UML Notation 26/092023

UML (Unified Modelling Language) is used for analysis and design model-making in Object-oriented analysis and design.

It is made keeping two things in mind :

- Syntax (Grammar)
- Semantics (Meaning)

Drawing a diagram in UML doesn’t constitute analysis and design, but it provides aid in clarifying the concept.

UML is not a System design or a development methodology. Its only a language that is used for modelling in analysis and design phase.

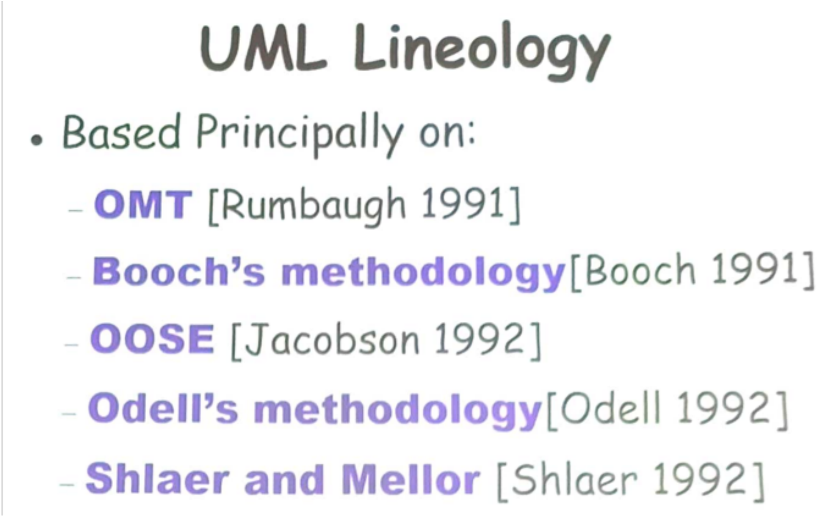
Primary modelling is used to analyze, specify, and design software system.

UML is independent of any design methodology.

UML as a standard

1. Adopted by OMG (Object Management Group) in 1997.
2. ISO adopted UML as a standard in 2005.

UML is also used outside software development. (Means that UML is used in different context than software, like construction).



UML Lineology

- Based Principally on:
  - **OMT** [Rumbaugh 1991]
  - **Booch's methodology**[Booch 1991]
  - **OOSE** [Jacobson 1992]
  - **Odell's methodology**[Odell 1992]
  - **Shlaer and Mellor** [Shlaer 1992]

Different versions of UML

UML 1.0 ⇒ Read-only

UML 1.X ⇒ Read and Write.

UML 3.0 ⇒ Read, Write and Execute

CASE Tools

CASE = Computer Aided Software Engineering

- Use case tools for drawing UML model
- Given the design, a code for UML model can be generated by using CASE tools.

Why are UML models required?

- Model = abstraction (consider few important aspects and ignores all unnecessary ones).
- UML is easy to understand
- **UML is graphical modelling language.**

Different views of UML Modelling

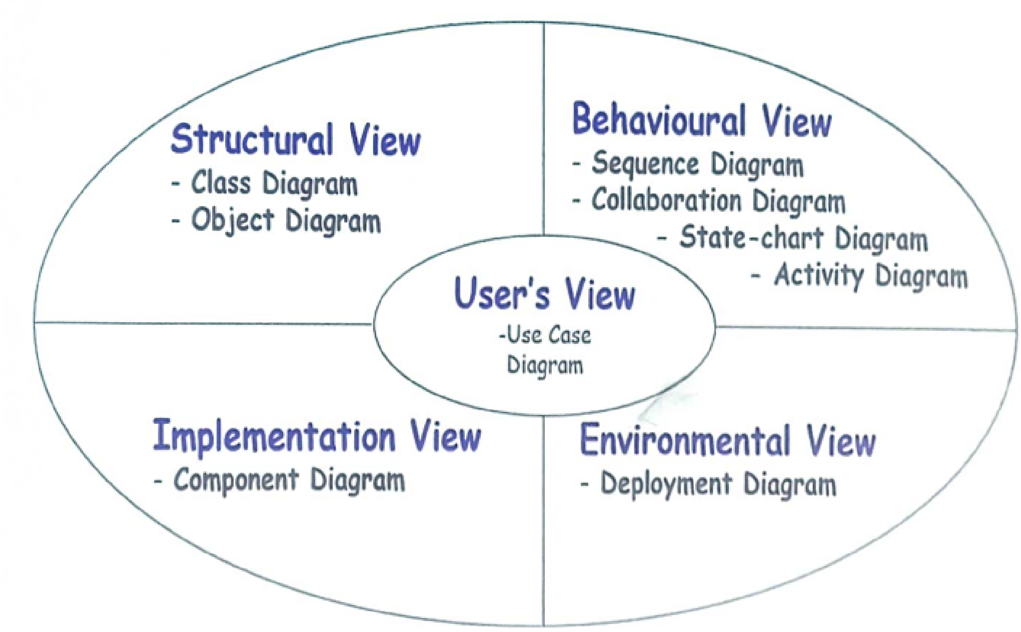
- There are total 13 UML diagrams categorized into different views.

