Object Oriented Analysis and Design using Java

Dr H L Phalachandra and Prof: Mahitha G
Department of Computer Science and Engineering

Object Oriented Analysis and Design using Java

Unit-02 : Advanced OO, Object Oriented Analysis and Static Models and Diagrams

Class 16: Requirements, Modelling and Analysis, Introduction to UML

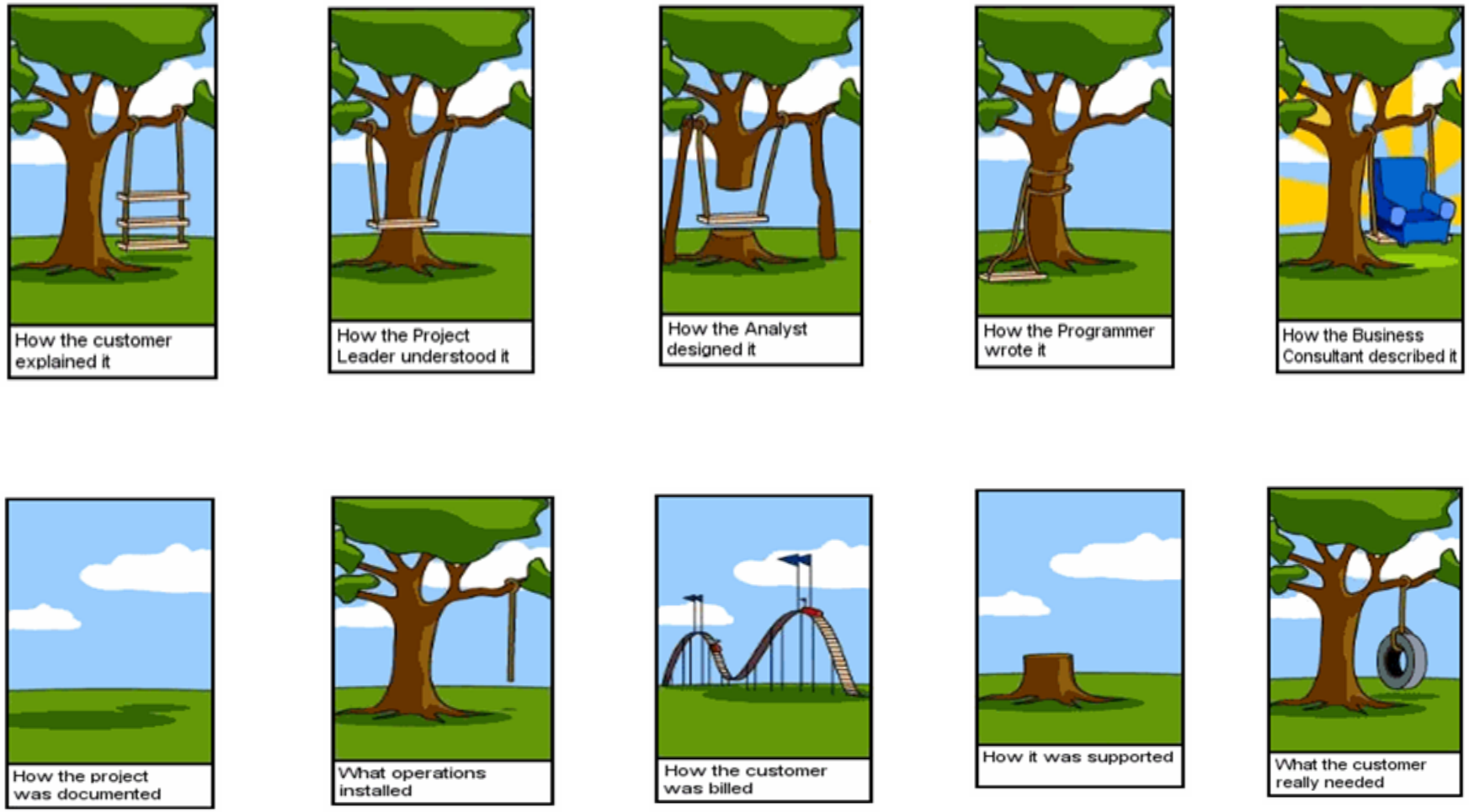
Requirement Engineering

Requirement Engineering is usually the first step in any software intensive development lifecycle irrespective of model

