
SWA – Rationale for SWA & Modeling

Lecture 02

BIL428 Software Architectures

Asst.Prof.Dr. Mustafa Sert

`msert@baskent.edu.tr`

Department of Computer Engineering, Başkent University

Ankara 06810 TURKEY

Contents

2

- What is the rationale for SWA ?
- Common misunderstandings..
- Relation of SWA Design to SWE
- Modeling Architecture

1 - Abstract Specification

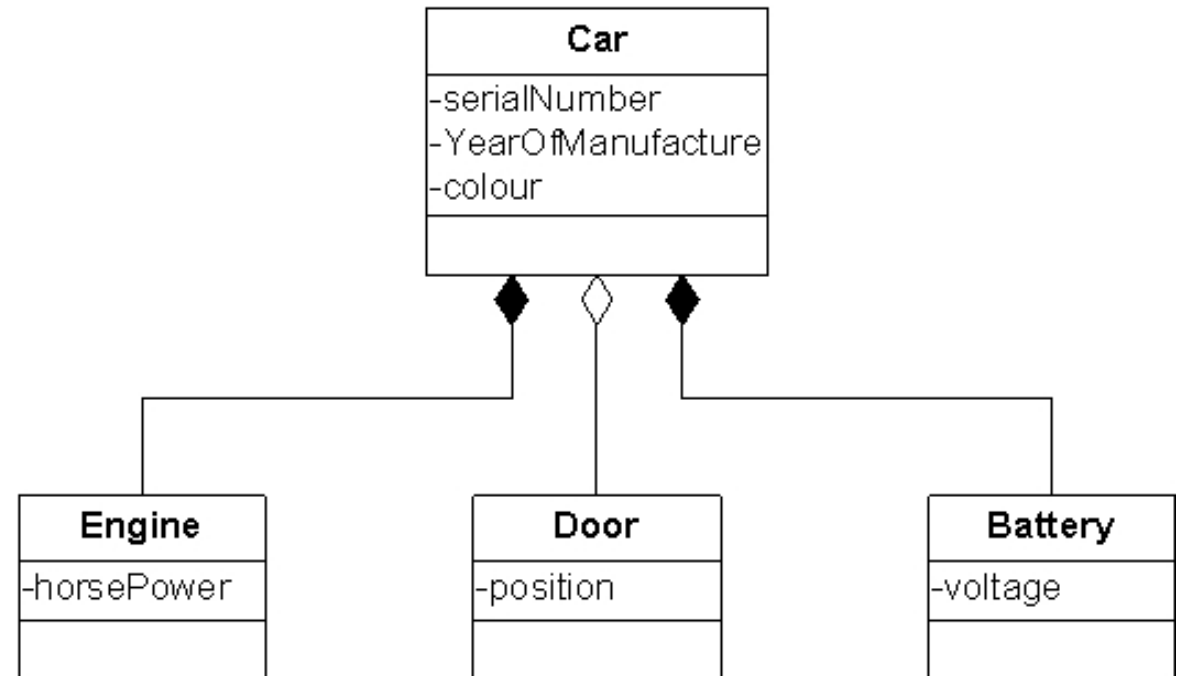
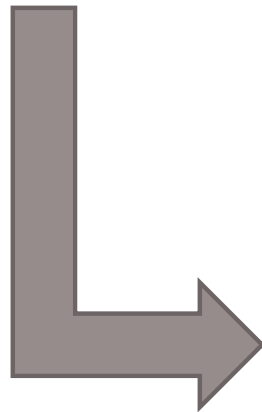
3

- Abstraction
 - ▣ Focus on only relevant properties of the problem
 - ▣ “Ignore” details



