are in at the moment.
- The State machine diagram below shows the basic states and actions.


State node
Transition
Initial state
New
Blocked
Final state
Dead
unblocked
blocking event
start()
run()
completes
Runnable
Scheduler
Running

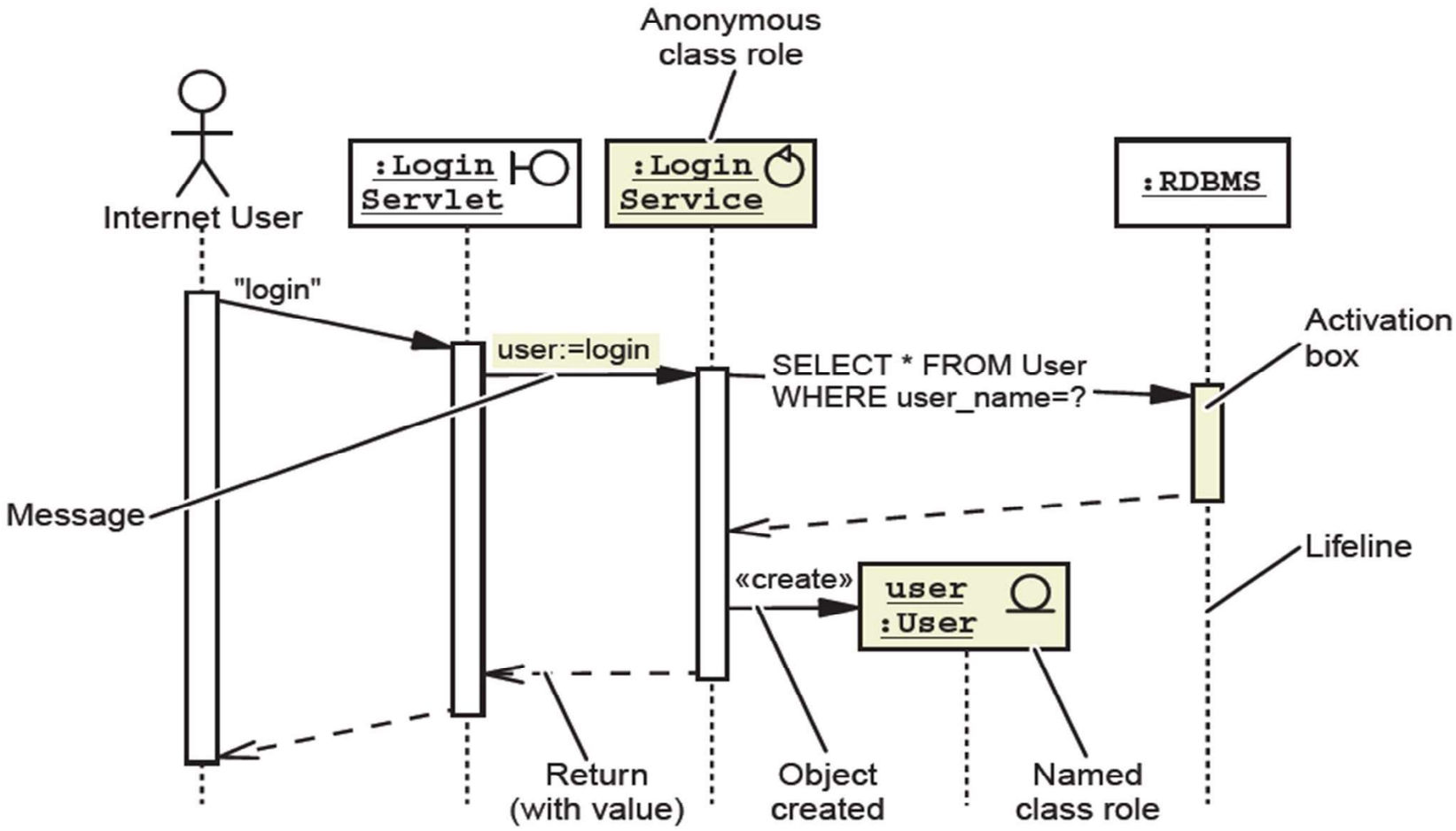
Example



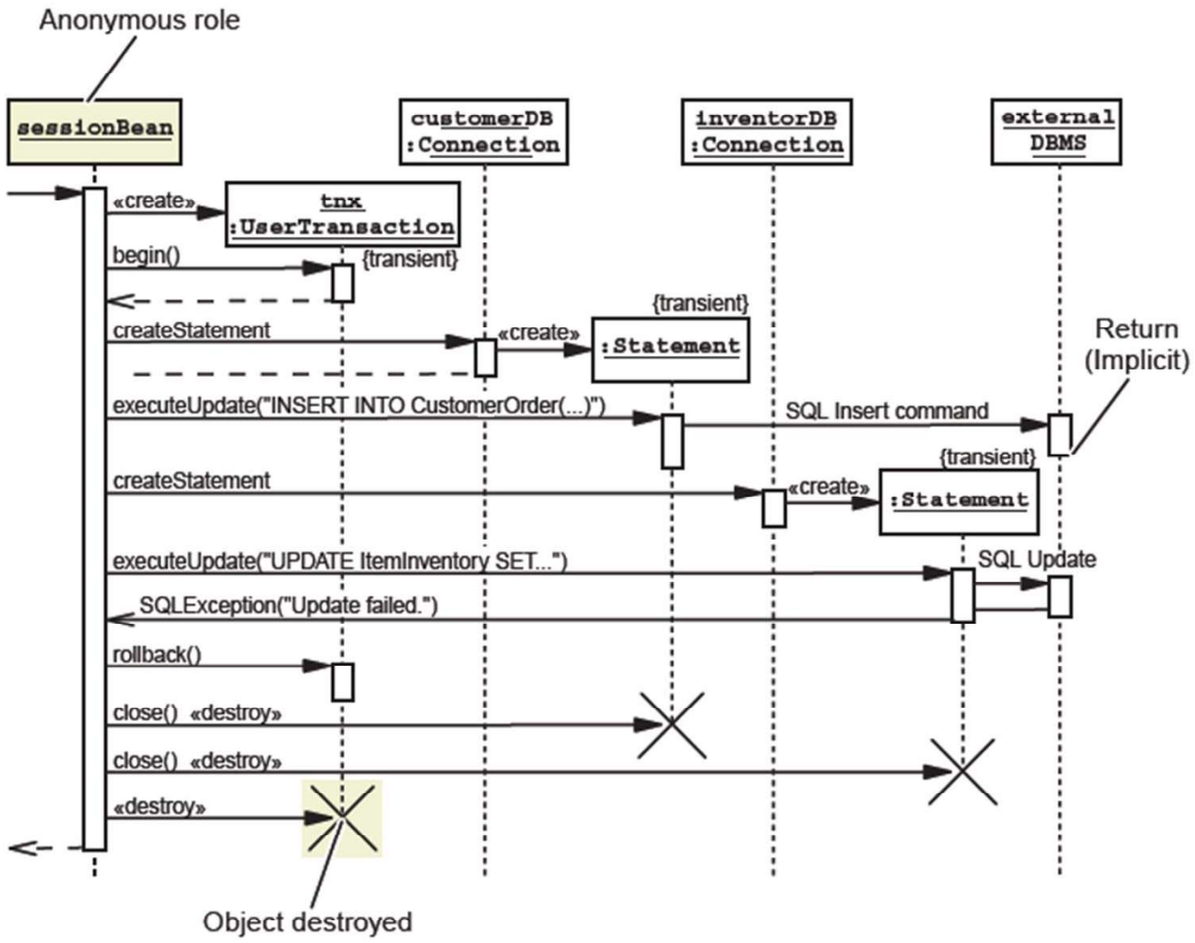
Sequence Diagram

- Sequence diagrams in UML show how objects interact with each other and the order those interactions occur.
- It's important to note that they show the interactions for a particular scenario. The processes are represented vertically and interactions are show as arrows.

