Guest Lecture Report

Guest Lecture

Delivered by Pooja Jain Ma’am

Head of the Department, CSE at IIIT, Nagpur

Name: Shivam Tawari

Roll no: A-58

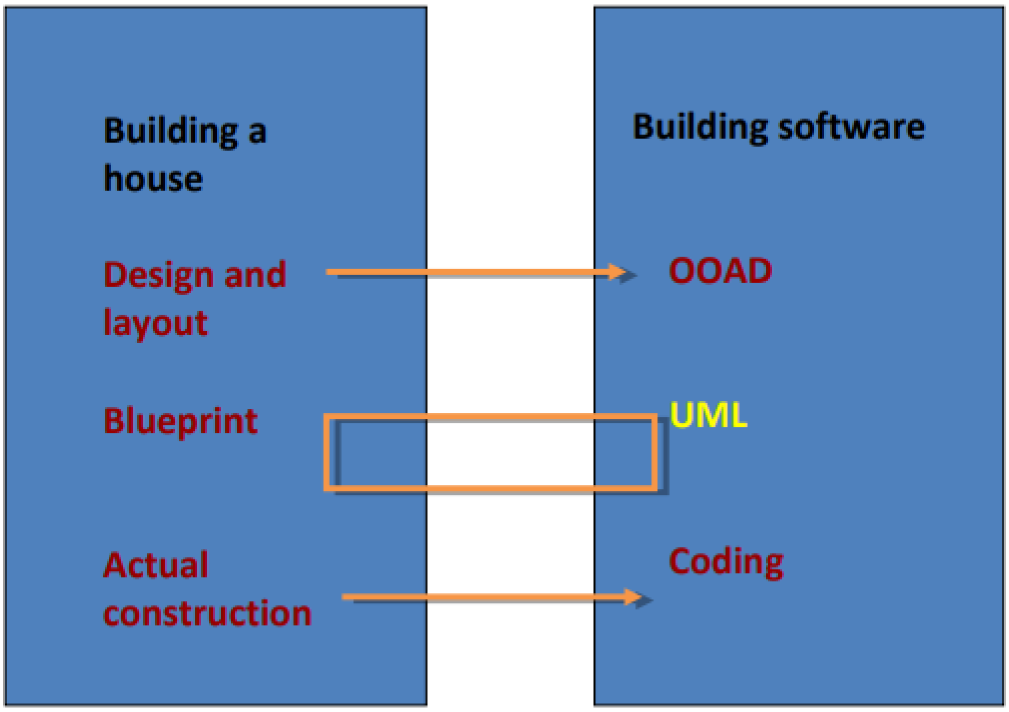
Subject: Software Engineering & Project Management

Topic: Object-Oriented Analysis and Design (OOAD)

The Object-Oriented Modelling Approach:

- Benefits
  1. The ability to tackle more challenging problem domains
  2. Improved communication among users, analysts, designers, and programmers
  3. Reusability of analysis, design, and programming results
  4. Increased consistency among the models developed during object-oriented analysis, design, and programming
- Object-Oriented Systems Development Life Cycle
  1. Process of progressively developing representation of a system component (or object) through the phases of analysis, design, and implementation
  2. The model is abstract in the early stages.
  3. As the model evolves, it becomes more and more detailed.

OOAD with UML:



OOAD - UML Analysis Model:

The Unified Modeling Language (UML) is a graphical language for OOAD that gives a standard way to write a software system's blueprint. It helps to visualize, specify, construct, and document the artifacts of an object-oriented system. It is used to depict the structures and the relationships in a complex system.

Systems and Models in UML:

System − A set of elements organized to achieve certain objectives form a system. Systems are often divided into subsystems and described by a set of models.

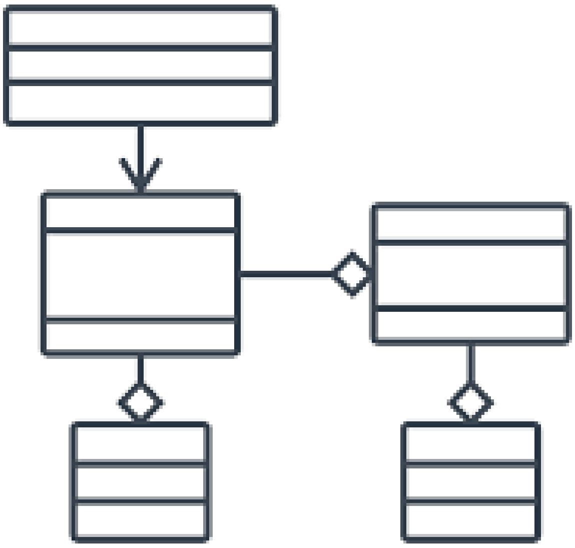
Model − Model is a simplified, complete, and consistent abstraction of a system, created for better understanding of the system.