- Usually difficult, error prone and costly
- Critical for successful development of all downstream activities
- Errors introduced during requirements phase if not handled properly will propagate into the subsequent phases
- Unnecessary, Late or invalid requirements can make the system cumbersome or slip
- Requirement errors are expensive to fix at a later stage

Object Oriented Analysis and Design using Java

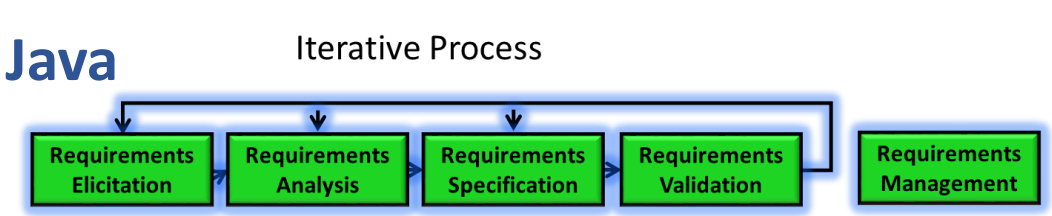
Requirement Engineering



Requirement Engineering

- Requirement is the property which must be exhibited by software developed/adapted to solve a particular problem
- Requirement should specify the externally visible behavior of **what** and **not how**
- Requirements can be looked at as
  - Individual requirements
  - Set of requirements

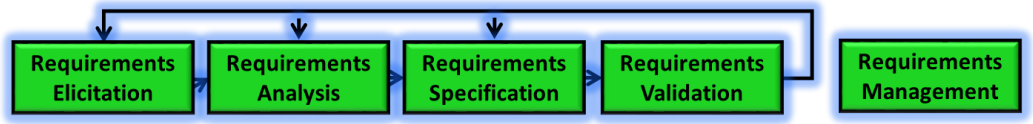
Requirement Engineering



Is a process of working proactively with all stakeholders gathering their needs, articulating their problem, identify and negotiate potential conflicts thereby **establishing a clear scope and boundary for a project**

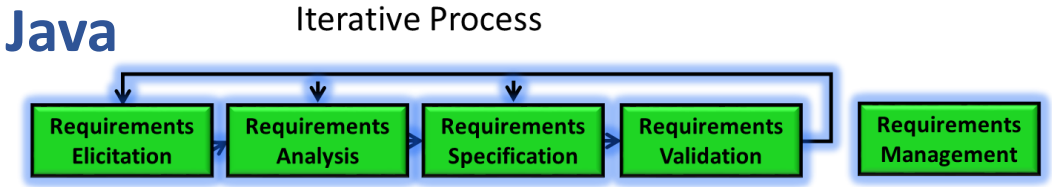
It can also be described as a process of ensuring that the stakeholders have been identified and they have been given an **opportunity to explain their problem and needs** and describe what they would like the new system to do

Requirement Analysis Process



1. **Understand the requirements in depth, both from a product and process perspective**
2. **Classify and Organize the requirements into coherent clusters**
   - **Functional, Non-Functional & Domain requirements**
   - **System and User Requirements**
3. **Model the requirements**
4. **Analyze the requirements (if necessary) using fish bone diagram**

Requirement Analysis Process



5. Recognize and resolve conflicts (e.g., functionality v. cost v. timeliness)
6. Negotiate Requirements
7. Prioritize the requirements (MoSCoW -Must have, Should have, Could have, Wont have)
8. Identify risks if any
9. Decide on Build or Buy (Commercial Of The Shelf Solution) and refine requirements

Requirement Modelling

A Model is a representation of a system in some form.
A is a Model of B if A can be used to answer questions about B
Part of Requirement Analysis and Specification phases.

