

Chapter 2

Overview about Object Oriented Analysis and Design (OOAD)

MS. Le Thanh Trong

Outline

1. Object Oriented concepts
2. Object Oriented modeling
3. Object Oriented Analysis and Design
4. Object Oriented Analysis
5. Object Oriented Design
6. Unified Modeling Language

Outline

- 1. Object Oriented concepts**
2. Object Oriented modeling
3. Object Oriented Analysis and Design
4. Object Oriented Analysis
5. Object Oriented Design
6. Unified Modeling Language

Introduction

- ❖ We live in a world of objects
- ❖ Object-Oriented view is an abstraction that models the world in ways that help us to better understand and navigate it
- ❖ As time passes, object technologies are replacing classical software development approaches
- ❖ Object technologies lead to reuse, OO software is easier to maintain, to adapt, and to scale

OO Paradigm

- ❖ For many years, the term OO was used to denote a software development approach that used one of a number of OO programming languages(e.g. Ada 95, C++, Eiffel, Smalltalk, Java, C#,..)
- ❖ Today, the OO paradigm encompasses a complete view of software engineering
- ❖ Although any one of process models, could be adapted for use with OO, the best choice would be an **evolutionary process model**

OO Concepts

- ❖ Classes
- ❖ Objects
- ❖ Attributes
- ❖ Methods
- ❖ Abstraction and hiding
- ❖ Encapsulation
- ❖ Inheritance
- ❖ Polymorphism
- ❖ Messages

Objects

- ❖ Object is an abstract representation of some real world entity, such as a person, a car or a bank account
- ❖ Objects hold
 - data items (attributes) about the entity (for example name or account number)
 - operations (or methods) to report the values held by attributes

objects



Audi



Nissan



Volvo

Classes

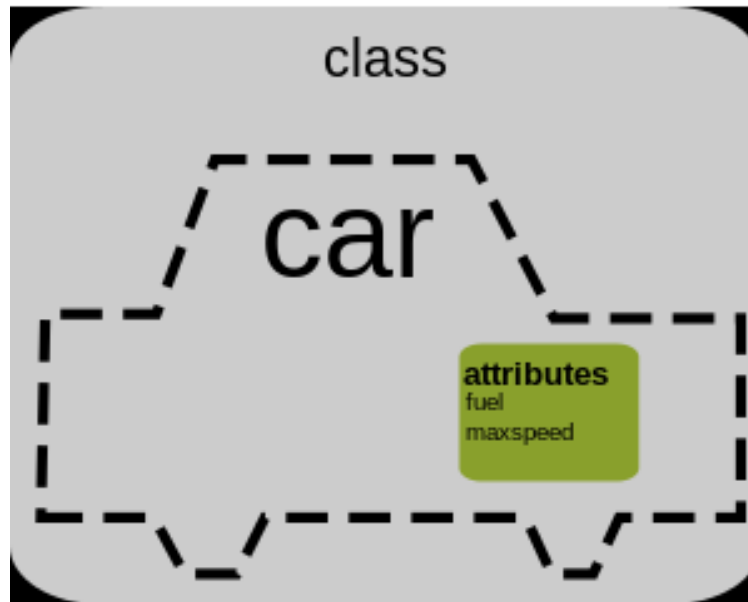
- ❖ Is a template that defines the **attributes** and **operations** that are common to the objects created from it
- ❖ An object is an instance (a specific example) of a class



BankAccount
AccountNo Balance
ReturnBalance UpdateBalance

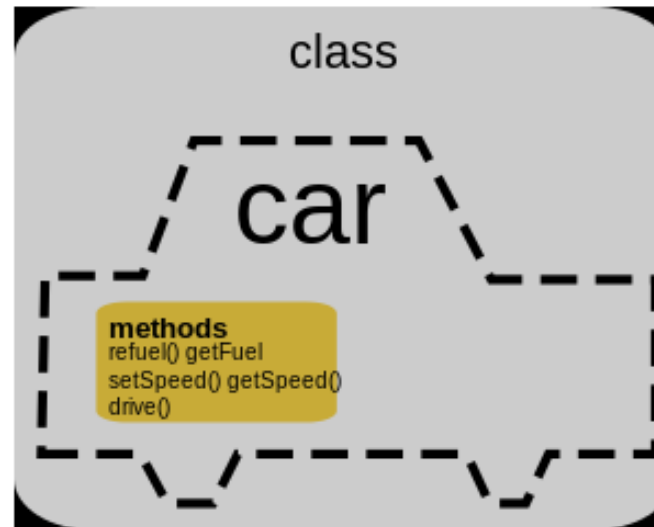
Attributes

- ❖ Attributes are data stored inside a class or instance and represent the state or quality of the class or instance
- ❖ In short, attributes store information about the instance



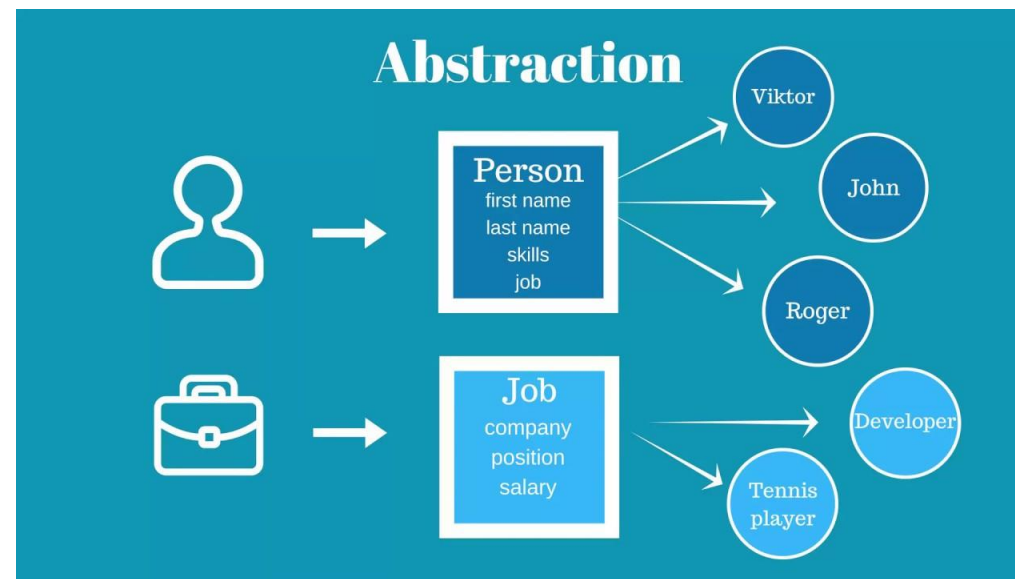
Methods

- ❖ Is a procedure associated with a message and an object. An object consists of data and behavior
- ❖ These comprise an interface, which specifies how the object may be utilized by any of its various consumers



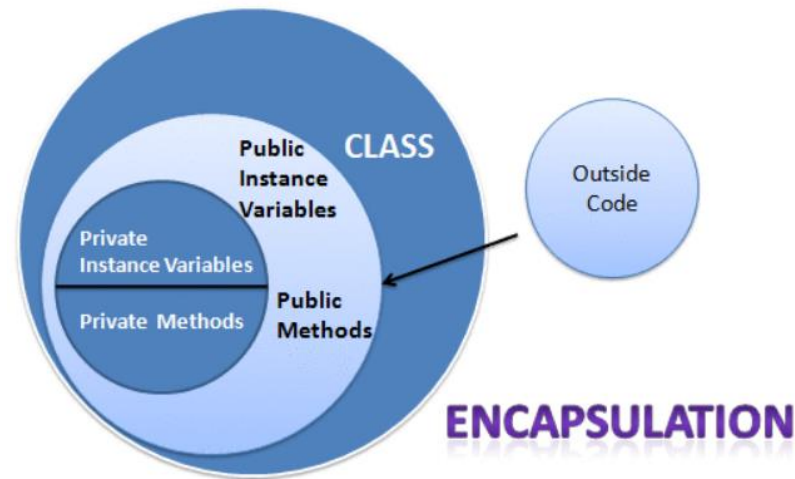
Abstraction

- ❖ Abstraction is the concept that "shows" only essential attributes and "hides" unnecessary information
- ❖ The main purpose of abstraction is hiding the unnecessary details from the users



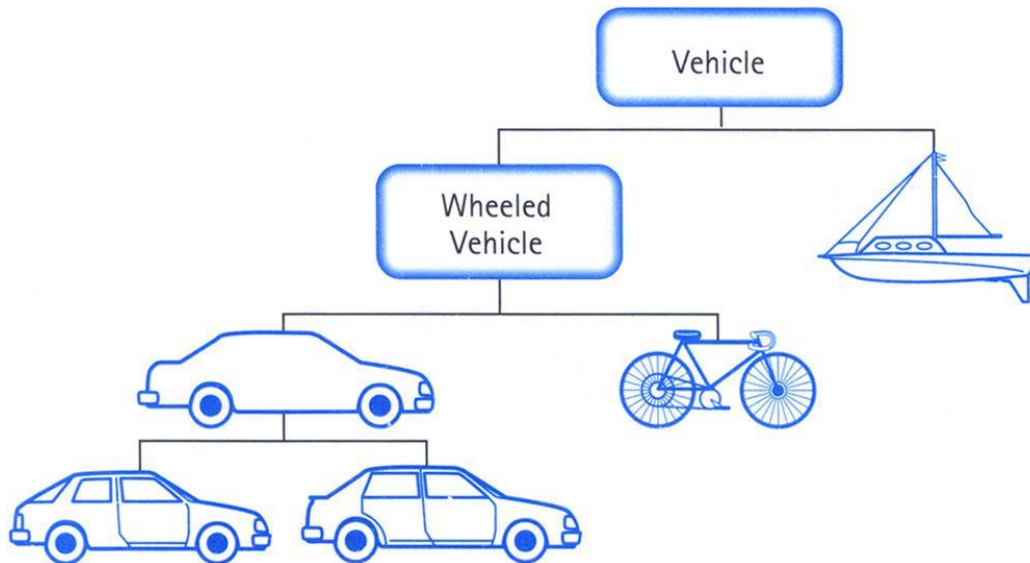
Encapsulation

- ❖ It describes the idea of bundling data and methods that work on that data within one class
- ❖ Often used to hide the internal representation, or state, of an object from the outside, this is called **information hiding**



Inheritance

- ❖ It is a mechanism where can to derive a class from another class for a hierarchy of classes that share a set of attributes and methods
- ❖ Provides code re-usability
- ❖ Is a way of representing real-world relationships



Polymorphism

- ❖ Polymorphism enables a number of different operations to have the same name
- ❖ It decouples objects from one another making each more independent

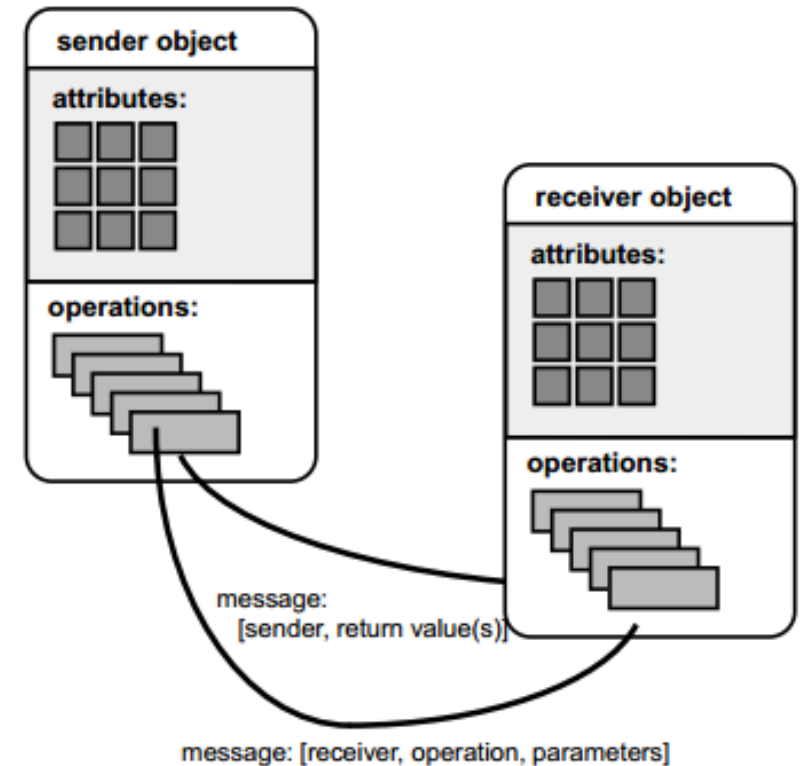


arbabwaseer@gmail.com

22

Messages

- ❖ Is a function or procedure call from one object to another, are information sent to objects to trigger methods
- ❖ Is a message is a function or procedure call from one object to another
- ❖ A message has four parts:
 - Identity of the recipient object
 - Code to be executed by the recipient
 - Arguments for the code
 - Return value



Object-oriented system

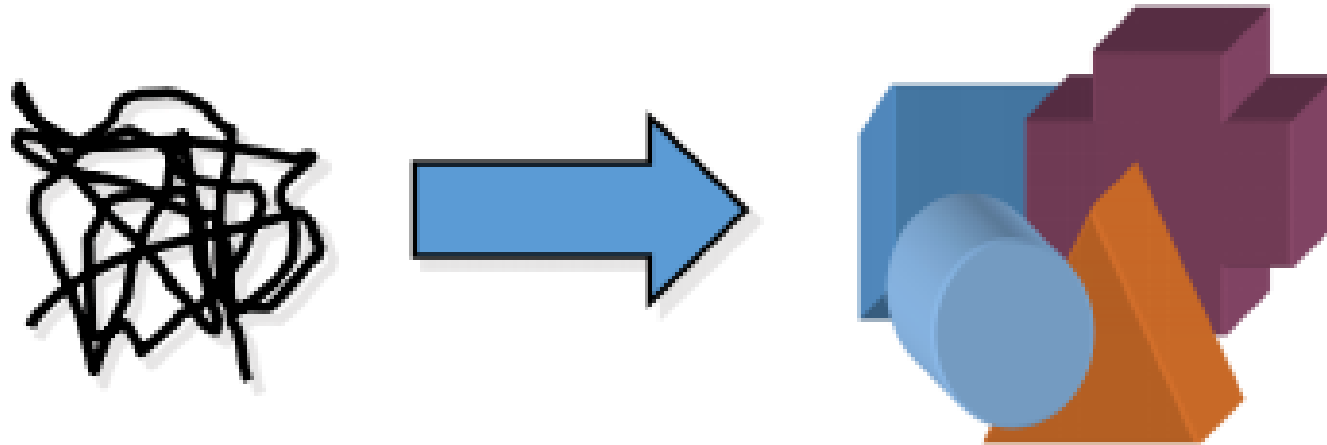
- ❖ Is an operating system that is designed, structured, and operated using object oriented programming principles
- ❖ All data is represented as discrete objects with which the user and other objects may interact
- ❖ The OO system model is beneficial:
 - Facilitates changes in the system at low cost
 - Promotes the reuse of components
 - Simplifies the problem of integrating components to configure large system
 - Simplifies the design of distributed systems