- If all the views captured in one single model, then very high complexity and implementation will be then difficult. Hence, we make several models for different views.
- What are views?
    - Different models provides different perspectives (views) of the software system.

There ae five important views or perspectives :

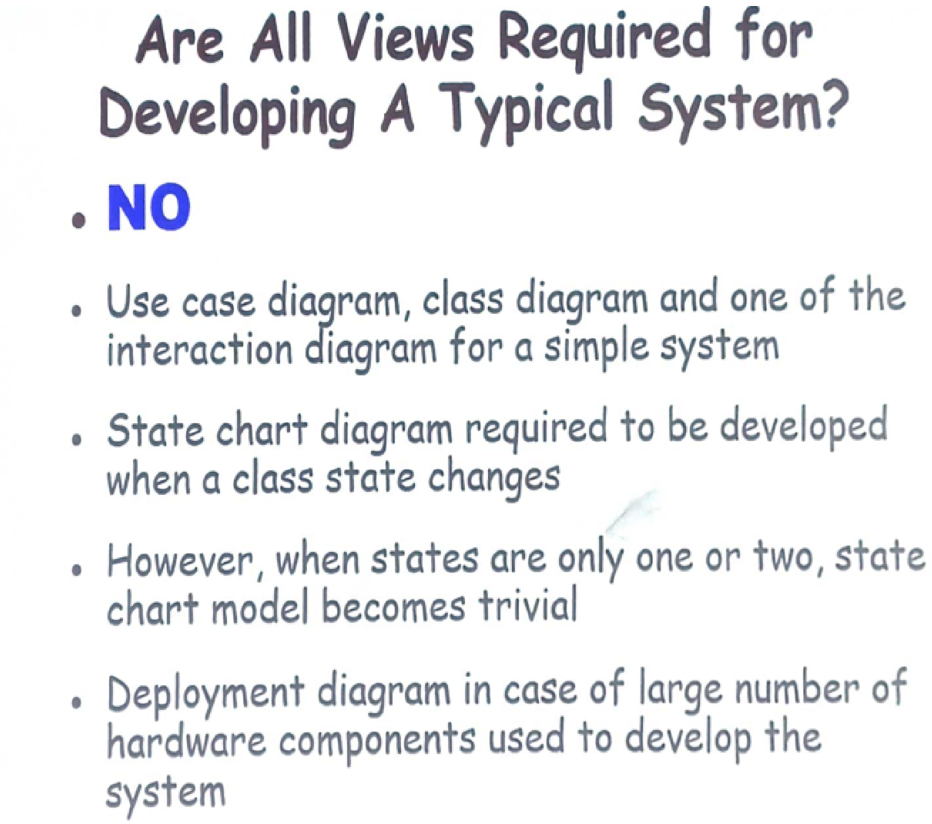
- User view
- Structure View
- Behavioral View
- Implementation View
- Environmental View

> 💡 User, structure and behavioral view for before implementation and implementation and environmental view after implementation.



Description of various views

1. **User's view :**
    - Captures views in terms of functionalities of system (called functional model diagram)
    - It is user-oriented and not developer oriented.
    - It is the only functional model and all other views are object views.
    - Blackbox view
    - It is the central view ⇒ all other views must conform to this.
2. **Structure View :**
    - Structure of problem (objects and classes) and solution (objects and classes)
    - Captures relation between classes (objects).
    - Also called static model as it doesn't change with time.
3. **Behavioral View :**
    - Captures time dependent dynamic behavior
    - How objects interact with each other in time to realize system behavior.
4. **Implementation View :**
    - Captures Important component of the system and their interdependencies.
    - Example :
        - It may show GUI part, database part and middleware as different components of the system being modelled.
5. **Environmental View :**
    - Captures how different components are implemented on different hardware.
    - Deployment related.



Are All Views Required for Developing A Typical System?

- NO
- Use case diagram, class diagram and one of the interaction diagram for a simple system
- State chart diagram required to be developed when a class state changes
- However, when states are only one or two, state chart model becomes trivial
- Deployment diagram in case of large number of hardware components used to develop the system

Static and the dynamic view

Static View

- Emphasis on static structure of the system using objects, attributes, operations and relationships.
- Includes class diagram, composite structure diagrams.

Dynamic View

- Showing collaborations among objects and changes in the internal state of the object.
- Includes sequence diagrams, activity diagrams and state machine diagrams.

Classification chart of different views :

1. Structure Diagrams :
    a. Package Diagram

    b. Component Diagram
    c. Deployment diagram
    d. Composite Structure diagram
    e. class diagram
    f. Sequence diagram
    g. Object Diagram

2. Behavior Diagrams :
    a. Use case diagram
    b. Activity diagram
    c. State machine diagram
3. Interaction diagrams :
    a. Communication Diagrams
    b. Interaction Overview Diagrams
    c. Timing Diagrams.