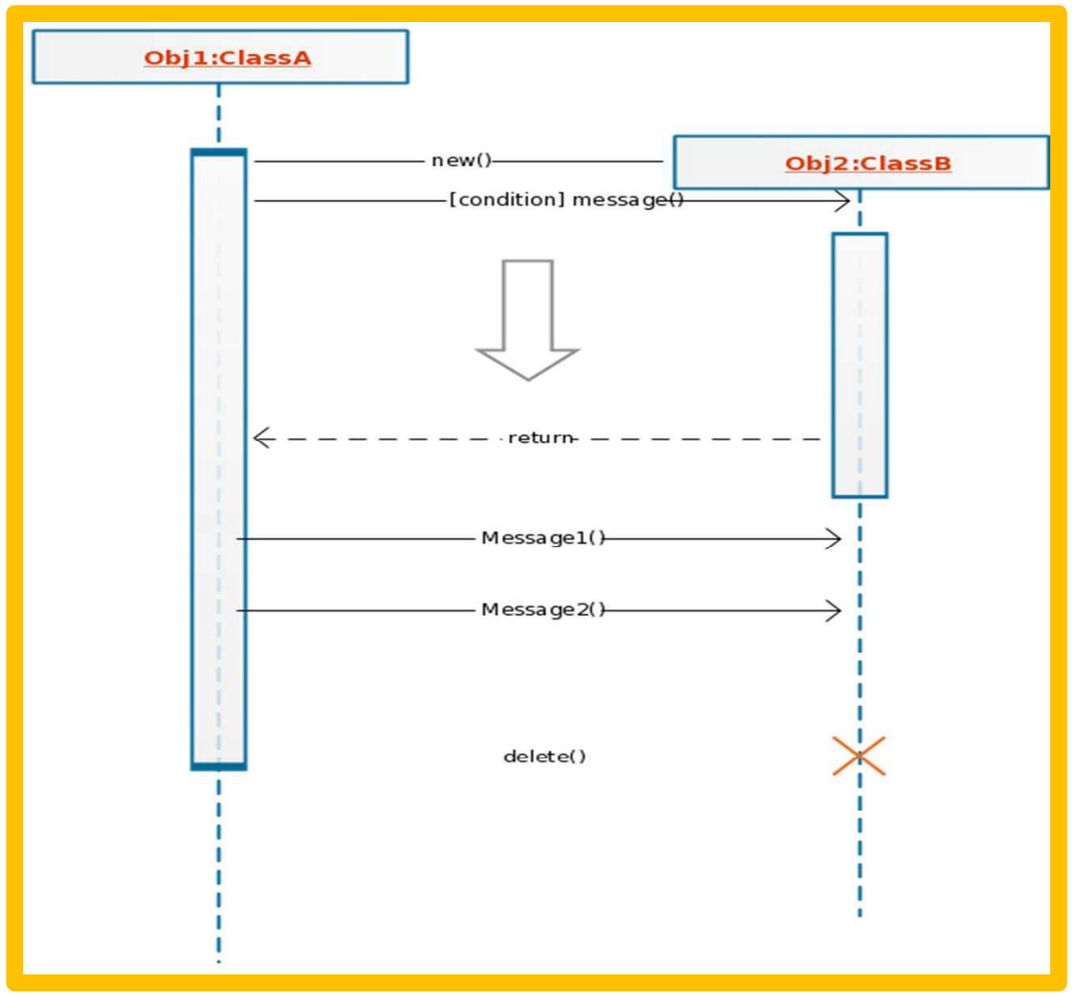
Example



Example



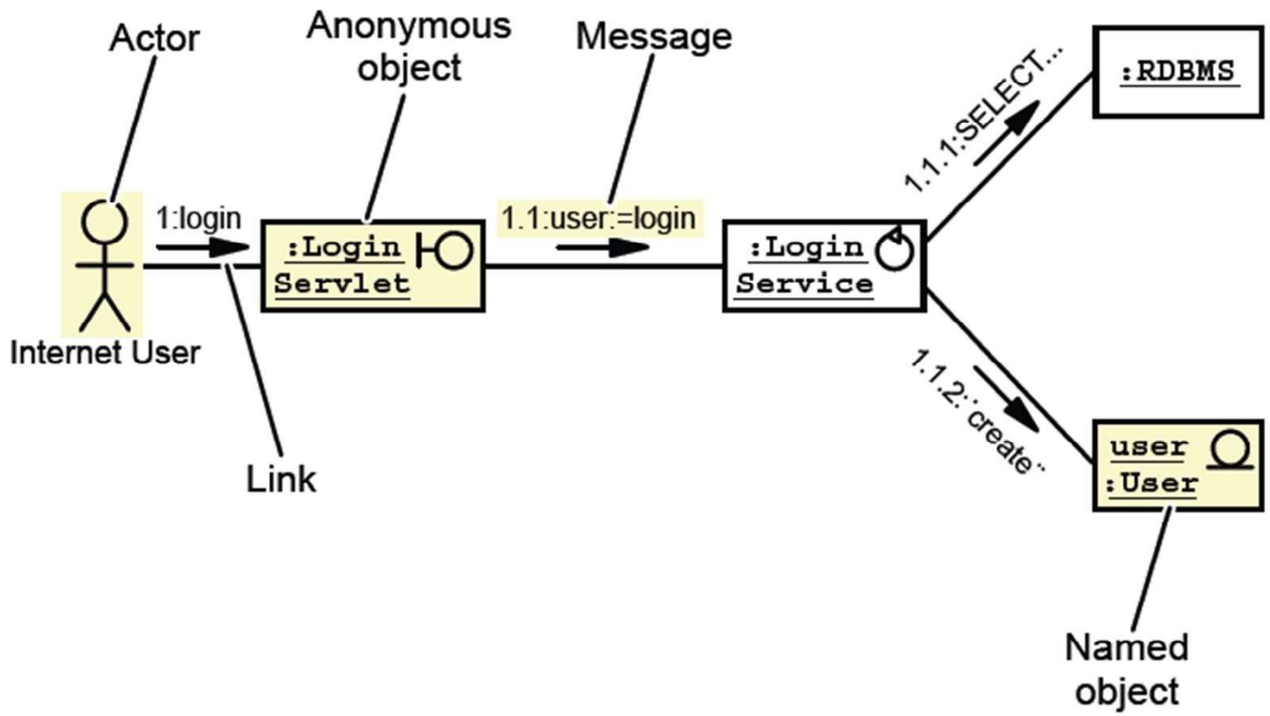
Example


Obj1:ClassA
Obj2:ClassB
new()
[condition] message()
return
Message1()
Message2()
delete()