Outline

1. Object Oriented concepts
- 2. Object Oriented modeling**
3. Object Oriented Analysis and Design
4. Object Oriented Analysis
5. Object Oriented Design
6. Unified Modeling Language

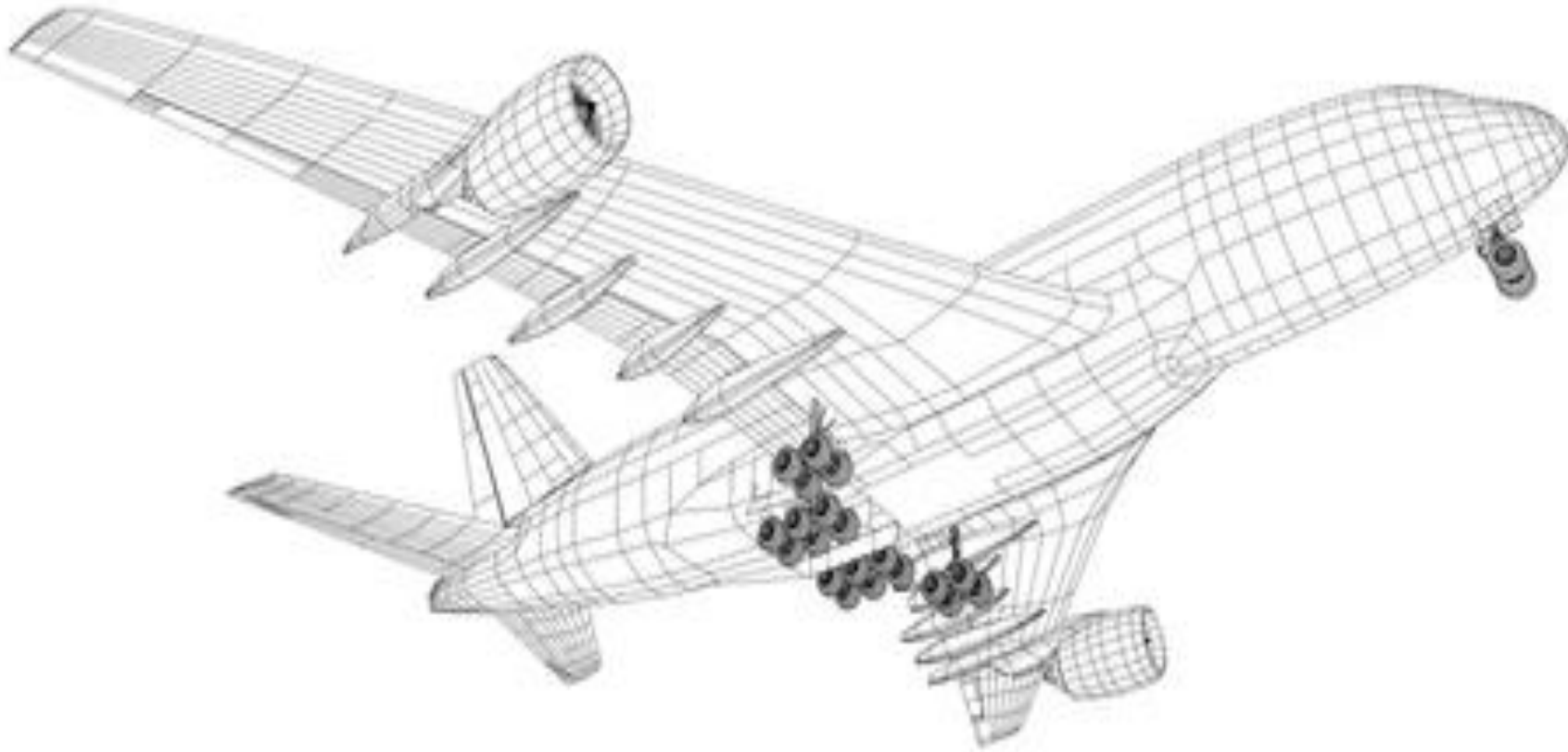
What is modeling?

- ❖ To represent how a business works and functions in such a way that it can productively be used as a means to simulate the real world usually on a smaller scale



The importance of modeling

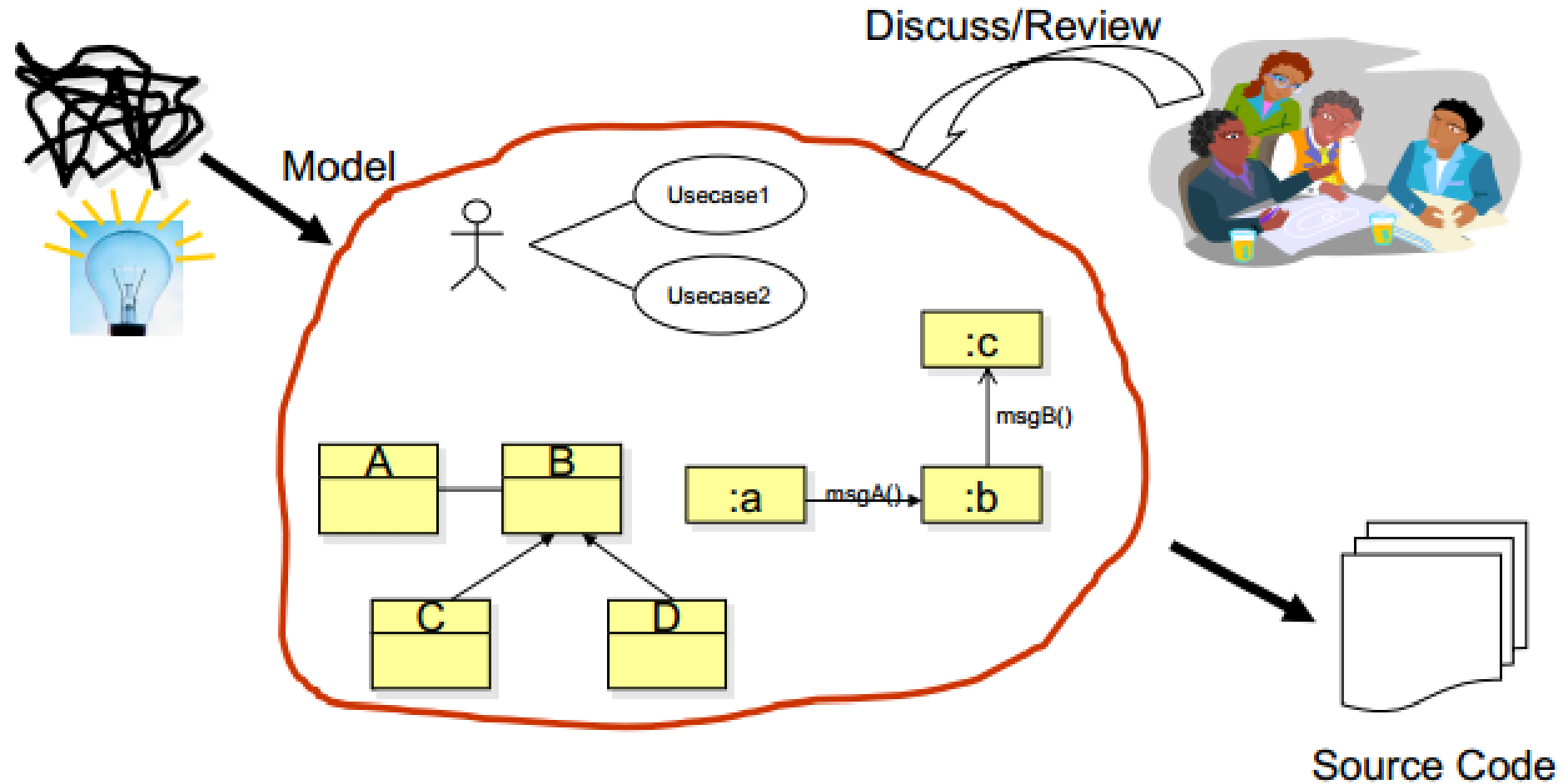
LOGO



Why do we modeling?

- ❖ To better understand the system we are developing
- ❖ Helps to visualize a system
- ❖ Permits us to specify the structure or behavior of a system
- ❖ Gives us a template that guides us in constructing a system
- ❖ Documents the decisions we have made
- ❖ We build models of complex systems because we can not comprehend such a system in its entirety

Modeling in software



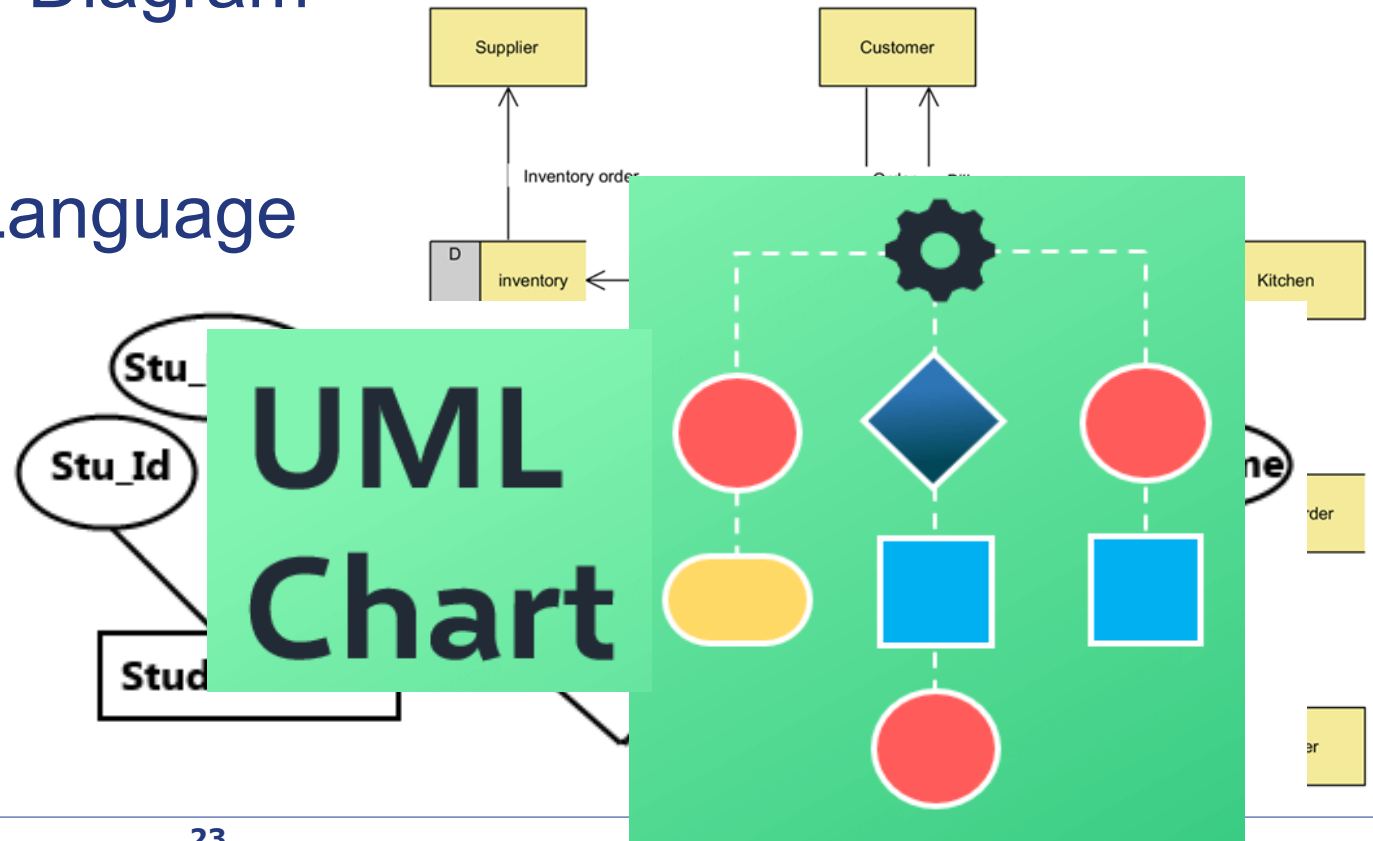
The advantage of using model in software

- ❖ It is easy to send the thinking image by clearing the small part
- ❖ It is able to sharing thing created by relationship between related person/team
- ❖ Can understand the outline of complicated software
- ❖ To adjust the problem, can correspond flexible to the changing of requirement

Modeling is to decrease complication, share information between related persons, correspond to changing requirement, effective in maintain software

Popular methodologies software modeling

- ❖ Structure oriented
 - DFD: Data Flow Diagram
 - ERD: Entity Relationship Diagram
- ❖ Object oriented
 - UML: Unified Modeling Language

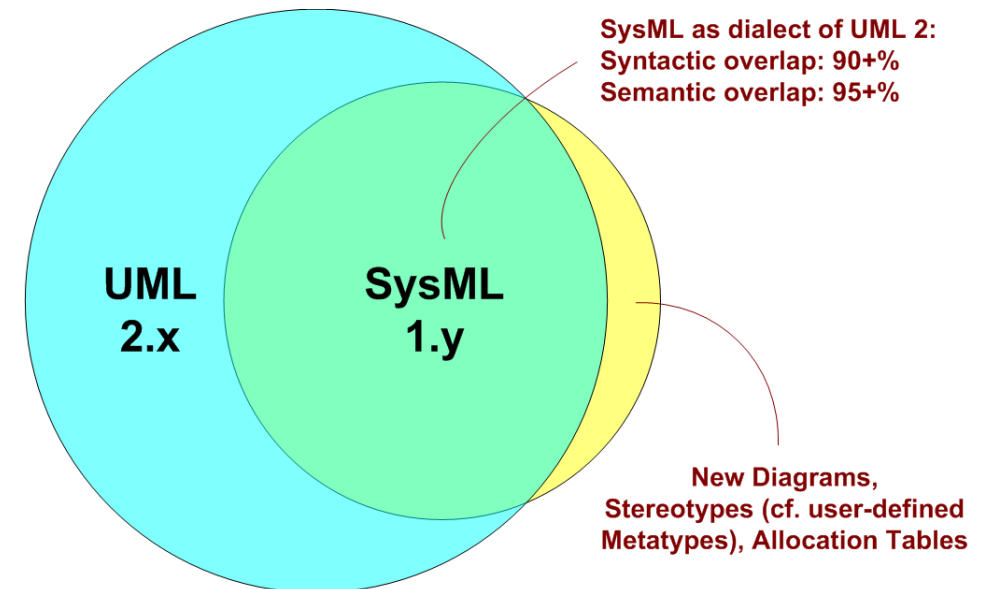


Object-oriented modeling (OOM)