Jackson Pollock, The She-Wolf, 1943

Abstract Specification



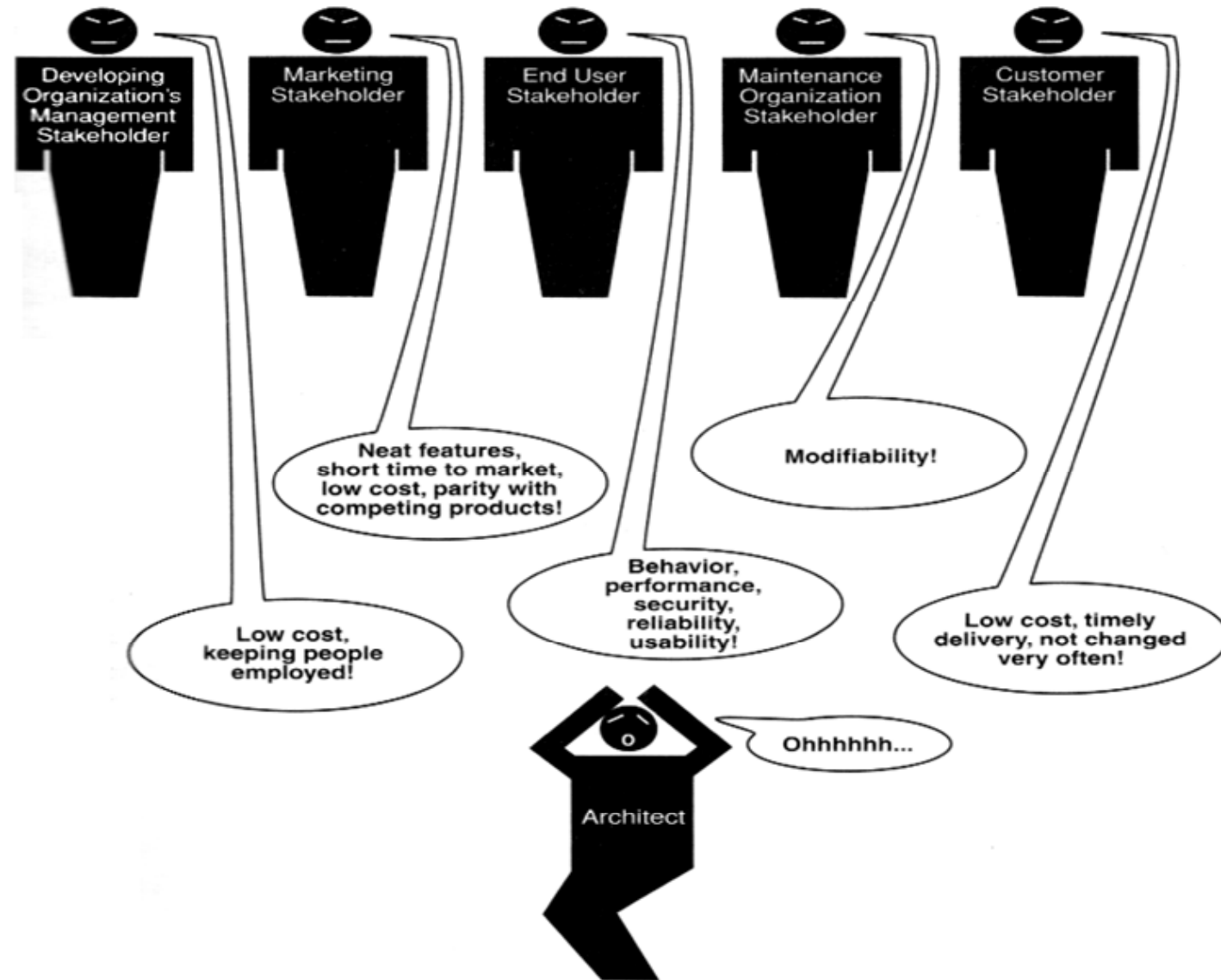
Abstract Specification

5

- Architecture represents a **high-level abstract specification**
- Abstraction helps to cope with **complexity**
- Abstraction **improves understanding** of the software systems

2 – Stakeholder Communication

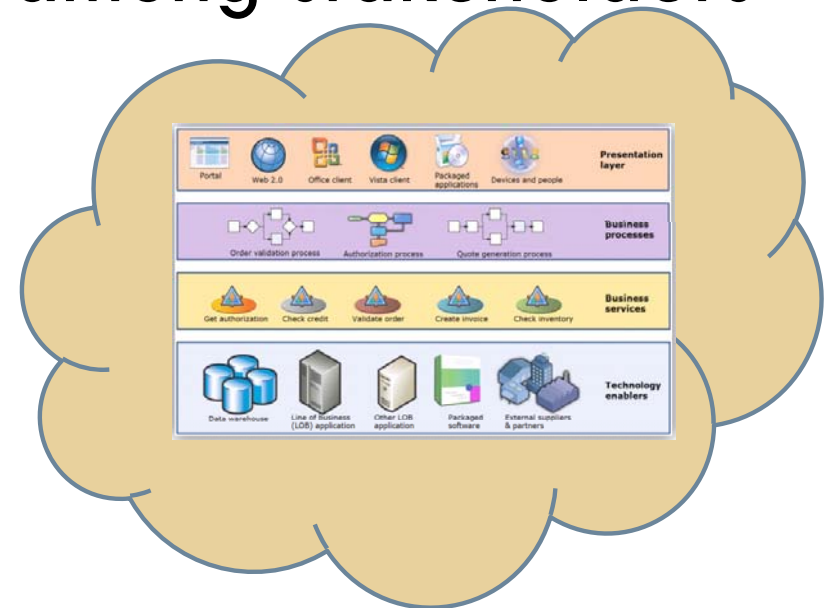
- Stakeholder is any person who has interest in the architecture



Stakeholder Communication

7

- Software architecture provides **a common medium for communication** among stakeholders



3 – Coping with Evolution

8

- ~ 80% cost of a software system occurs after initial deployment
- Software systems change over their lifetimes, very often!!!
- Changes can be grouped as
 - ▣ Local → change to a single element
 - ▣ Non-Local → change to multiple elements, but leaves architecture intact
 - ▣ Architectural → Change is systemic, and affects the overall structure..