Couple of important goals of Modelling

- Providing an Understanding (existing) System
  - Analyzing and Validating the requirements in terms of visible requirements within the problem
- Communicating the requirements in terms of who, what and interpreting it in the same way

- **Discussed different kinds of Models**
  - Structural Models and Behavioral models

What is UML

Unified Modeling Language

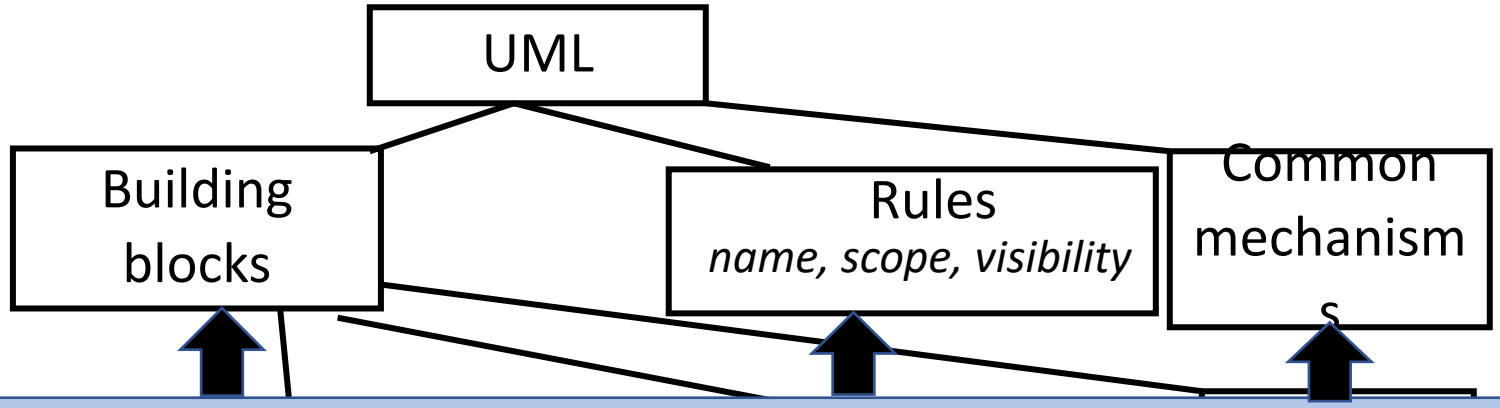
- Language to express different types of models
- Language defines
- Syntax – Symbols and rules for using them
- Semantics – What these symbols represent

UML can be used to

- Visualize (Graphical Notation)
- Specify (Complete and Unambiguous)
- Construct (Code Generation)
- Document (Design, Architecture, etc.)

the artifacts of software-intensive systems

Conceptual Model of UML



UML is made up of three conceptual elements. They are

- Building blocks which make up the UML

Things are abstractions in the model.

There are four kinds of things or abstraction

- Relationship ties together the things or abstractions
- There are four different kinds of relationships

Diagrams : Graphical representation of set of elements, often rendered as a set of connected graph of vertices (things) and arcs (relationships)