- ❖ Is a common approach to modeling applications, systems, and business domains by using the object-oriented paradigm
- ❖ Is a main technique heavily used by both OOD and OOA activities in modern software engineering
- ❖ Divides into two aspects of work:
 - dynamic behaviors like business processes and use cases
 - static structures like classes and components

Object-oriented modeling (OOM)

- ❖ OOA and OOD are the two distinct abstract levels during OOM
- ❖ The Unified Modeling Language (**UML**) and **SysML** are the two popular international standard languages used for object-oriented modeling



The benefits of OOM

- ❖ Efficient and effective communication
 - Visual model diagrams can be more understandable
 - Can allow users and stakeholders to give developers feedback on the appropriate requirements and structure of the system
 - System be constructed using terminology that is almost the same as the stakeholders use in everyday business
- ❖ Useful and stable abstraction
 - Producing abstract and accessible descriptions of both system requirements and designs
 - Make specific solutions to these abstract requirements, and refine them into detailed designs

Outline

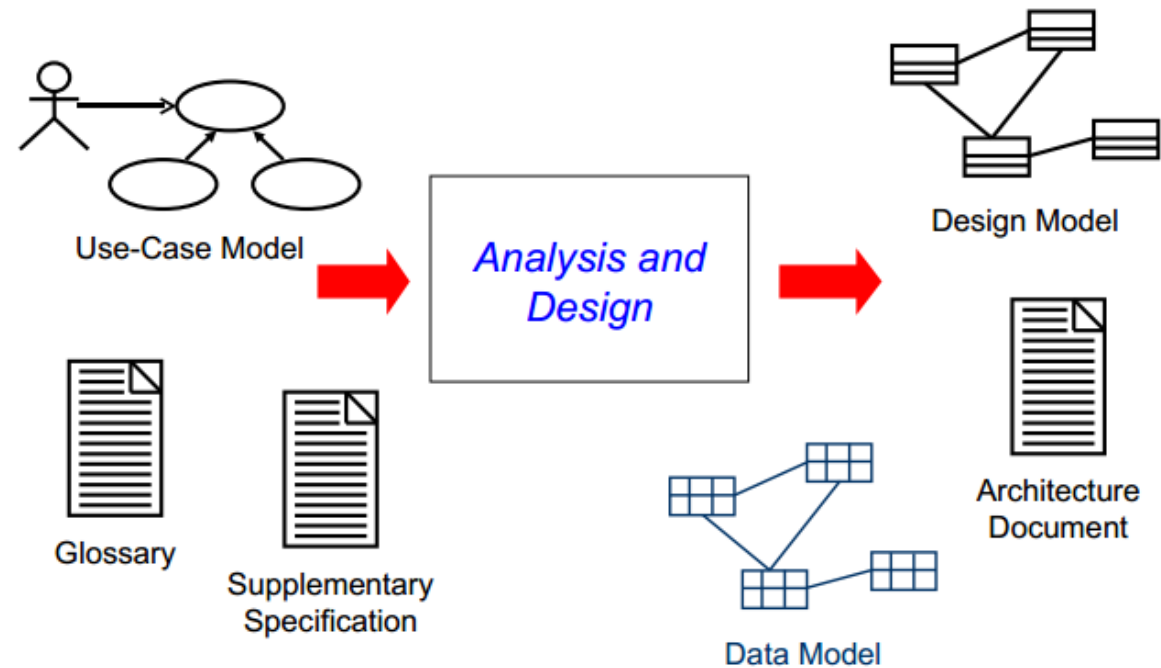
1. Object Oriented concepts
2. Modeling and Object Oriented modeling
- 3. Object Oriented Analysis and Design**
4. Object Oriented Analysis
5. Object Oriented Design
6. Unified Modeling Language

Object Oriented Analysis and Design

- ❖ Is a technical approach for analyzing and designing an application, system, or business by applying object-oriented technique
- ❖ Using visual modeling throughout the software development process to guide stakeholder communication and product quality
- ❖ The purposes of Analysis and Design are to:
 - Transform the requirements into a design of the system
 - Evolve a robust architecture for the system
 - Produce a blueprint for the physical implementation, designing it for performance

Object Oriented Analysis and Design

- ❖ In modern software engineering is typically conducted in an iterative and incremental way



Object Oriented Analysis and Design

Analysis	Design
<ul style="list-style-type: none">□ Focus on understanding the problem□ Idealized design□ Behavior□ System structure□ Functional requirements□ A small model <p>WHAT?</p>	<ul style="list-style-type: none">□ Focus on understanding the solution□ Operations and attributes□ Performance□ Close to real code□ Object lifecycles□ Nonfunctional requirements□ A large model <p>HOW?</p>

Outline

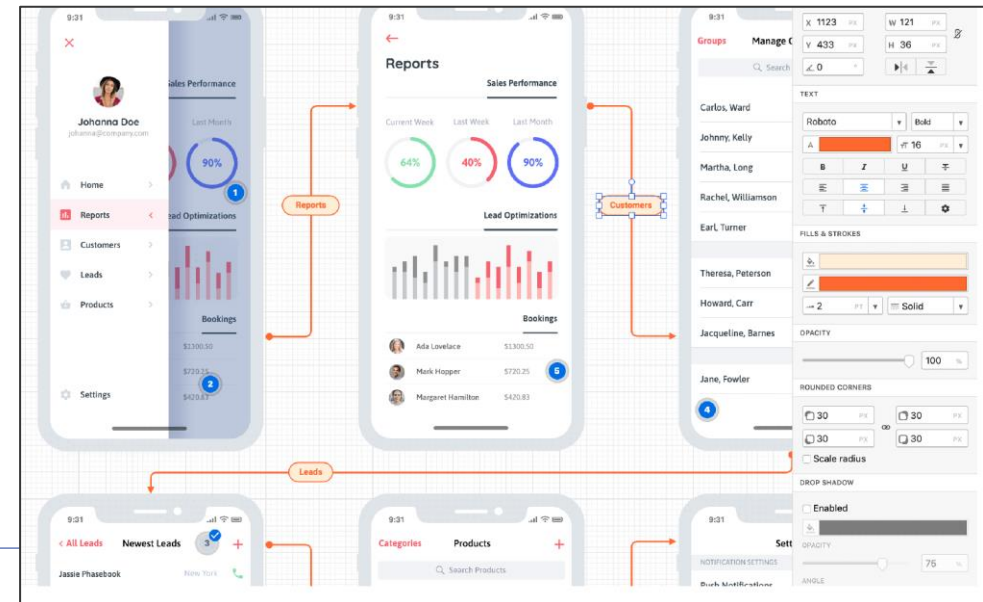
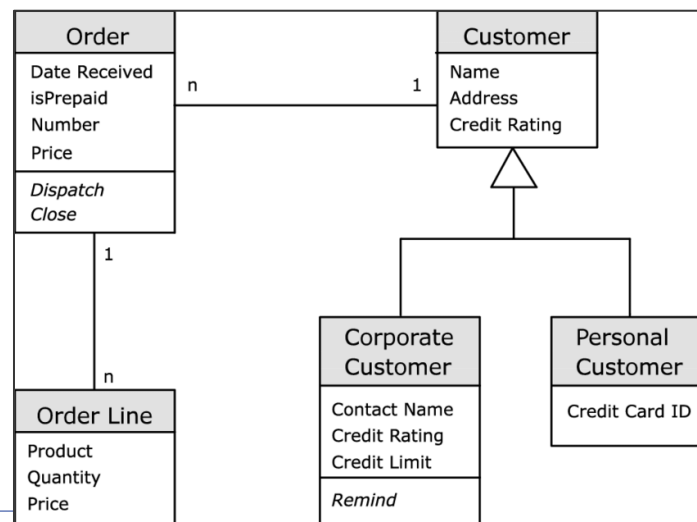
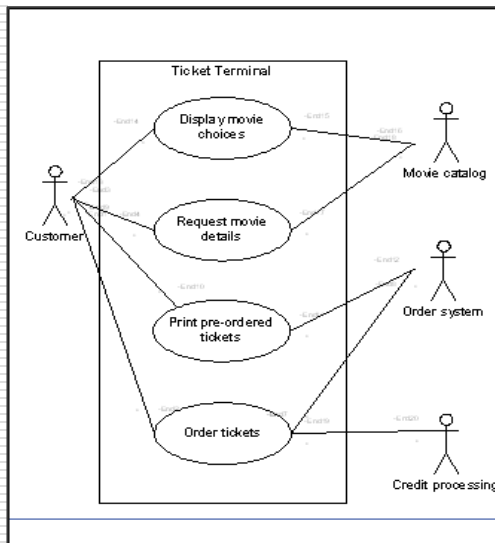
1. Object Oriented concepts
2. Modeling and Object Oriented modeling
3. Object Oriented Analysis and Design
- 4. Object Oriented Analysis**
5. Object Oriented Design
6. Unified Modeling Language

Object Oriented Analysis

- ❖ Is a method of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain (*Grady Booch*)
- ❖ Organize requirements around objects which integrate both behaviors (processes) and states (data)
- ❖ The primary tasks in object-oriented analysis (OOA) are:
 - Identifying objects
 - Organizing the objects by creating object model diagram
 - Defining the internals of the objects, or object attributes
 - Defining the behavior of the objects, i.e., object actions
 - Describing how the objects interact

Object Oriented Analysis

- ❖ Most use models: use cases and object/class models
 - Use cases describe scenarios for standard domain functions that the system must accomplish
 - Object/class models describe the names, class relations, operations, and properties of the main objects
 - User-interface mockups or prototypes can also be created



Object Oriented Analysis

- ❖ Three analysis techniques that are used in conjunction with each other for object-oriented analysis are
 - object modeling
 - dynamic modeling
 - functional modeling

Object modeling

- ❖ Develops the static structure of the software system in terms of objects

- ❖ Ident

be gr

- ❖ Ident

chara

- ❖ Follo

- Ident

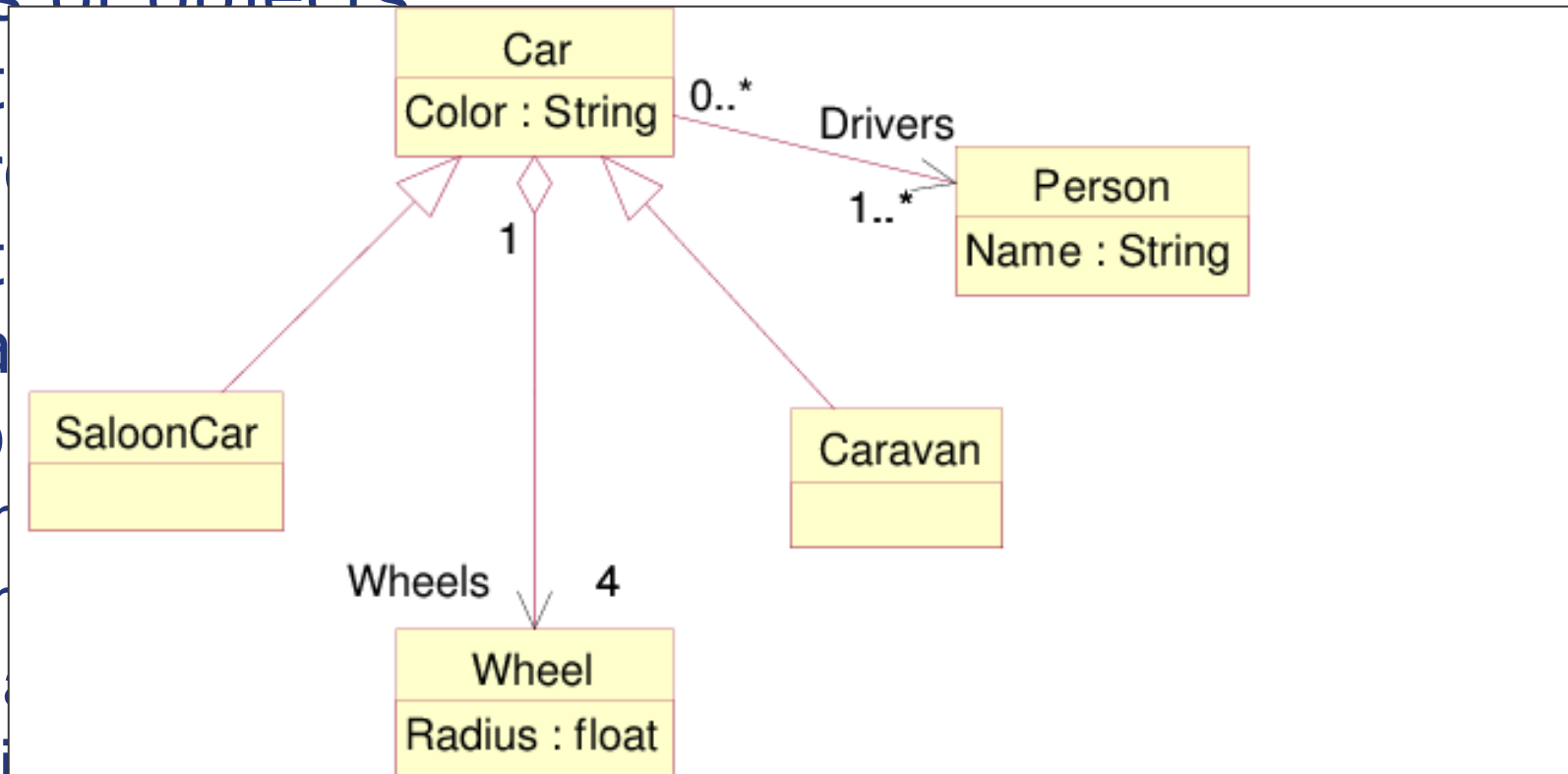
- Ident

- Cre

- Defi

- Define the operations that should be performed on the classes

- Review glossary



objects can
the objects
at

Dynamic Modeling

- ❖ Analyze its behavior with respect to time and external changes needs to be examined