Coping with Evolution

9

- Architecture can help dealing with changes and evolution
- In case of proper architecture definition, the changes will be limited to the abstraction boundaries..
- Architecture provides the balance btw fixed and adaptable parts of the system..

4 – Guides Software Development Process

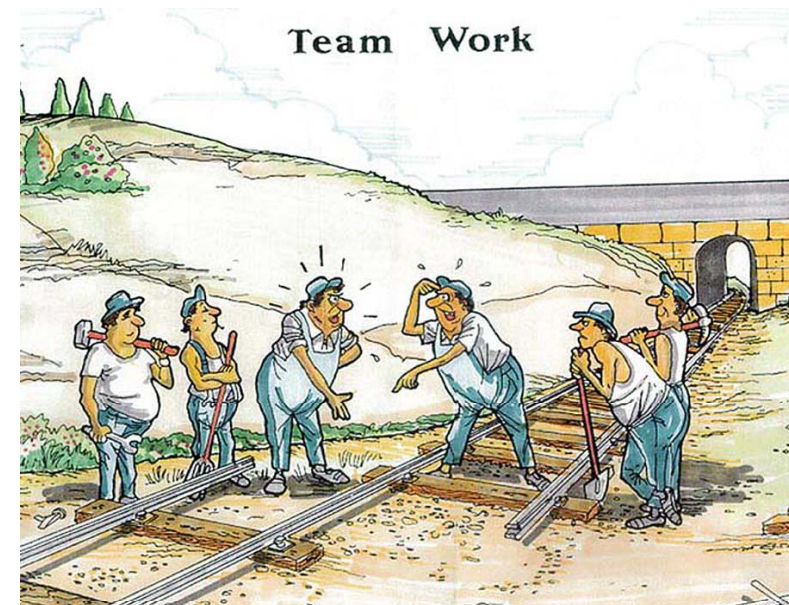
10

- Architecture is explicit
- Focus on architectural components
- A&D based on architectural components

5 – Organization of the Development Project

11

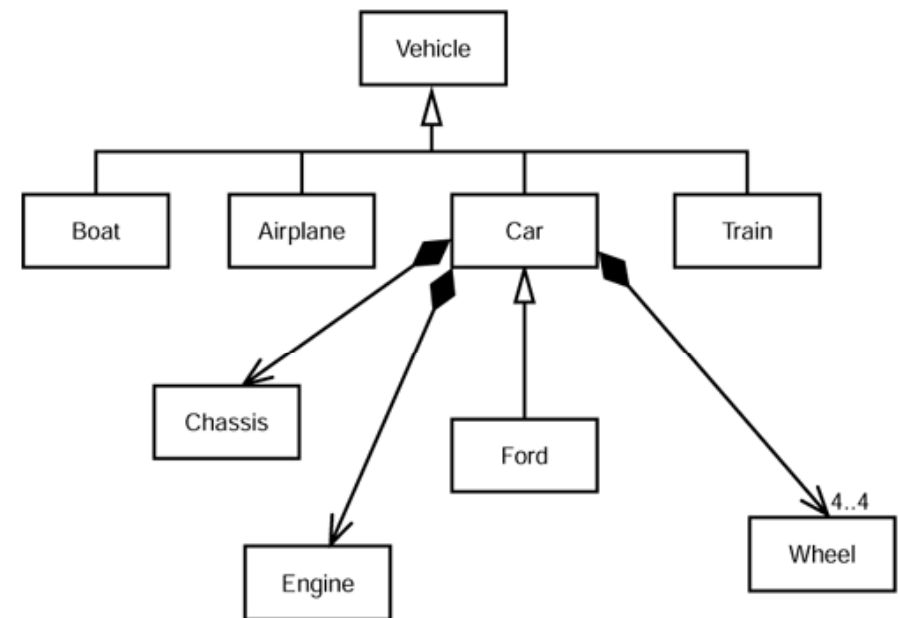
- The architecture influences the organizational structure for development/maintenance efforts..
- Examples
 - ▣ Division into teams
 - ▣ Units for budgeting, planning
 - ▣ Basis of work breakdown structure
 - ▣ Organization of documentation
 - ▣ Basis of integration
 - ▣ Basis of test plans and testing
 - ▣ Basis of maintenance
 - ▣ Incremental development



6 – Large Scale Reuse

12

- Software architecture is an abstract specification
 - ▣ representing set of systems
 - ▣ and as such can be reused for systems exhibiting similar structure and requirements
 - ▣ can promote software product lines



To sum up Rationale for SWA..