View − A view is a projection of a system's model from a specific perspective.

Conceptual Model of UML:

The Conceptual Model of UML encompasses three major elements −

- Basic building blocks
- Rules
- Common mechanisms



Basic Building Blocks:

The three building blocks of UML are −

- Things
- Relationships
- Diagrams

Things:

There are four kinds of things in UML, namely −

- **Structural Things** − These are the nouns of the UML models representing the static elements that may be either physical or conceptual. The structural things are class, interface, collaboration, use case, active class, components, and nodes.

- **Behavioral Things** − These are the verbs of the UML models representing the dynamic behavior over time and space. The two types of behavioral things are interaction and state machine.
- **Grouping Things** − They comprise the organizational parts of the UML models. There is only one kind of grouping thing, i.e., package.
- **Annotational Things** − These are the explanations in the UML models representing the comments applied to describe elements.

Relationships:

Relationships are the connection between things. The four types of relationships that can be represented in UML are −

- **Dependency** − This is a semantic relationship between two things such that a change in one thing brings a change in the other. The former is the independent thing, while the latter is the dependent thing.
- **Association** − This is a structural relationship that represents a group of links having common structure and common behavior.
- **Generalization** − This represents a generalization/specialization relationship in which subclasses inherit structure and behavior from super-classes.
- **Realization** − This is a semantic relationship between two or more classifiers such that one classifier lays down a contract that the other classifiers ensure to abide by.

Diagrams:

A diagram is a graphical representation of a system. It comprises of a group of elements generally in the form of a graph. UML includes nine diagrams in all, namely −

- Class Diagram
- Object Diagram
- Use Case Diagram
- Sequence Diagram

- Collaboration Diagram
- State Chart Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram

Rules:

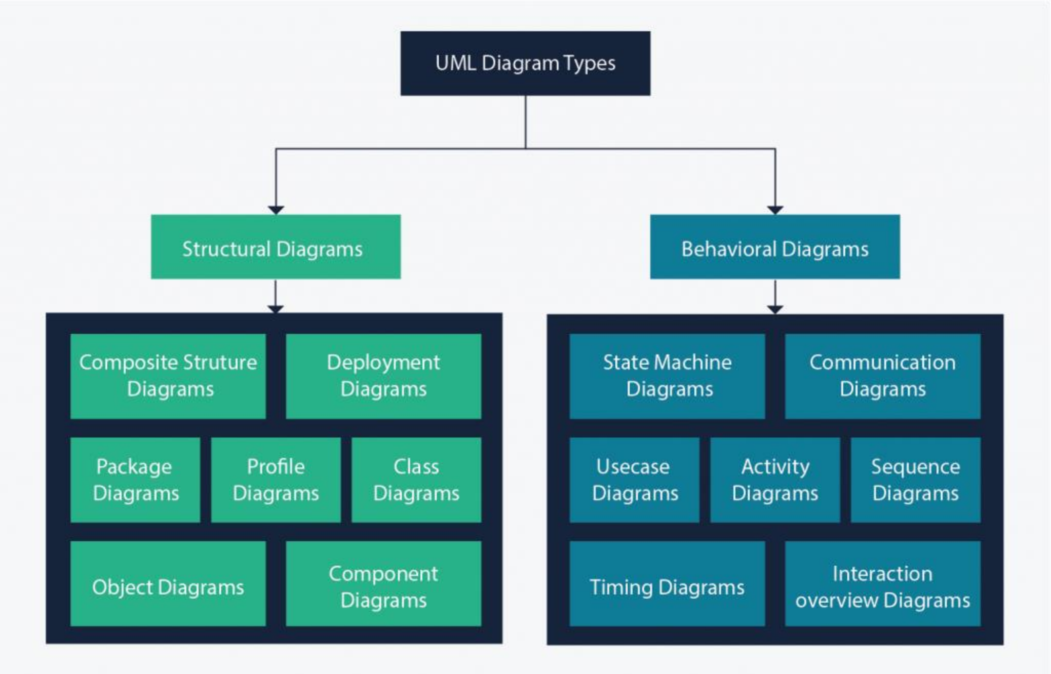
UML has a number of rules so that the models are semantically self-consistent and related to other models in the system harmoniously. UML has semantic rules for the following −

- Names
- Scope
- Visibility
- Integrity
- Execution

Common Mechanisms:

UML has four common mechanisms −

- Specifications
- Adornments
- Common Divisions
- Extensibility Mechanisms



System Design:

Object-oriented system design involves defining the context of a system followed by designing the architecture of the system.

- **Context −** The context of a system has a static and a dynamic part. The static context of the system is designed 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 modelled using **use case diagrams**.
- **System Architecture −** The 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. Typically, a system is partitioned into layers and each layer is decomposed to form the subsystems.