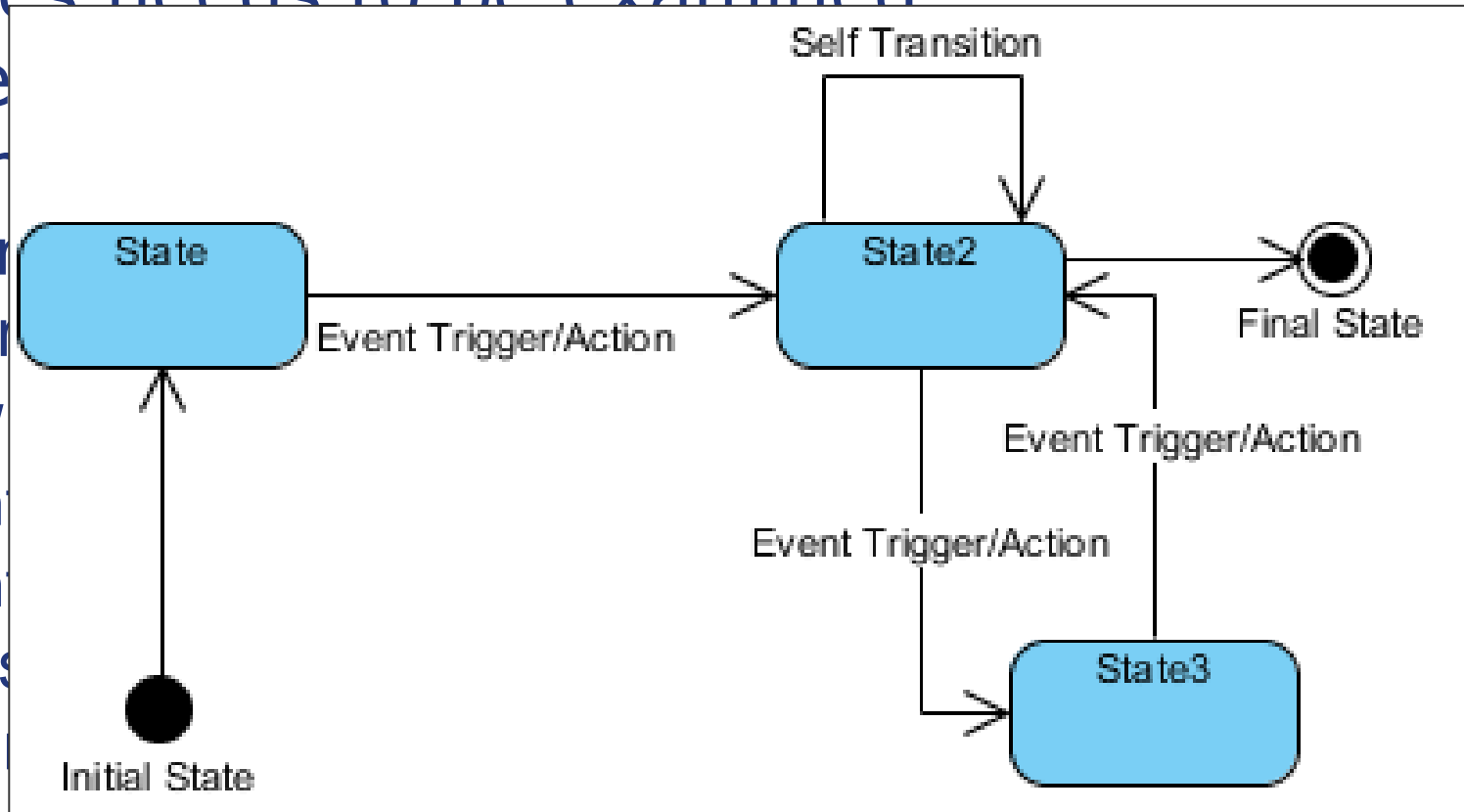
- ❖ Define response of the object

- Internal
- External

- ❖ Follow

- Identify
- Identify
- Construct
- Diagram

- Express each state in terms of object attributes
- Validate the state–transition diagrams drawn

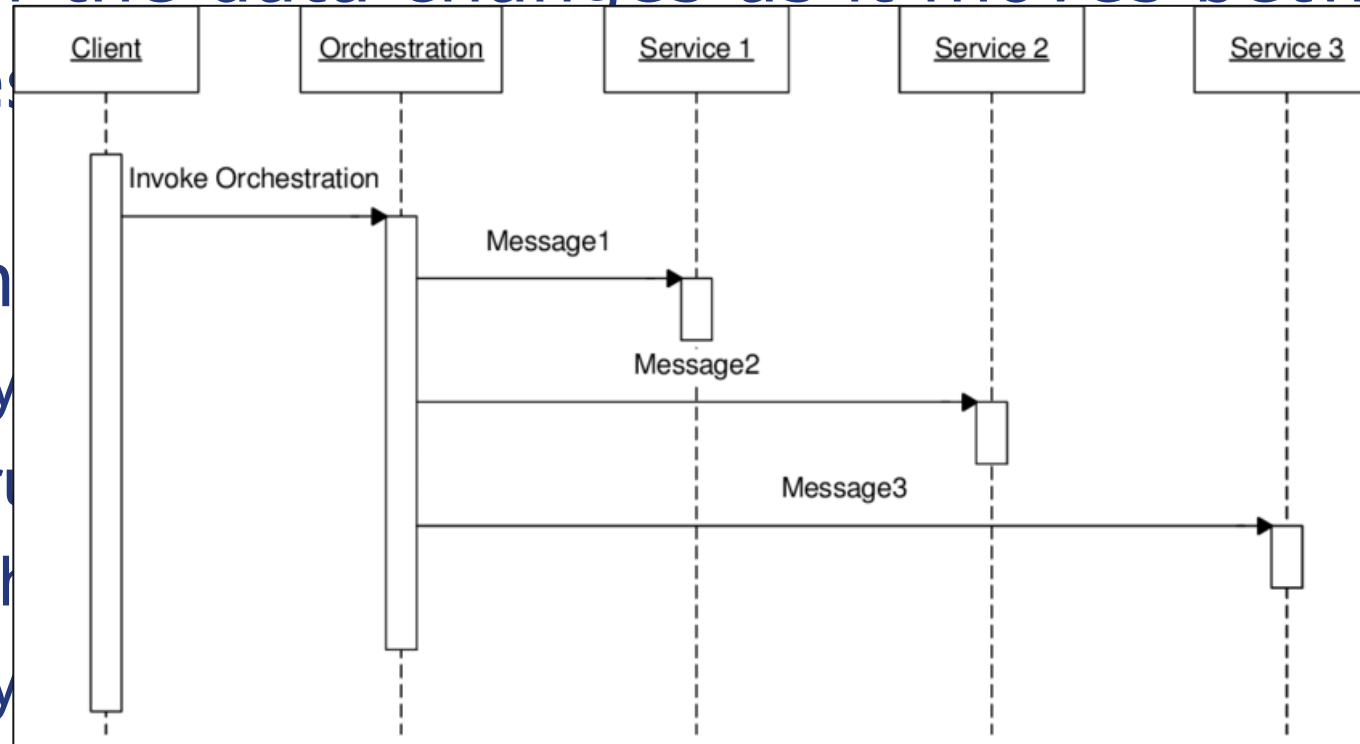


ual object

State transition

Functional Modelling

- ❖ Shows the processes that are performed within an object and how the data changes as it moves between methods
- ❖ Specifies and the
- ❖ Following
 - Identify
 - Constr
 - State th
 - Identify
 - Specify optimization criteria



Outline

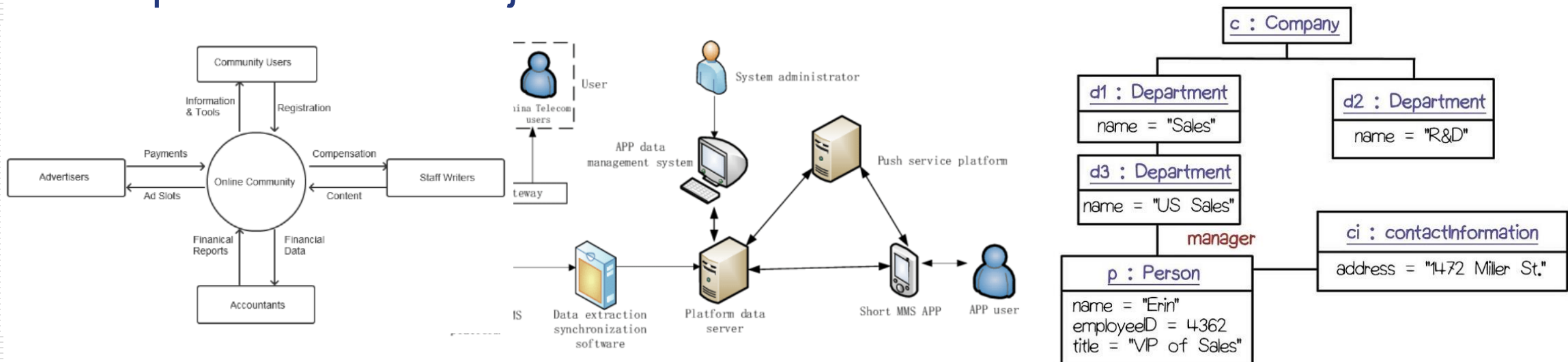
1. Object Oriented concepts
2. Modeling and Object Oriented modeling
3. Object Oriented Analysis and Design
4. Object Oriented Analysis
- 5. Object Oriented Design**
6. Unified Modeling Language

Object Oriented Design

- ❖ The conceptual model is developed further into an object-oriented model using object-oriented design
- ❖ The analysis model are mapped onto implementing classes, constraints are identified, and interfaces are designed
- ❖ A detailed description is constructed specifying how the system is to be built on concrete technologies

Object Oriented Design

- ❖ The stages for object-oriented design can be identified as
 - Definition of the context of the system
 - Designing system architecture
 - Identification of the objects in the system
 - Construction of design models
 - Specification of object interfaces



System Design

- ❖ Involves defining the context of a system followed by designing the architecture of the system
 - **Context:** has a static and a dynamic part
 - The static context: using a simple block diagram of the whole system which is expanded into a **hierarchy of subsystems**. The subsystem model is represented by **UML packages**
 - The dynamic context describes **how the system interacts** with its environment. It is modeled using use case diagrams
 - **System Architecture**
 - Is designed on the basis of the context of the system in accordance with the principles of architectural design as well as domain knowledge
 - A system is partitioned into layers and each layer is decomposed to form the subsystems

Object Design

- ❖ Object identification
- ❖ Object representation, construction of design models
- ❖ Classification of operations
- ❖ Algorithm design
- ❖ Design of relationships
- ❖ Implementation of control for external interactions
- ❖ Package classes and associations into modules

Packaging Classes

- ❖ Classes and objects are grouped into packages to enable multiple groups to work cooperatively on a project
- ❖ The different aspects of packaging
 - Hiding Internal Information from Outside View
 - Coherence of Elements
 - Construction of Physical Modules

Design Optimization

- ❖ To make the implementation more efficient
- ❖ Minimize the cost in terms of time, space, and other metrics
- ❖ Design optimization are:
 - Add redundant associations
 - Omit non-usable associations
 - Optimization of algorithms
 - Save derived attributes to avoid re-computation of complex expressions

Design Documentation

- ❖ High-level system architecture (Process diagrams and module diagrams)
- ❖ Key abstractions and mechanisms (Class diagrams and object diagrams)
- ❖ Scenarios that illustrate the behavior of the main aspects (Behavioural diagrams)

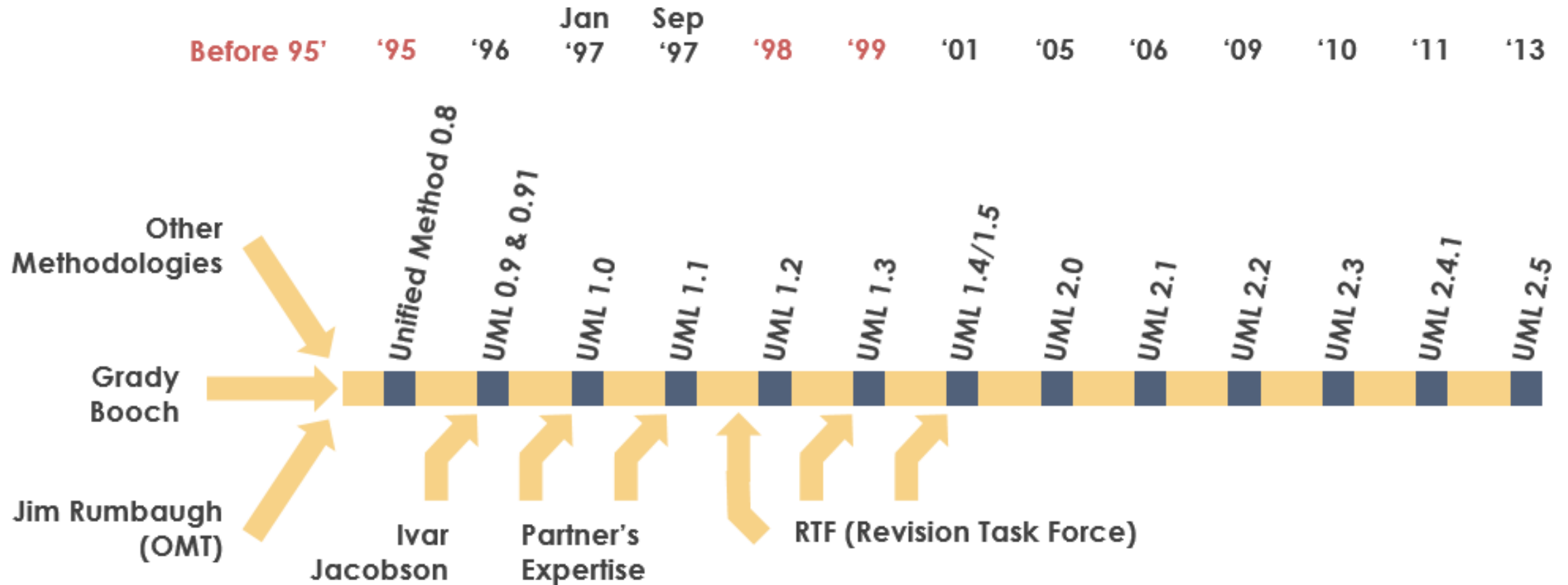
Outline

1. Object Oriented concepts
2. Modeling and Object Oriented modeling
3. Object Oriented Analysis and Design
4. Object Oriented Analysis
5. Object Oriented Design
- 6. Unified Modeling Language**

What is UML?