13

- ❑ Improved understanding because of high level abstract specification
- ❑ Tool for communication among different stakeholders because of common abstract specification
- ❑ Manifestation of the earliest design decisions
- ❑ Guides software development process
- ❑ Support organization of development project
- ❑ Provides gross level reuse..

Common Misunderstandings

14

- **Architecture is just a documentation!**
 - ▣ Every system has an architecture, either visible or NOT..
 - ▣ The architecture eventually resides within executable code
 - ▣ A system's architecture may be visualized in model, which can be executable!

Common Misunderstandings

15

□ **Architecture vs. Design**

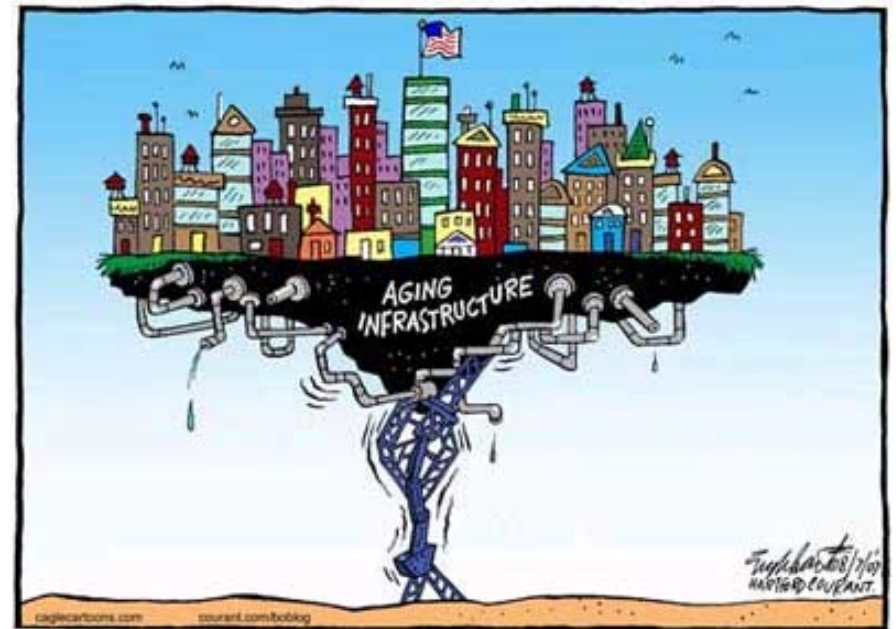
- ▣ An architecture is design, but NOT all design is architecture (e.g., detailed designs)
- ▣ Architecture is at higher abstraction level
- ▣ Software architecture elements such as components and connectors may have detailed designs..

Common Misunderstandings

16

□ Architecture vs. Infrastructure

- Infrastructure is a fundamental part of architecture
- But architecture is more than JUST infrastructure
- Infrastructure could be just a view on the architecture..



National Association of Water
Companies

Common Misunderstandings

17

□ **Architecture vs. Technology**

- ▣ A given technology only serves to implement some dimension of an architecture
 - The network is the architecture
 - The transaction server is the architecture
 - J2EE is the architecture
- ▣ Architecture is more than JUST a list of products
- ▣ Architecture implementation is shaped by technology, BUT a robust architecture is NOT directly bounded by a technology!..