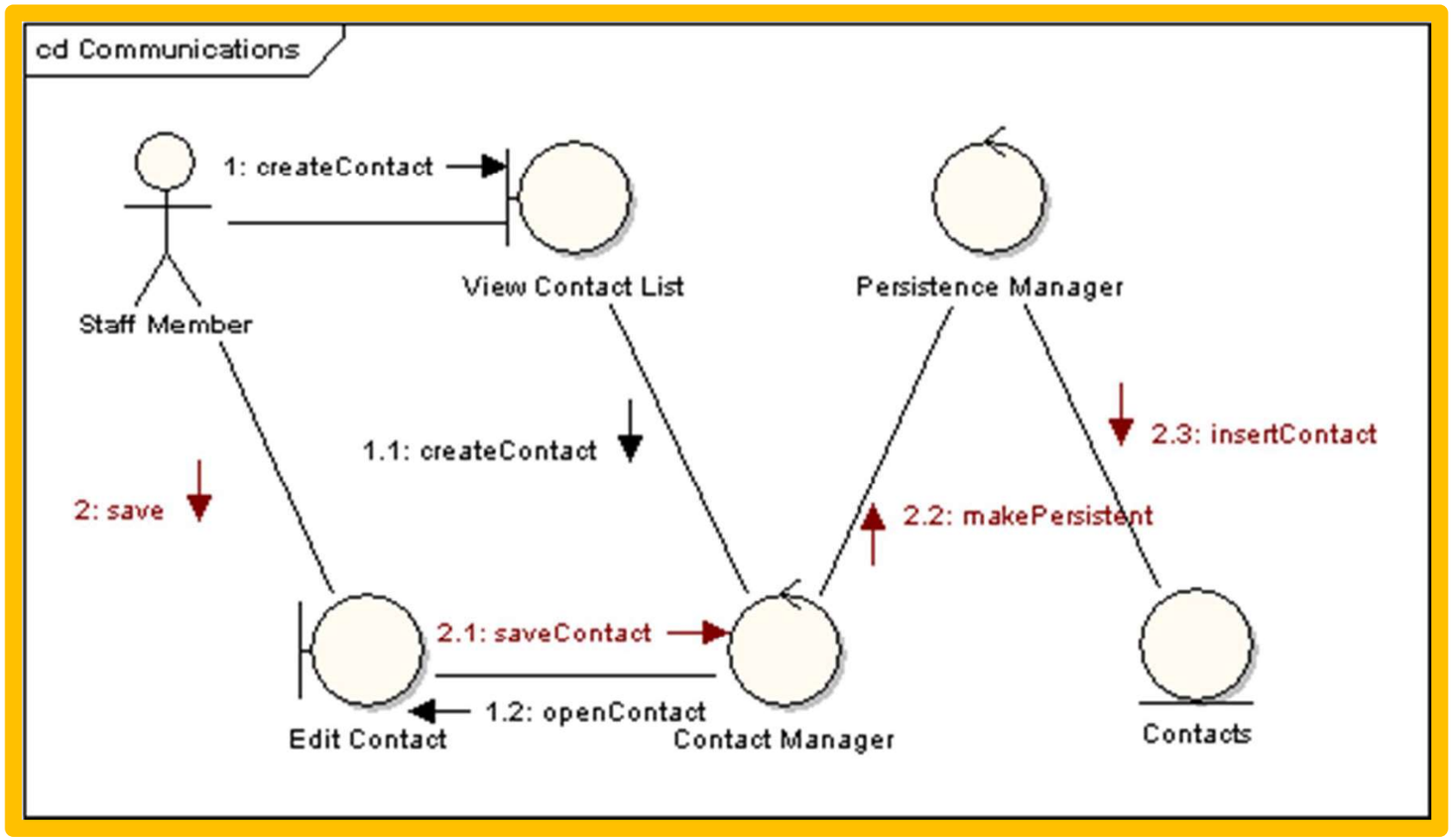
Communication Diagram

- Communication diagram was called collaboration diagram in UML 1.
- It is similar to sequence diagrams, but the focus is on messages passed between objects.
- The same information can be represented using a sequence diagram and different objects.