- ❖ Stands for **Unified Modeling Language**
- ❖ It is used for creating object-oriented models for representing the design and functioning of a system
- ❖ It is a ISO and standard industry-standard graphical language for specifying, visualizing, constructing, and documenting the artifacts of software systems
- ❖ It was developed by Grady Booch, Ivar Jacobson, and James Rumbaugh and was adopted by the Object Management Group (OMG) in 1997 as the standard for object-oriented modelling
- ❖ Simplifies the complex process of software design

History of UML



Before 95' - Fragmentation



95' - Unification



98' - Standardization



99' - Industrialization

Why is UML?

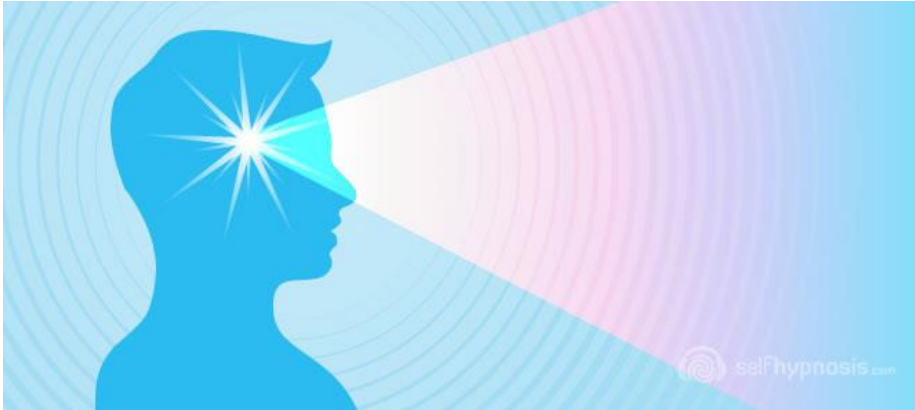
- ❖ Use graphical notation to communicate more clearly than natural language (imprecise) and code(too detailed)
- ❖ Help acquire an overall view of a system
- ❖ Not dependent on any one language or technology
- ❖ Moves us from fragmentation to standardization
- ❖ Currently, UML is very commonly used in software projects

Characteristics of UML

- ❖ A generalized modeling language
- ❖ Different from software programming languages such as Python, C, C++, etc
- ❖ A pictorial language which can be used to generate powerful modeling elements
- ❖ Related to object-oriented designs and analysis
- ❖ Has unlimited applications even outside the software industry. It can be used to visualize the workflow of a factory

UML use for

LOGO



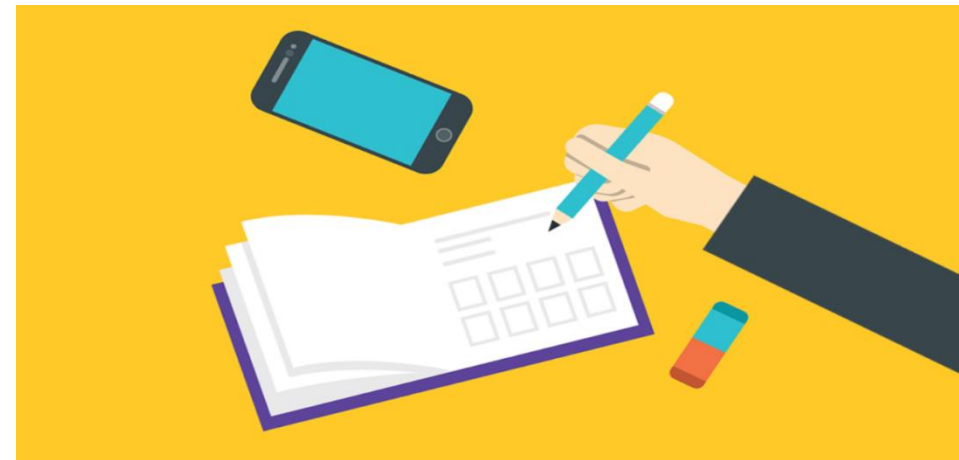
Visualizing



Specifying



Constructing



Documenting

Visualizing

- ❖ Use a rich set of graphical notations to representation of the system
- ❖ The notation system has strict semantics tight, can be understood by many other tools together
- ❖ Help other designers, programmers,.. are understandable

Specifying

- ❖ UML helps build accurate, sufficient and unambiguous models
- ❖ Be used in all stages from analysis, design to to the implementation
- ❖ Use case (for analysis); Class, Sequence, Activity... (for design); Components, Deployment (for deployment)

Constructing

- ❖ UML models can be connected to many languages programming language like C++, Java, C#...
- ❖ Converting models in UML to source code in programming language (Forward engineering)
- ❖ Converting back from source code in a language programming language to UML (Reverse Engineering)
- ❖ Need tool to convert “forward” & “reverse”

Documenting

- ❖ Requirements Specification
- ❖ Architecture Document
- ❖ Design Document
- ❖ Source code
- ❖ Testing Document
- ❖ Prototype Document
- ❖ Deployment Document

UML Tools

- ❖ There are many software tools to support analysis and design using UML
- ❖ Many tools can generate code from UML and vice versa back (code word to UML-Reverse Eng)
- ❖ UML is not a programming language

UML Tools



Visio

**Rational
Rose**



UML
Designer



Visual Paradigm



StarUML

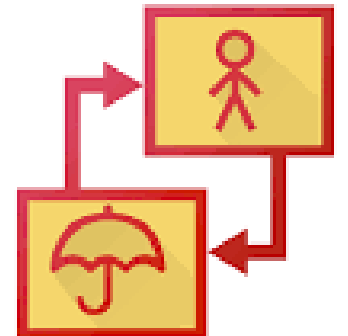


Draw.IO



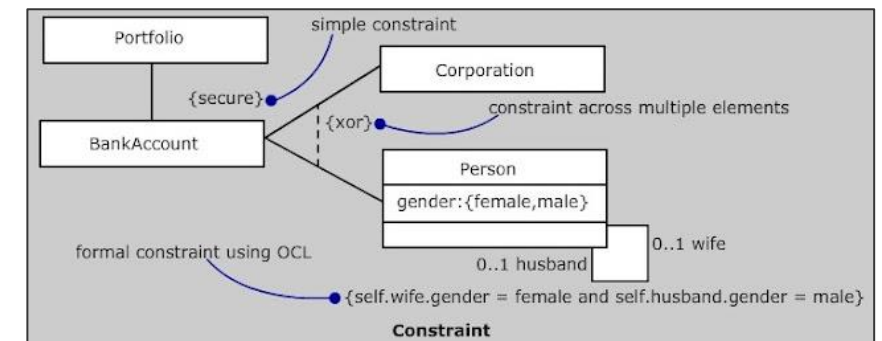
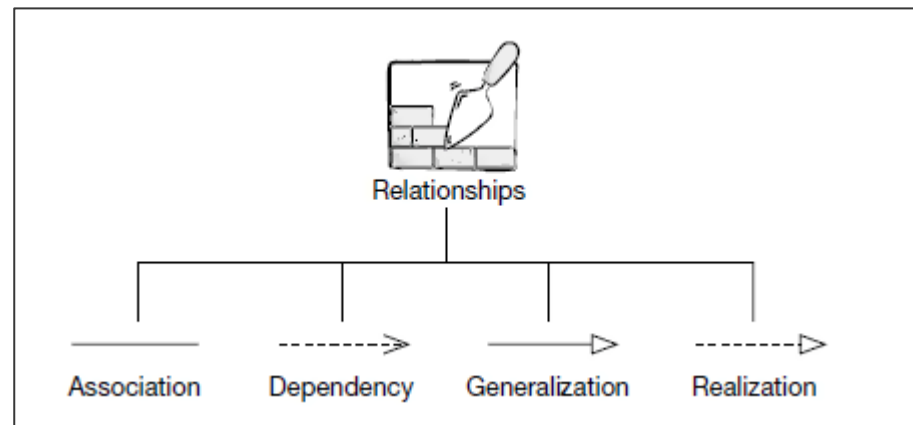
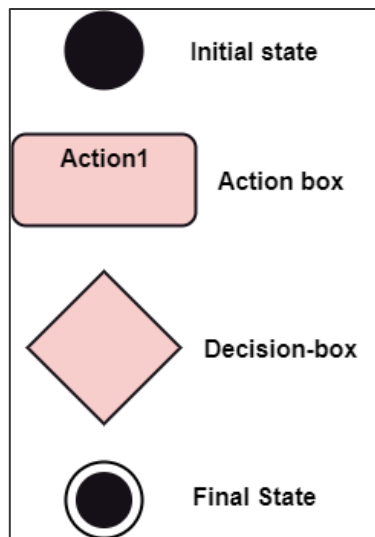
Lucidchart

Umbrello

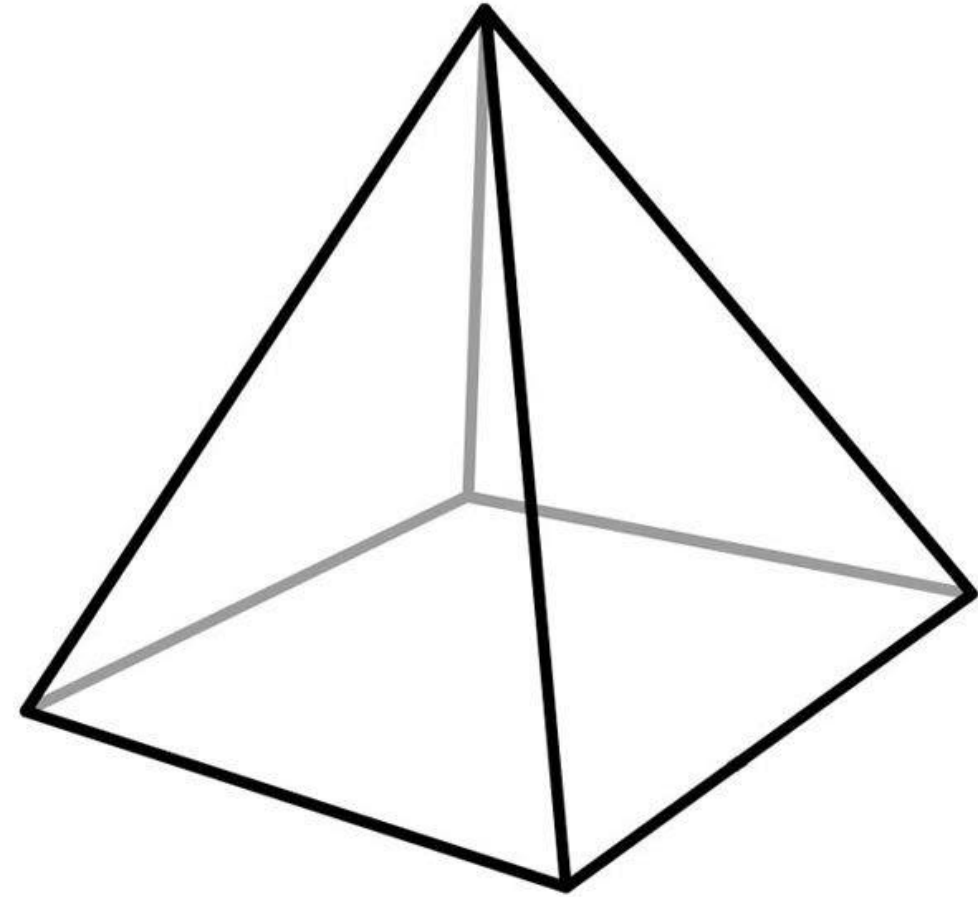
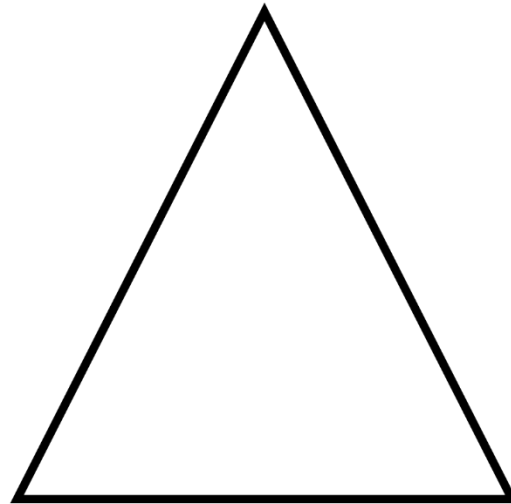
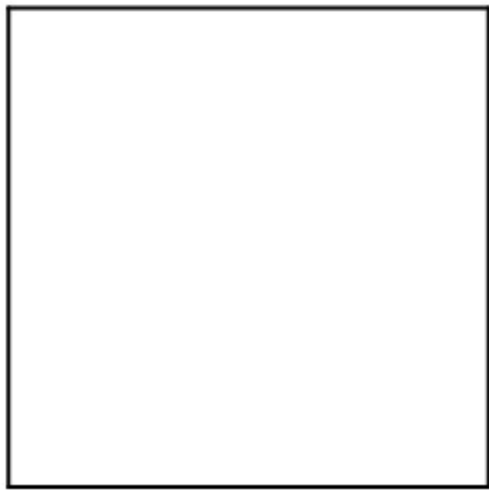


UML component

- ❖ UML building blocks
- ❖ Rules to connect the building blocks
- ❖ Common mechanisms of UML