Common Misunderstandings

18

□ **Architecture vs. Structure**

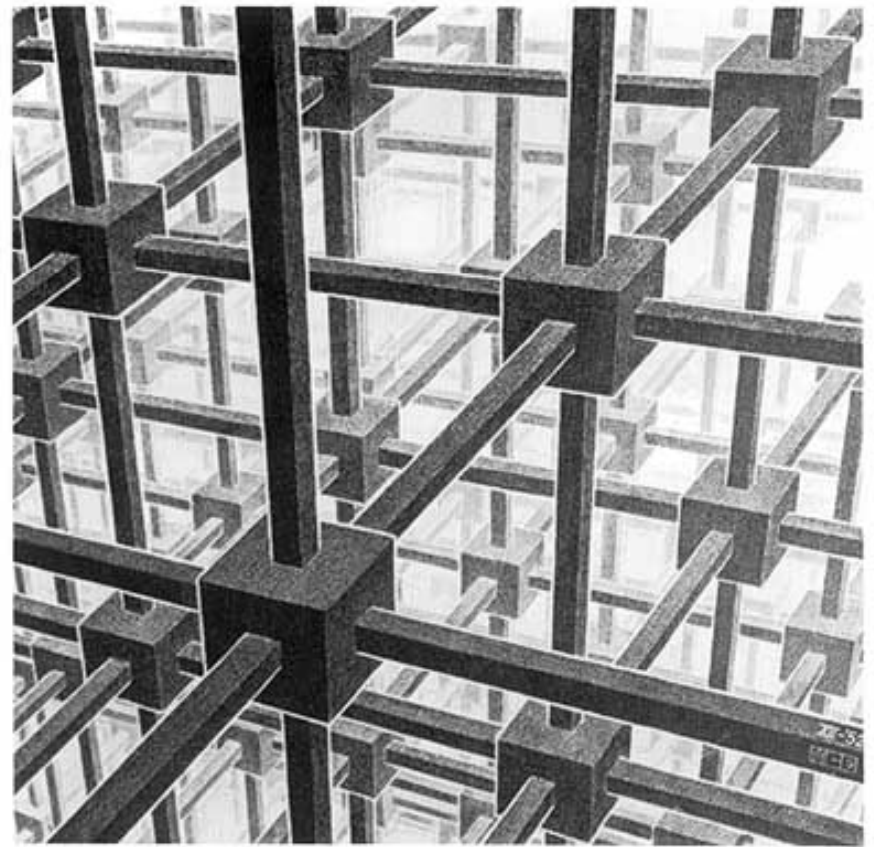
- ▣ Architecture includes structure, but not every structure is an architecture
- ▣ Architecture is more than structure
 - It also involves behavior, design decisions, constraints...

Common Misunderstandings

19

□ Architecture Views

- Architecture is **flat** only in trivial systems, in which one architectural view is sufficient
- Multiple stakeholders have multiple concerns, so that needed multiple architectural views
- A complex system can usually NOT be represented with a single architectural view



Common Misunderstandings

20

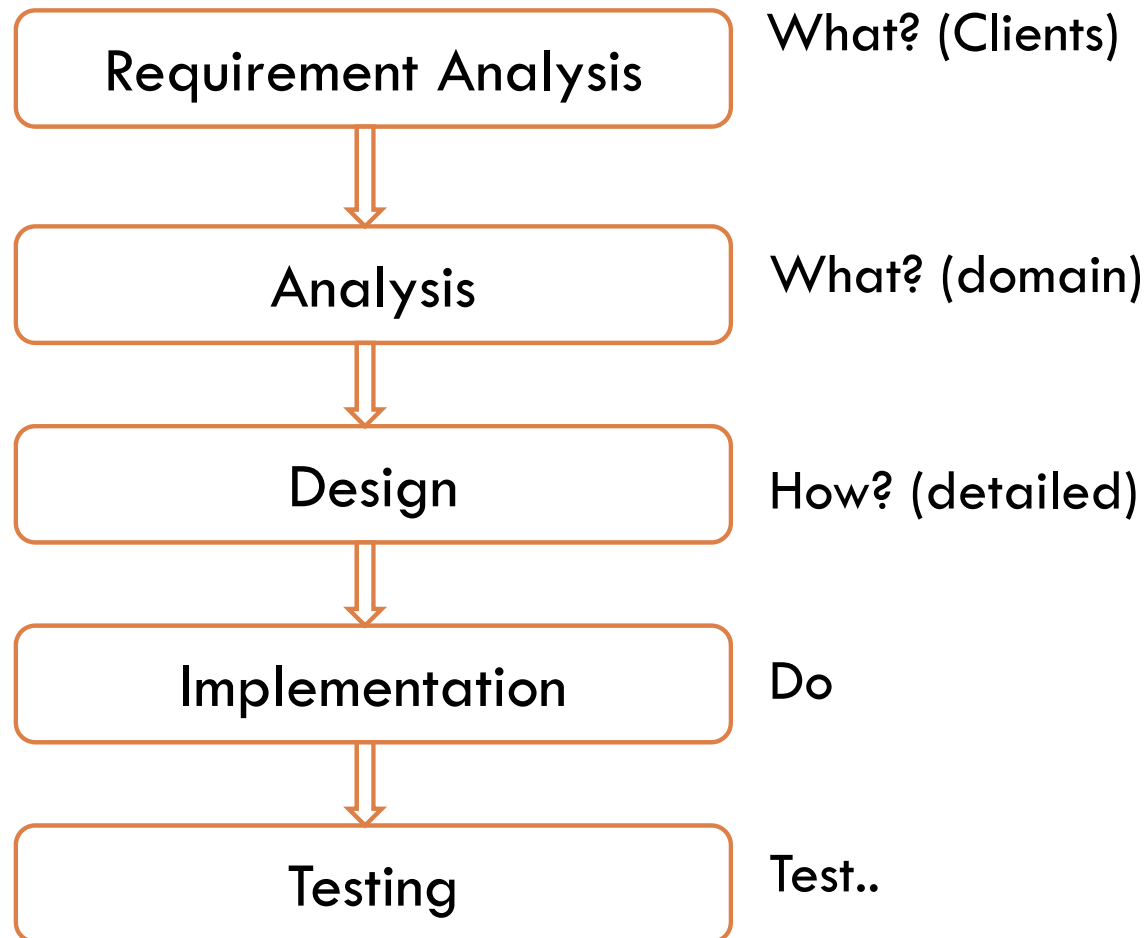
Architecture is:

- ▣ Just documentation
- ▣ Design
- ▣ Infrastructure
- ▣ Structure
- ▣ Technology
- ▣ Is flat

Relation of SWA design w/SWE Phases

21

□ SWE phases (waterfall, iterative, whatever..)

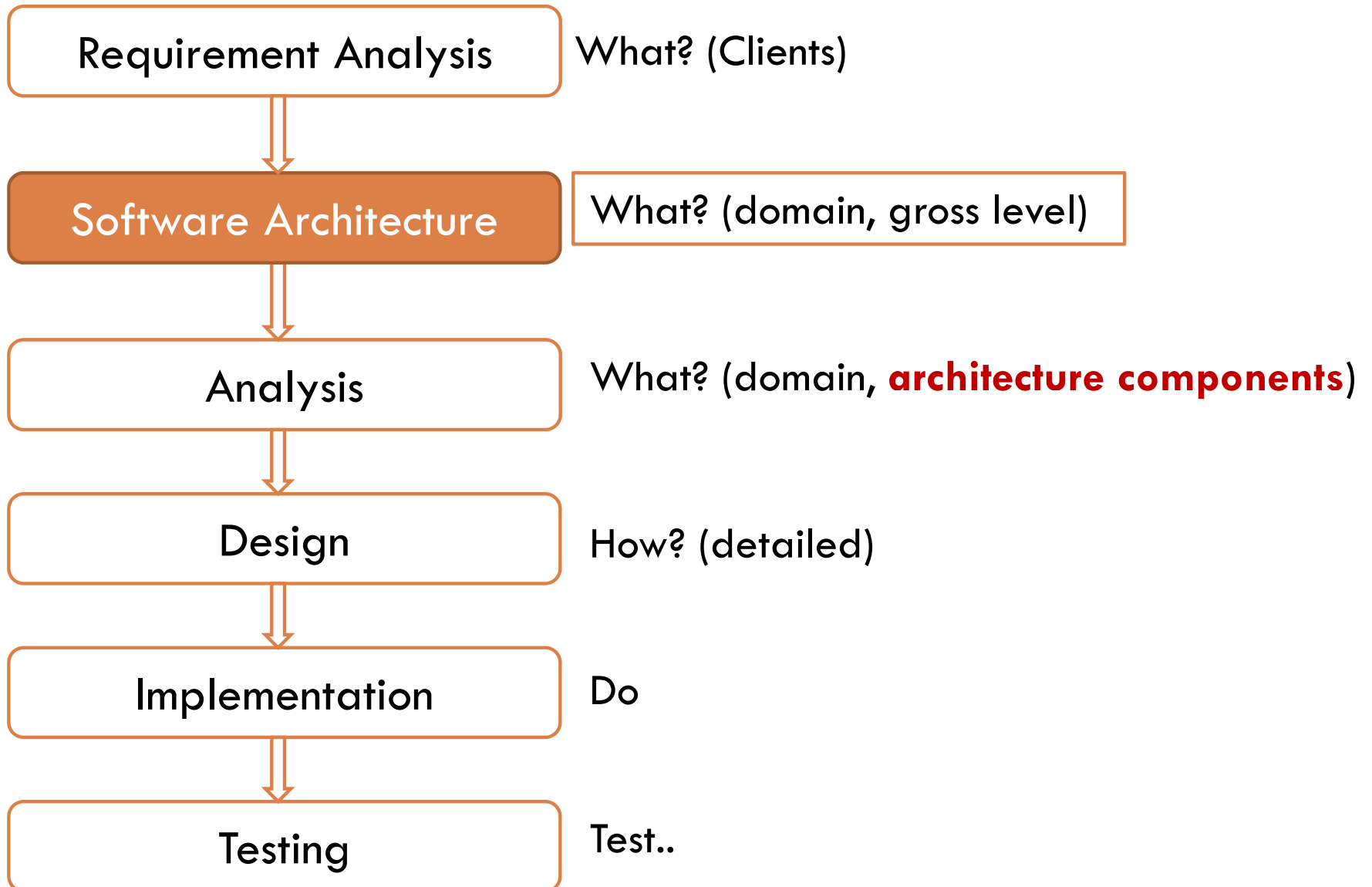


Where is
Architecture
Design ?