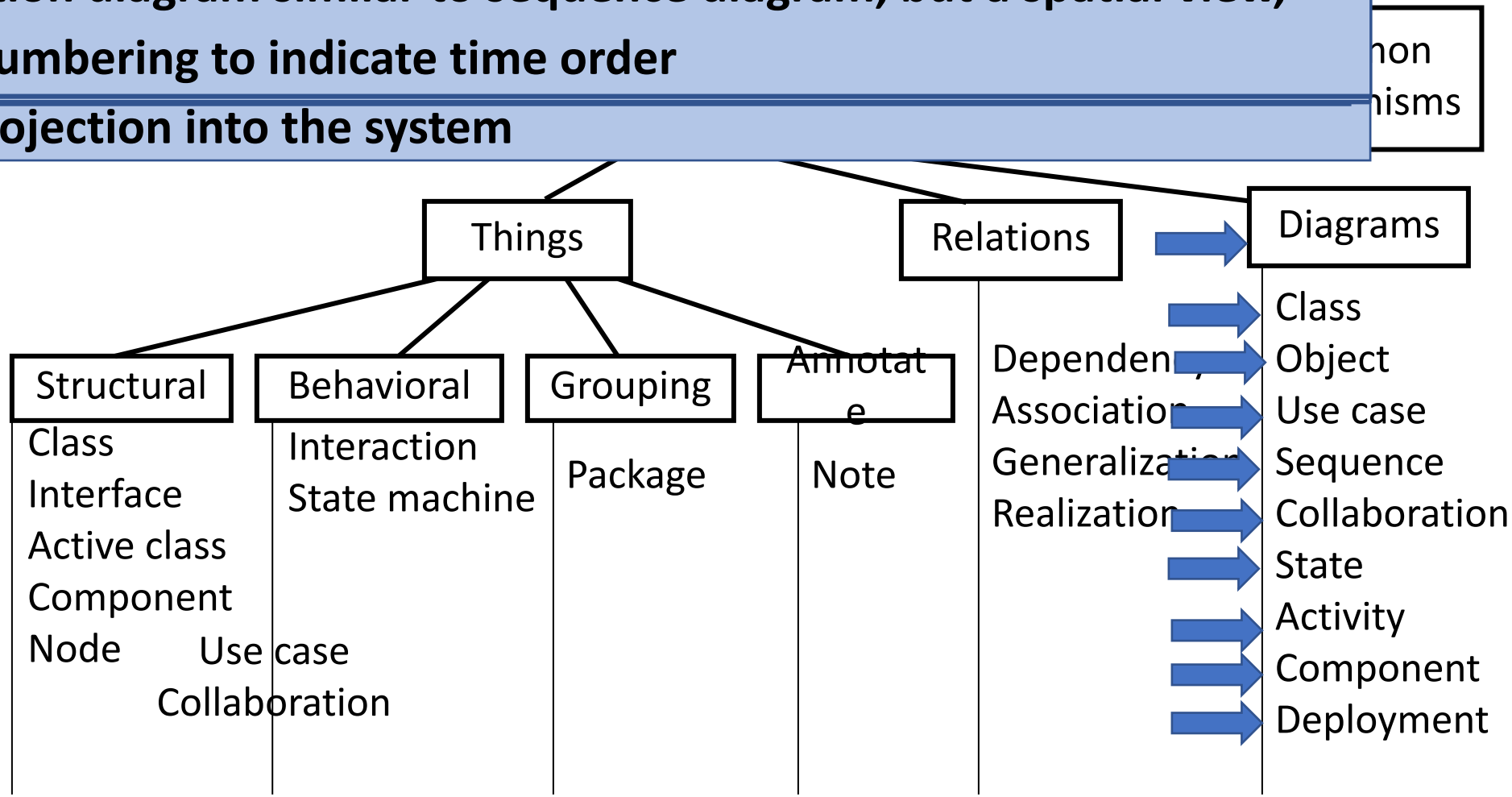
It's a projection into the system

Conceptual Model of UML

Collaboration Diagram

Interaction diagram similar to sequence diagram, but a spatial view, using numbering to indicate time order