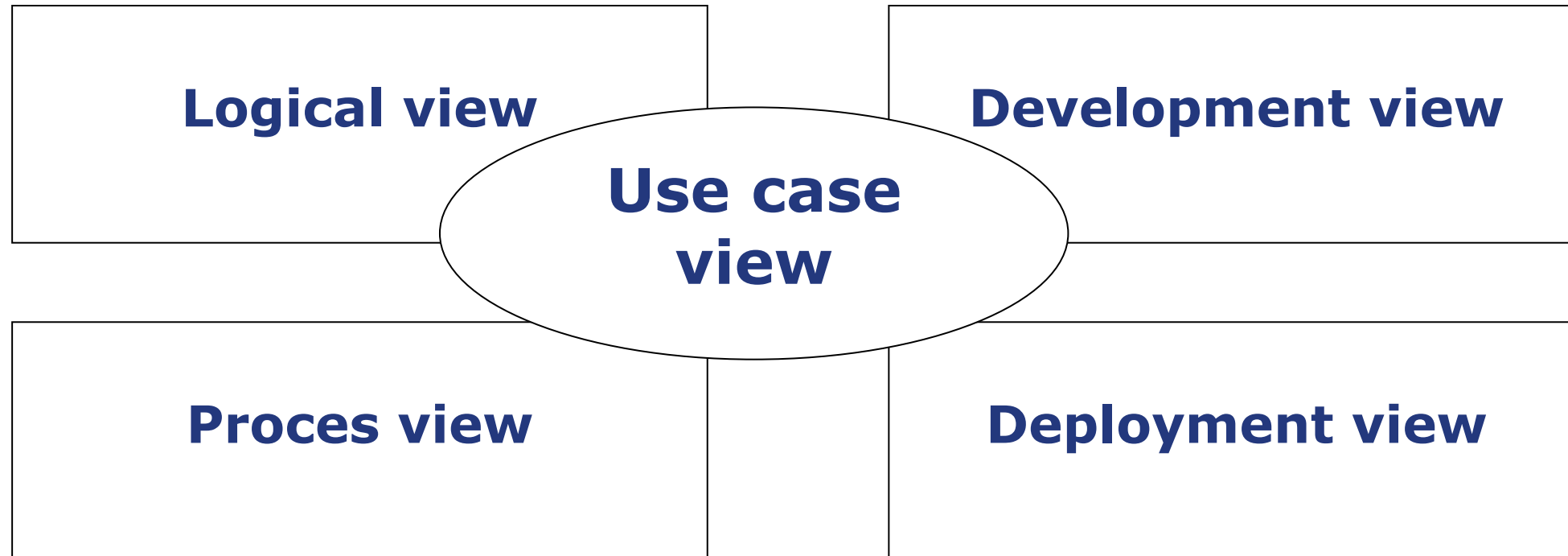


System Views



System Views

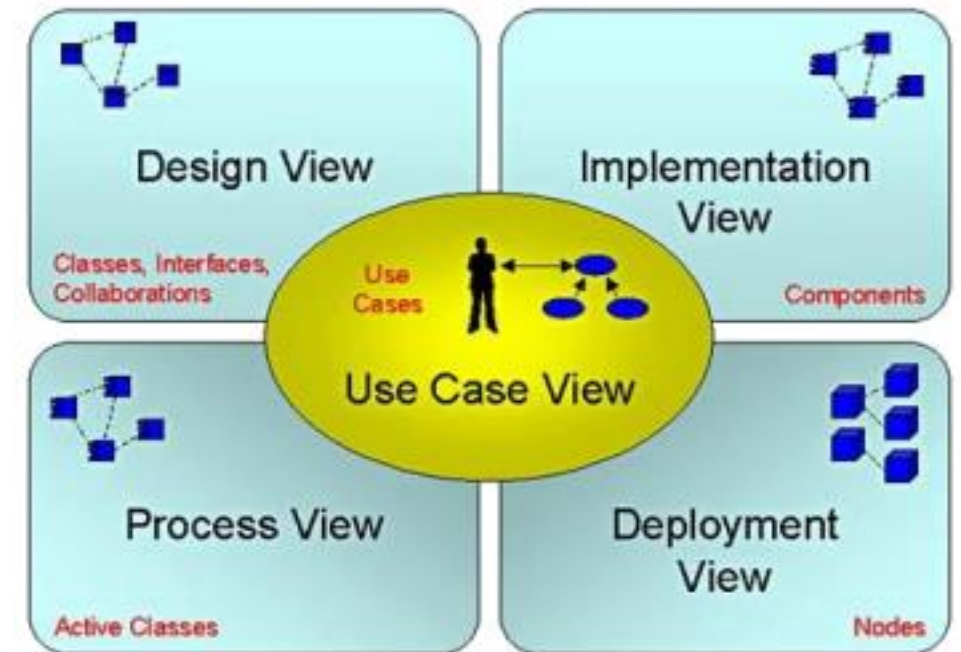
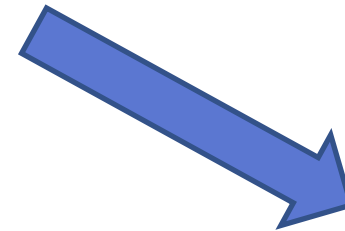
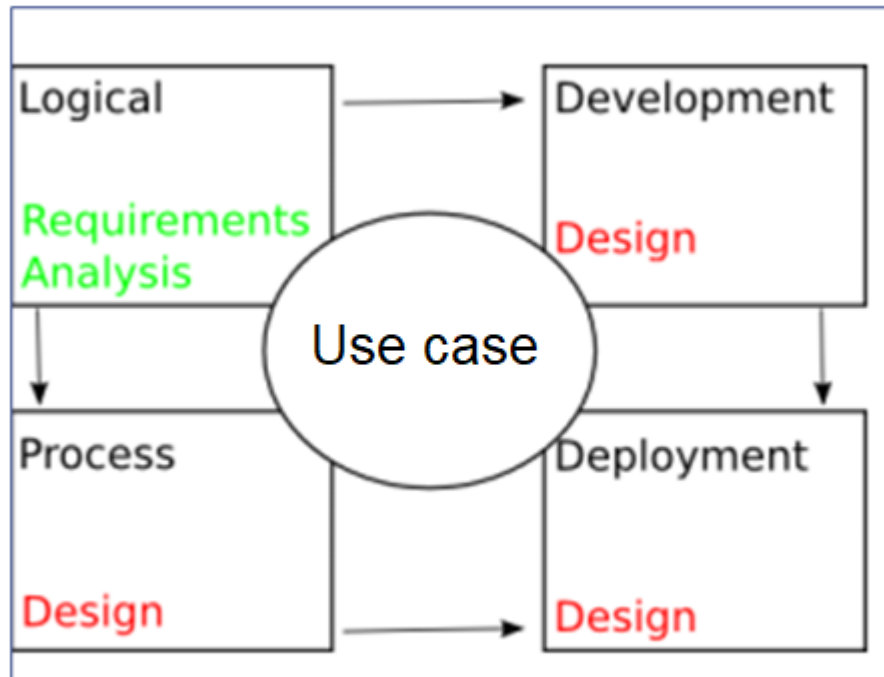
LOGO



System Views

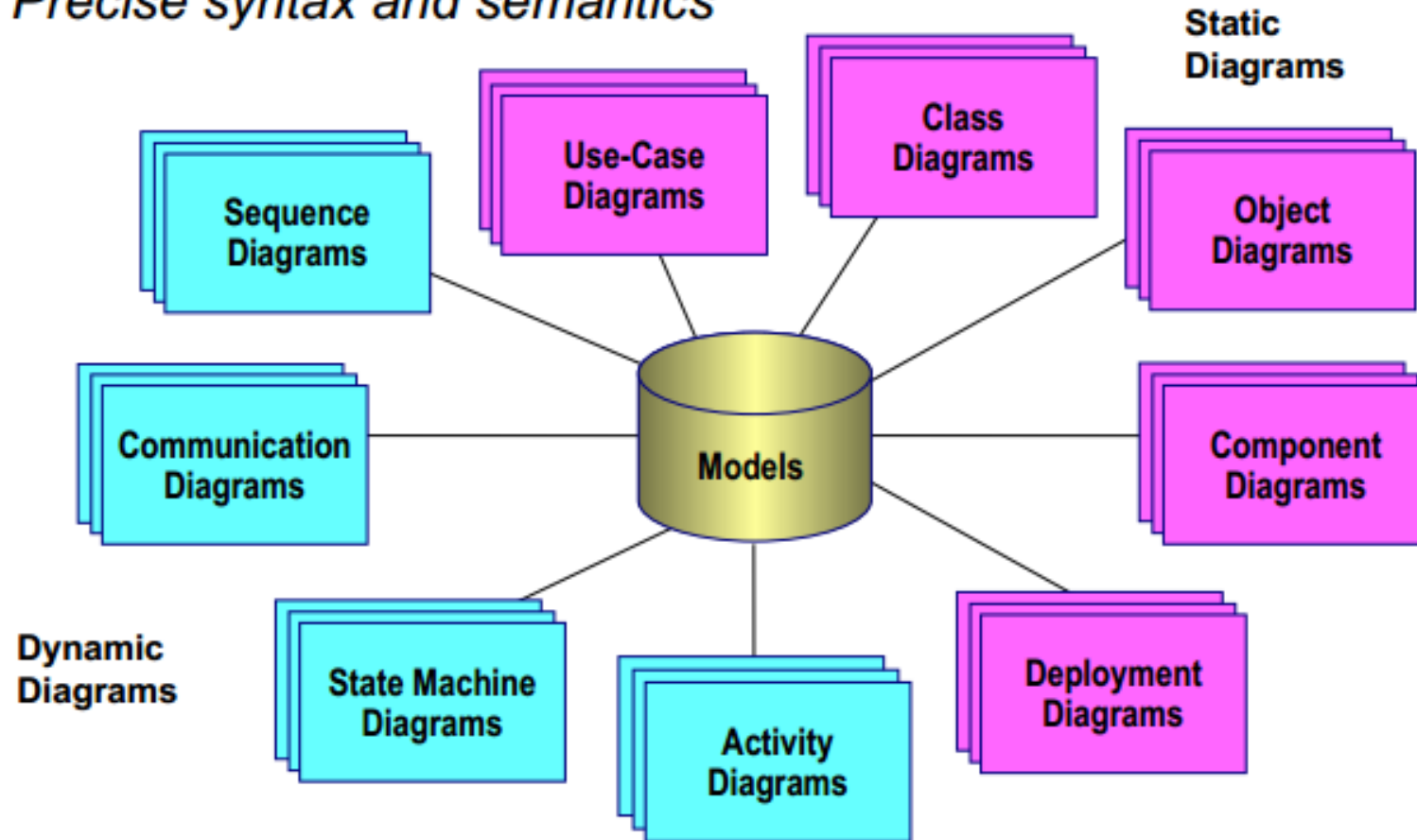
- ❖ **Logical view:** concerned with the functionality that the system provides to end-users
- ❖ **Process view:** dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the run time behavior of the system
- ❖ **Development view:** system from a programmer's perspective and is concerned with software management. This view is also known as the implementation view
- ❖ **Deployment view:** The physical view depicts the system from a system engineer's point of view. It is concerned with the topology of software components on the physical layer as well as the physical connections between these components
- ❖ **Use case view:** The description of an architecture is illustrated using a small set of use cases, or scenarios