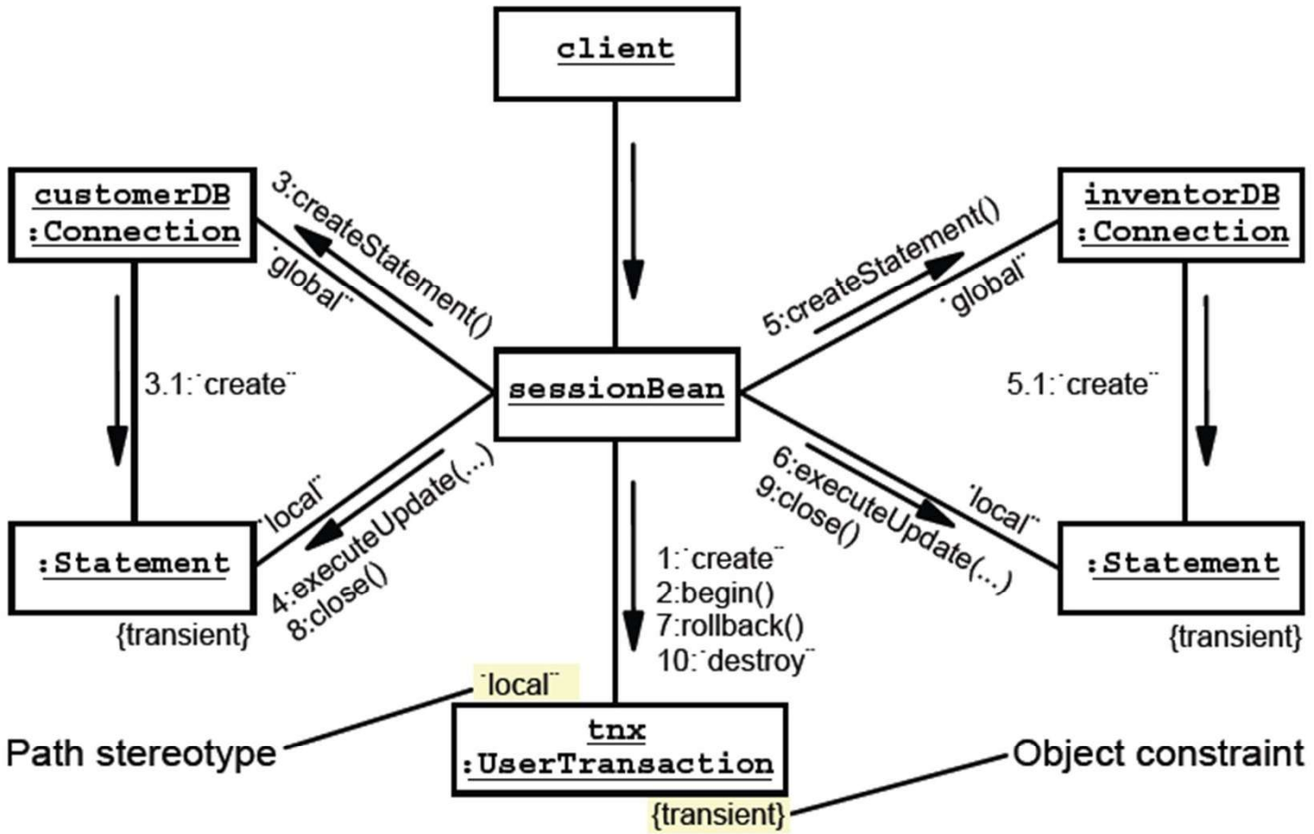
Example


Actor
Anonymous object
Message
1:login
:Login Servlet
1.1:user:=login
:Login Service
1.1.1:SELECT...
:RDBMS
1.1.2:"create"
user :User
Internet User
Link
Named object

Example


cd Communications
1: createContact
Staff Member
View Contact List
Persistence Manager
1.1: createContact
2.3: insertContact
2: save
2.2: makePersistent
2.1: saveContact
1.2: openContact
Edit Contact
Contact Manager
Contacts