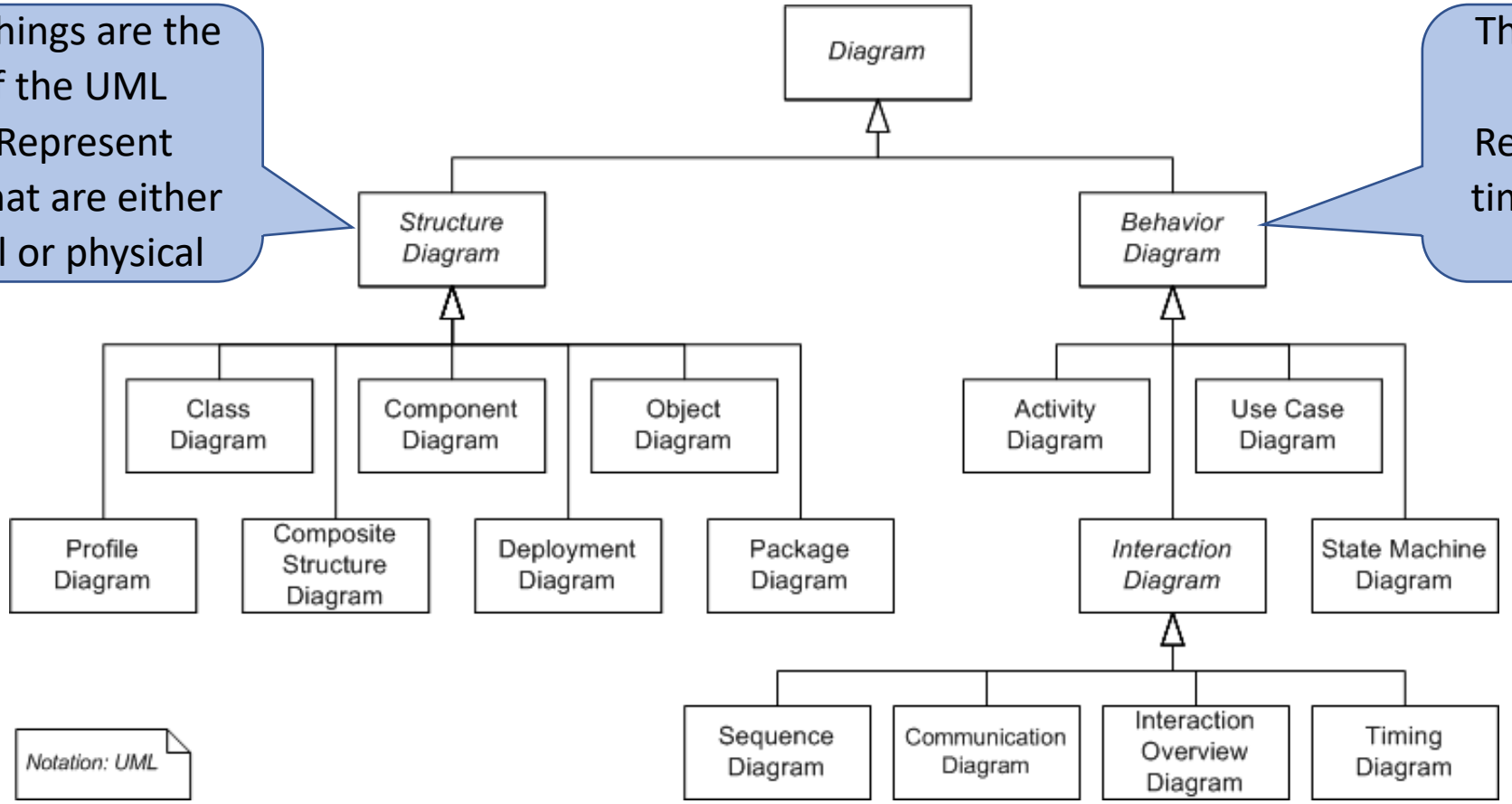
It's a projection into the system



Object Oriented Analysis and Design using Java

UML Diagrams

Structural things are the nouns of the UML models. Represent elements that are either conceptual or physical



These are dynamic parts of the UML models. Represent behavior over time and space. Typically verbs of the model

UML Diagrams – Repeated-In Summary

- Use Case Diagram
  - Shows actors, use cases, and relationships
- Class Diagram
  - Shows classes, interfaces, relationships
  - Most common – address static view
- Object Diagram
  - Shows objects (instances)
- Component Diagram
  - Shows deployable components, including interfaces, ports, and internal structure
- Activity Diagram
  - Shows processes, including flow of control and data

UML Diagrams – Repeated-In Summary

- State Diagram
  - Shows states, transitions, events, and activities depicting the dynamic view of internal object states
- Sequence Diagram
  - Shows interactions between objects as a time-ordered view
- Collaboration Diagram
  - Similar to sequence diagram, but a spatial view, using numbering to indicate time order
- Deployment Diagram
  - Shows the configuration of run-time processing nodes and the components that are deployed on them

THANK YOU

Mahitha G
Department of Computer Science and Engineering

mahithag@pes.edu