System Views

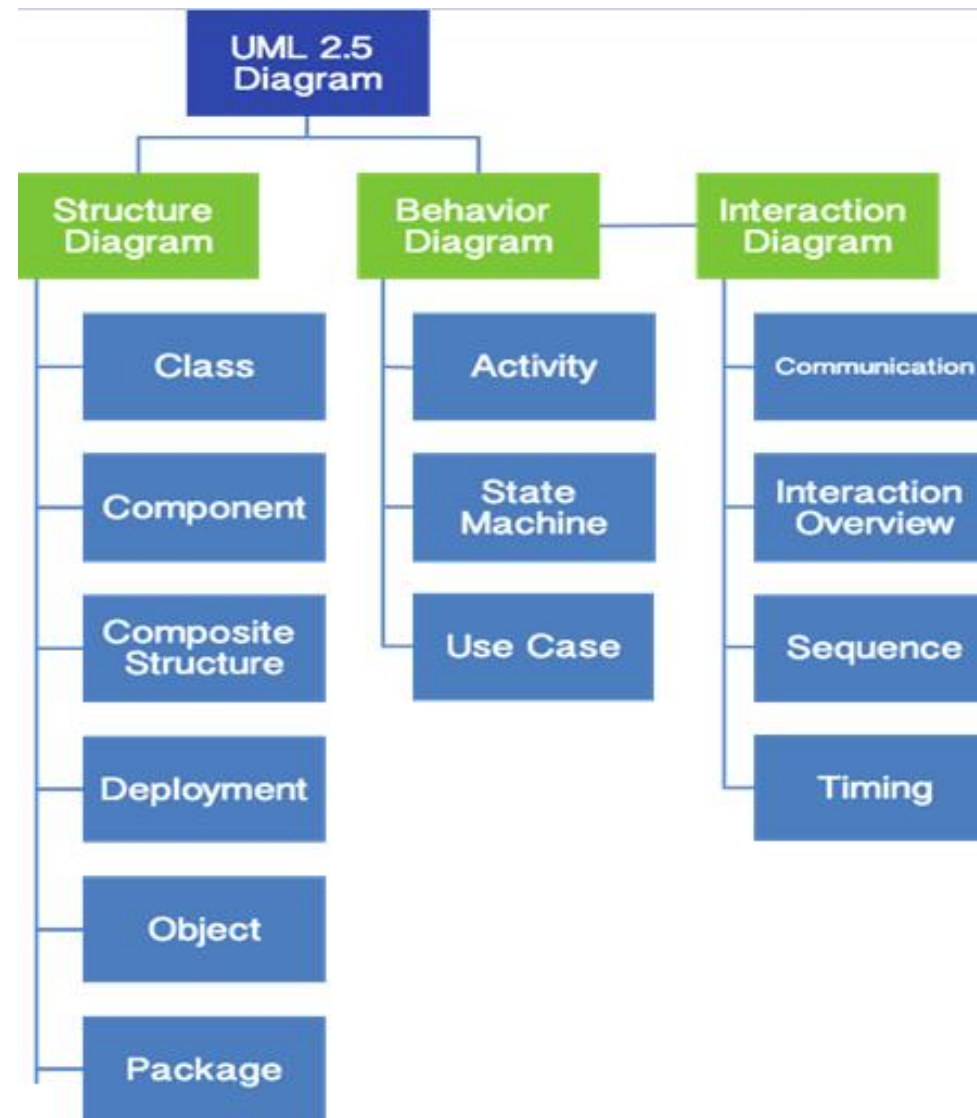


UML diagrams

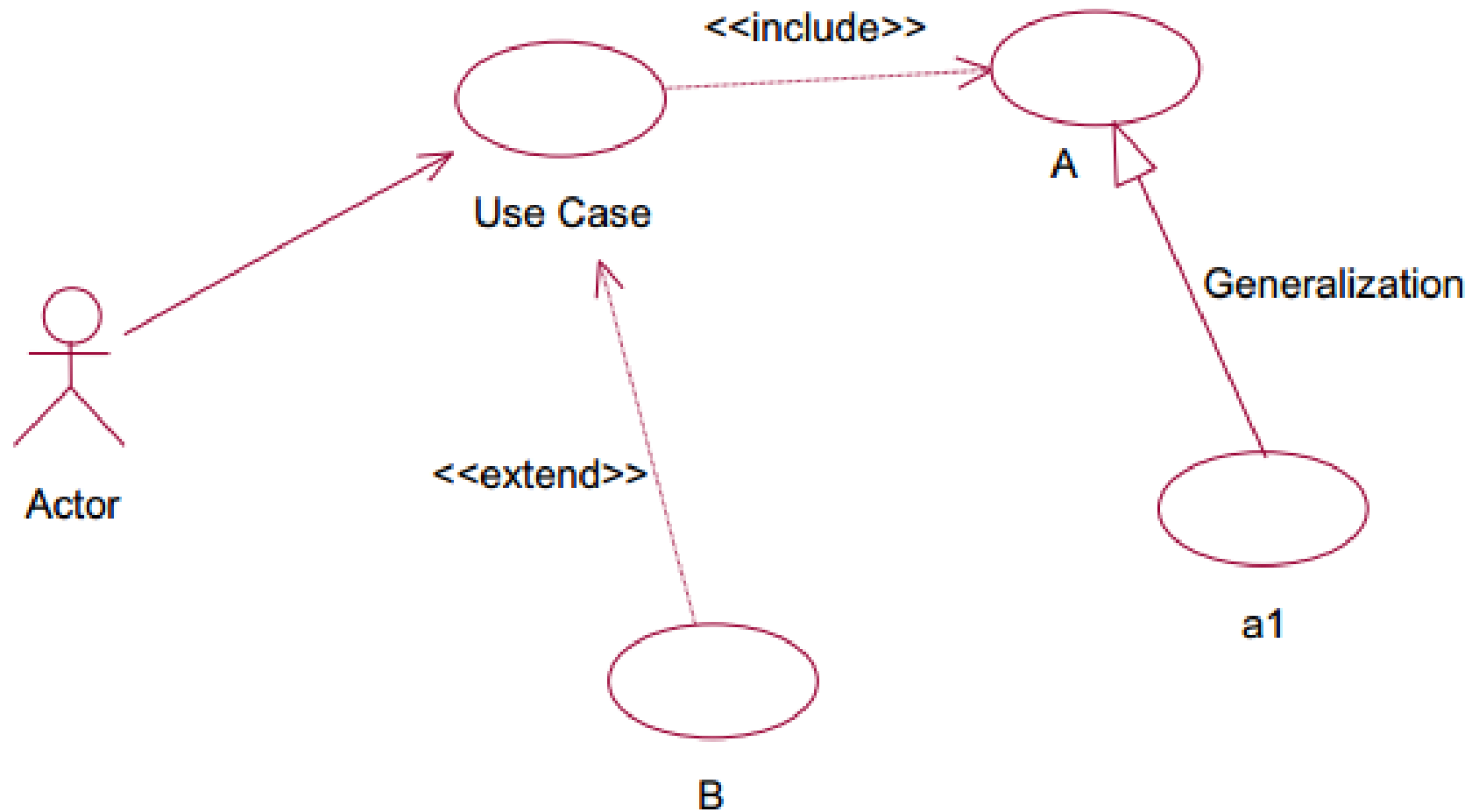
Multiple views
Precise syntax and semantics