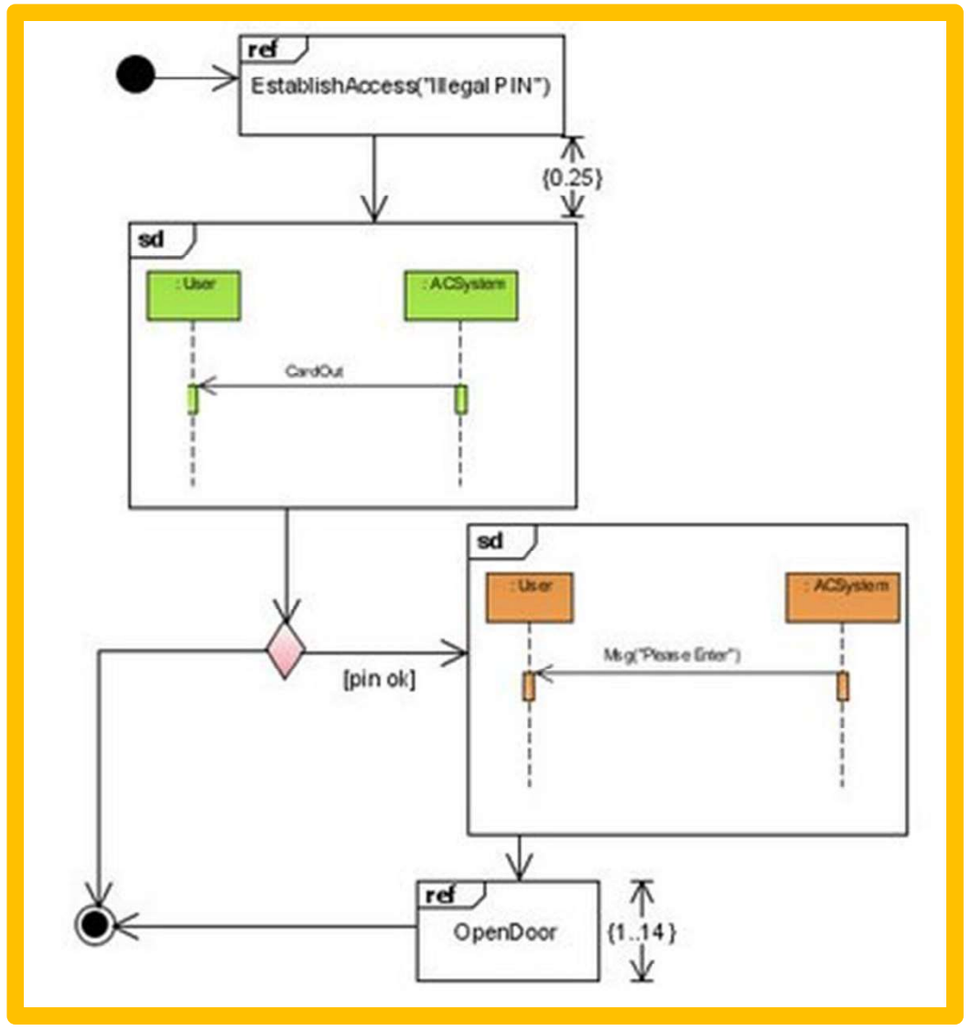
Example



Interaction Overview Diagram

- Interaction overview diagrams are very similar to activity diagrams.
- While activity diagrams show a sequence of processes, Interaction overview diagrams show a sequence of interaction diagrams.
- In simple terms, they can be called a collection of interaction diagrams and the order they happen.