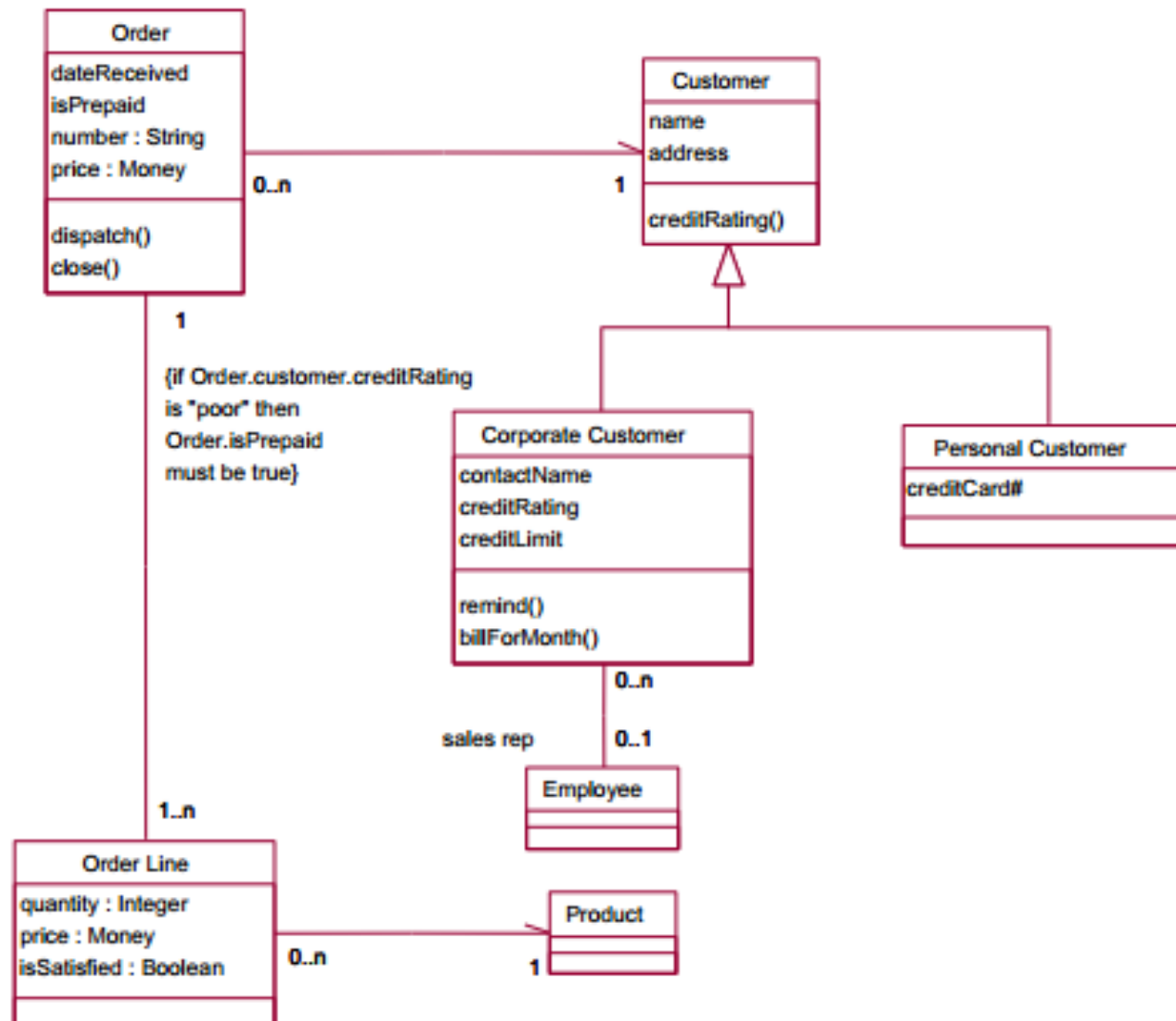
UML 2.5



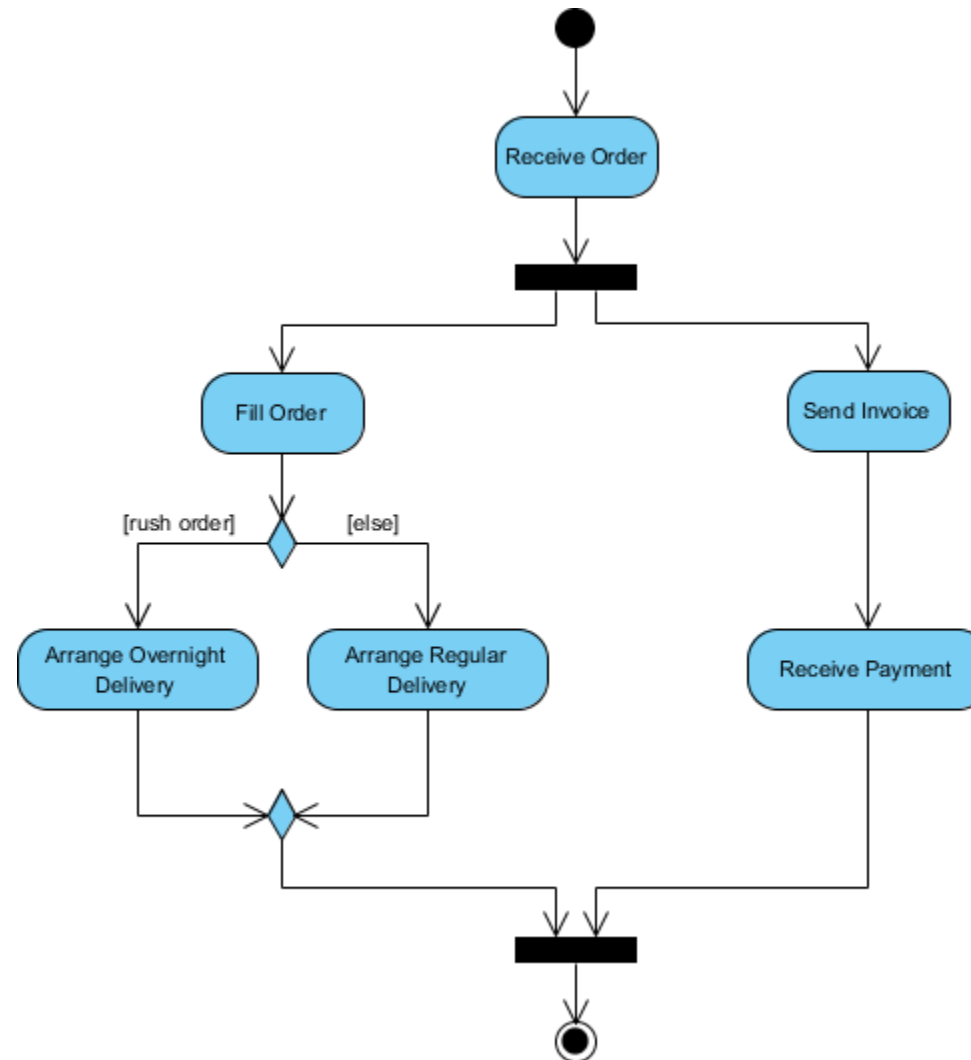
Use case diagram



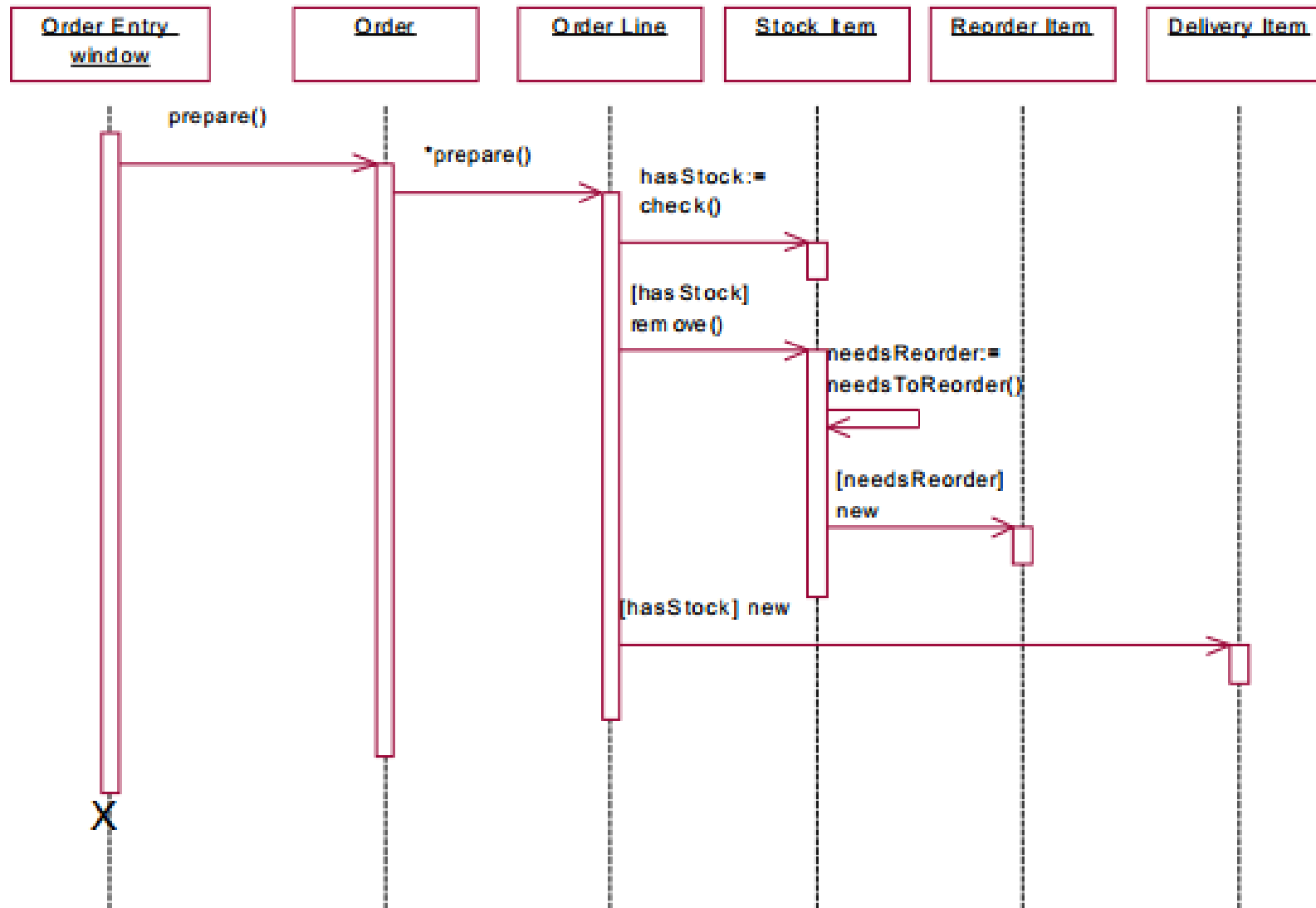
Class diagram



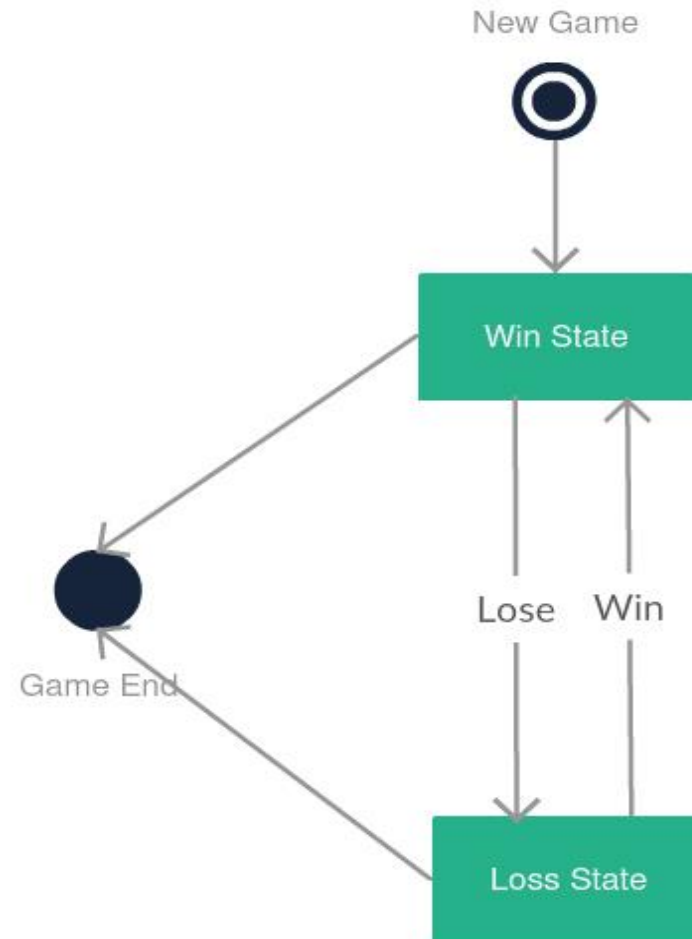
Activity diagram



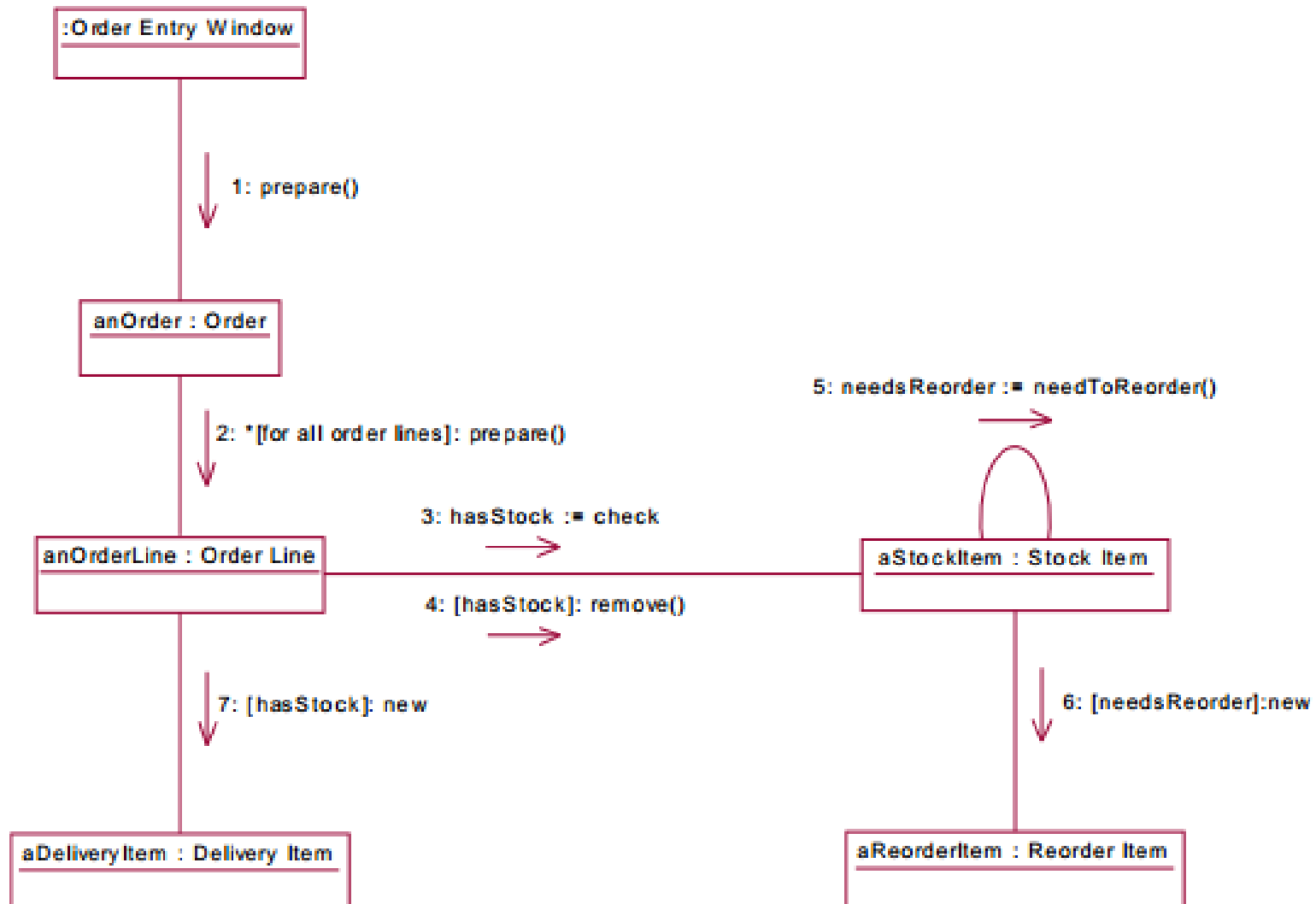
Sequence diagram



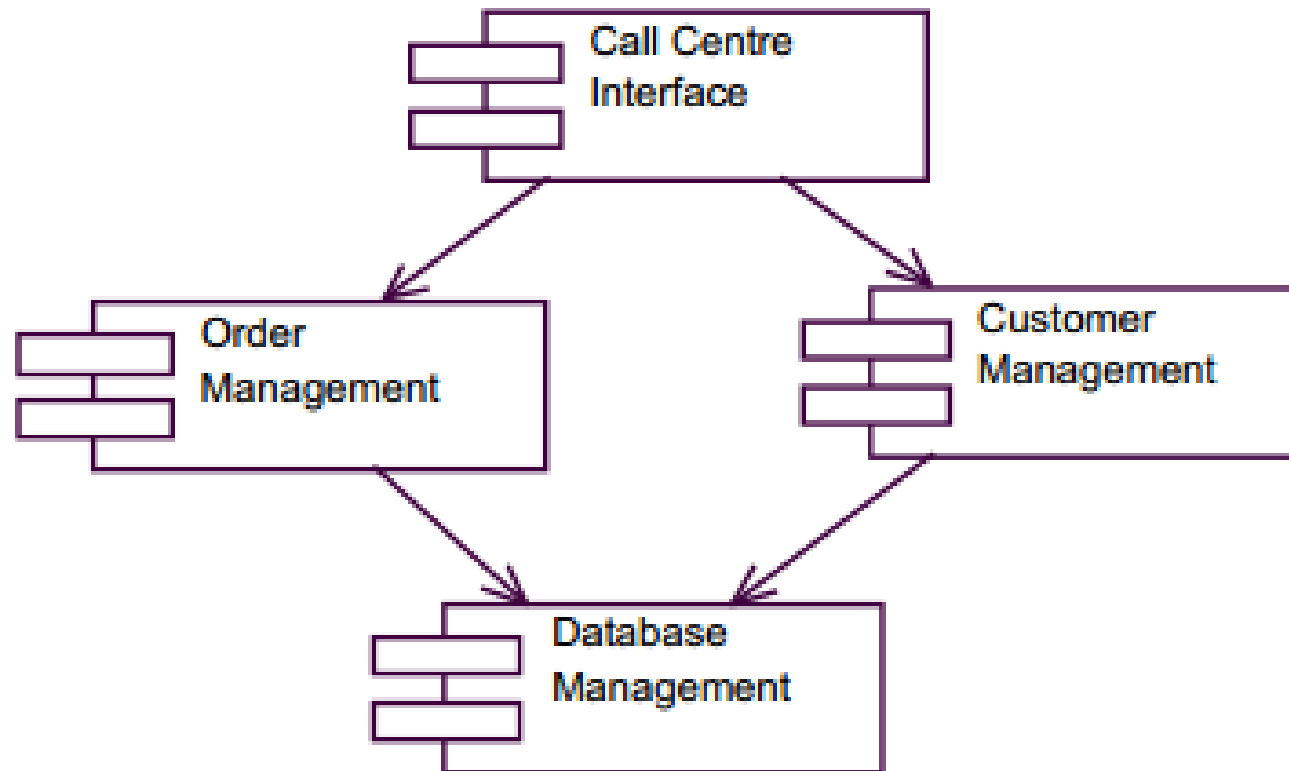
State diagram



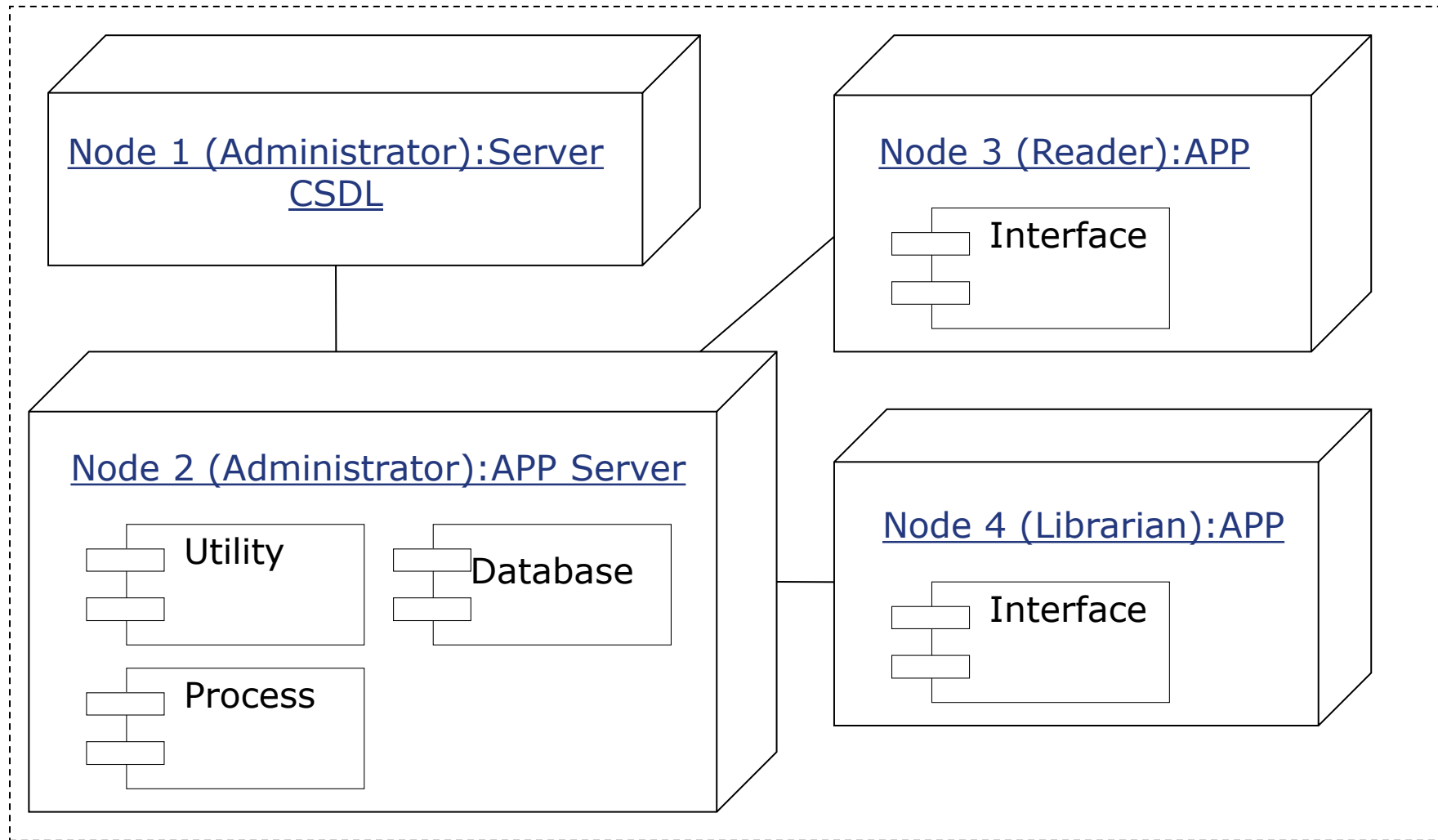
Collaboration diagram



Component diagram



Deployment diagram



Summary

- ❖ Object-oriented software development strategy is to observe the world/system as a set of objects
- ❖ An object-oriented approach that focuses on both information (attribute) and behavior (method)
- ❖ Objects interact with each other with messages (method)
- ❖ Modeling is to decrease complication, share information between related persons, correspond to changing requirement, effective in maintain software
- ❖ The purposes of Analysis and Design are to transform the requirements into a design, evolve a robust architecture and adapt the design to match the implementation
- ❖ Software architecture encompasses a set of significant decisions about selection of the structural elements, interfaces, behavior, subsystems

Summary

- ❖ Analysis stage is derived by all the usual means, such as written requirements statements, interviews, observation, and a study of the system's documentation
- ❖ Design-phase chosen physical architecture, and take into consideration any constraints associated with the environment in which the system must operate, or the technical options available, develop in sufficient detail to provide the necessary input to the implementation
- ❖ ULM is one of the most widely used notations for depicting objects and the way they interact with each other, effectively support the system development process especially the communication between the teams
- ❖ UML components are: building blocks, rules to connect the building blocks and common mechanisms