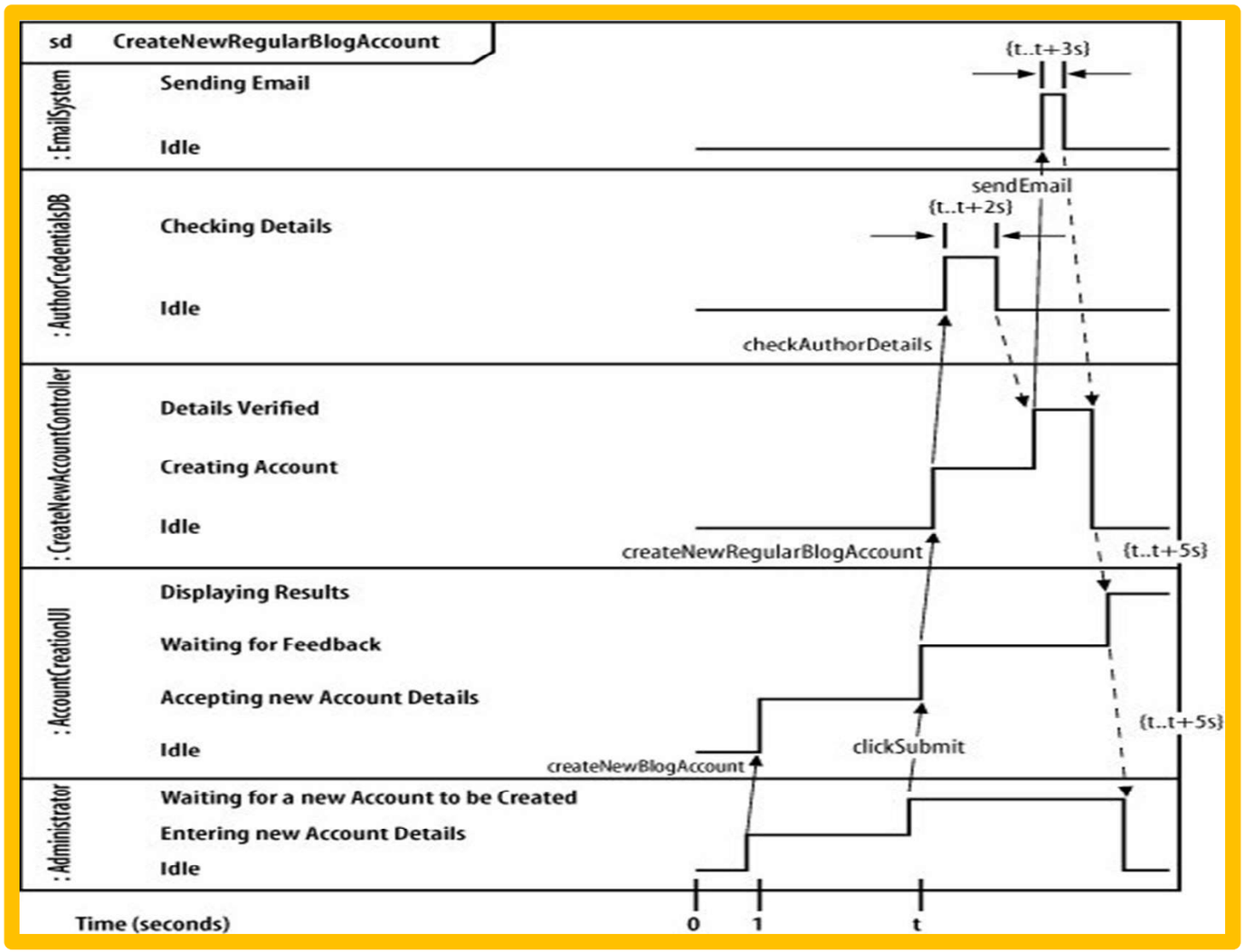
Example


ref
EstablishAccess("Illegal PIN")
{0.25}
sd
: User
: ACSystem
CardOut
sd
: User
: ACSystem
Msg("Please Enter")
[pin ok]
ref
OpenDoor
{1..14}

Timing Diagram

- Timing diagrams are very similar to sequence diagrams.
- They represent the behavior of objects in a given time frame.
- If it's only one object, the diagram is straight forward, but if there are more than one object involved, they can be used to show interactions of objects during that time frame as well.

Example



Summary

In this lesson, you should have learned the following:

- Explore different types of Structural Diagrams
- Explore different types of Behavioral Diagrams