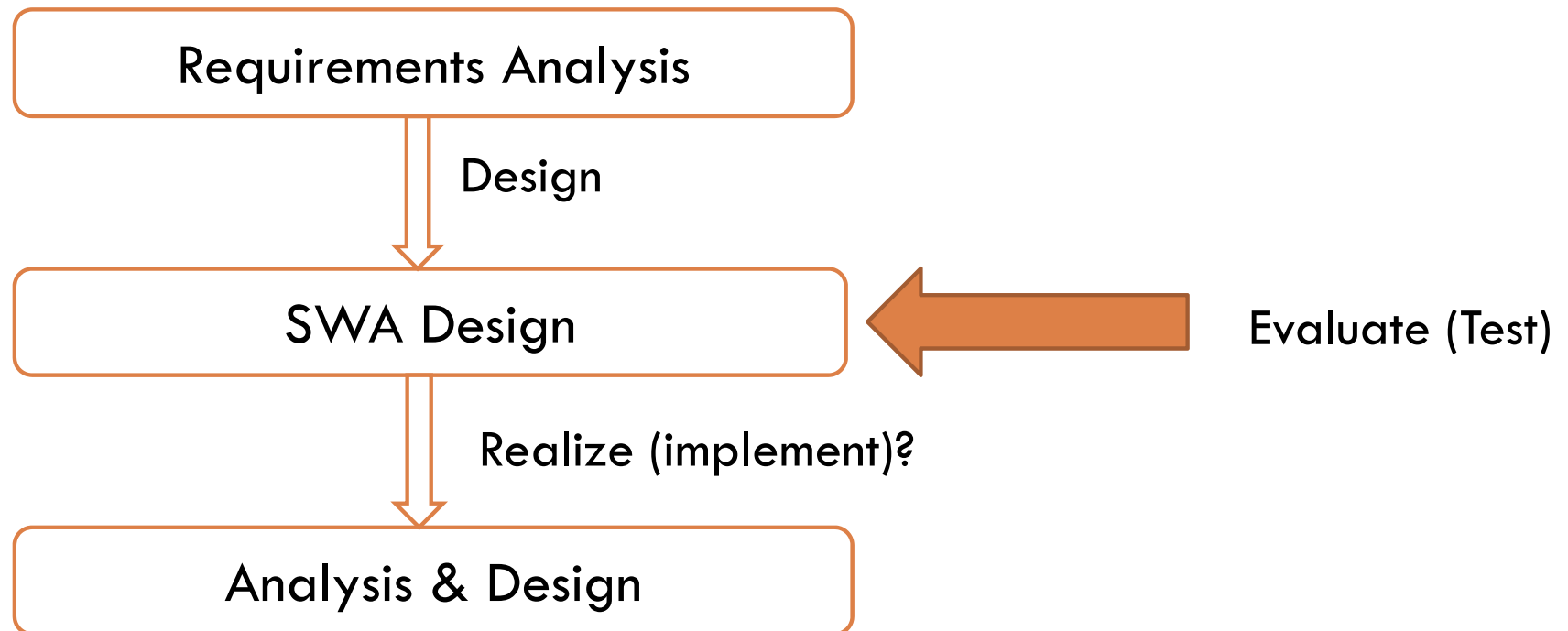


Relation of SWA design w/SWE Phases

22



Design, Evaluate, and Realize SWA



24

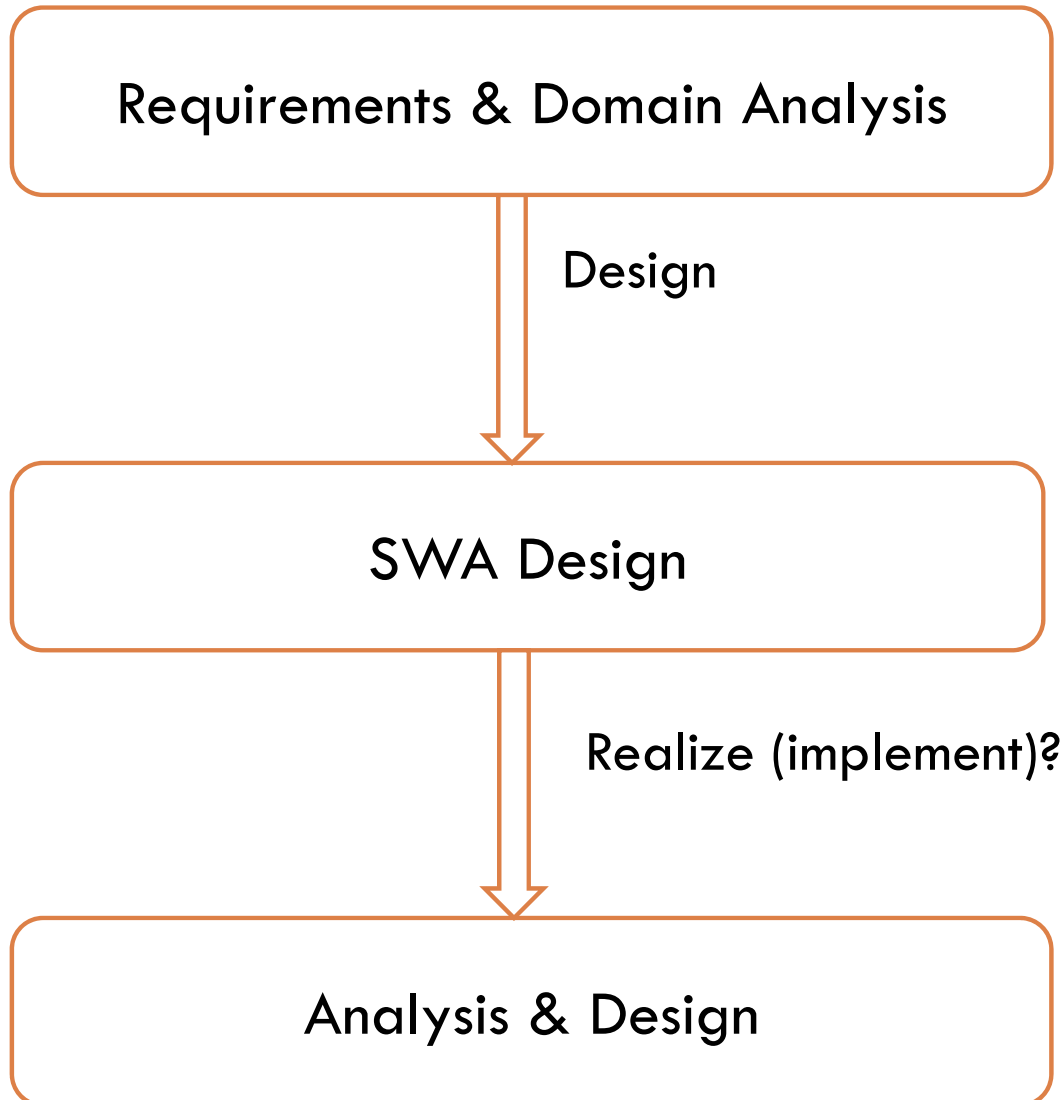
Modeling Architecture

Modeling is Essential!

25

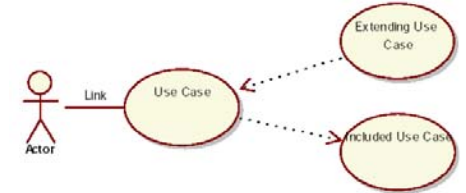
- Visualize any human activity before it is produced (implicit or explicit)
- Provide template for guiding the production
- Documents the decisions that are made
- Communicate our ideas..

Architecture Modeling

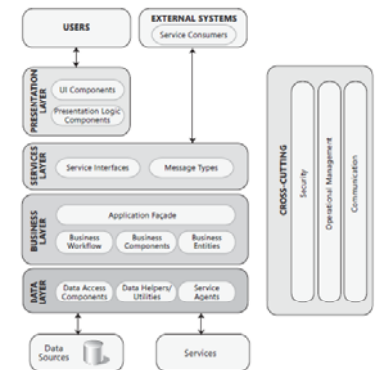


Models (Artifacts)

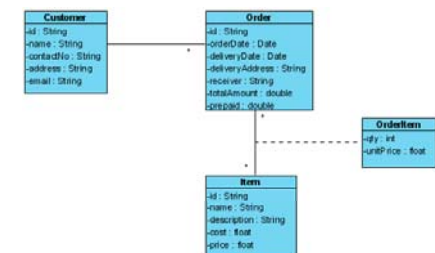
- Use-Case diagram / domain model



- Architecture Model



- Class Diagrams



Architecture Context

27

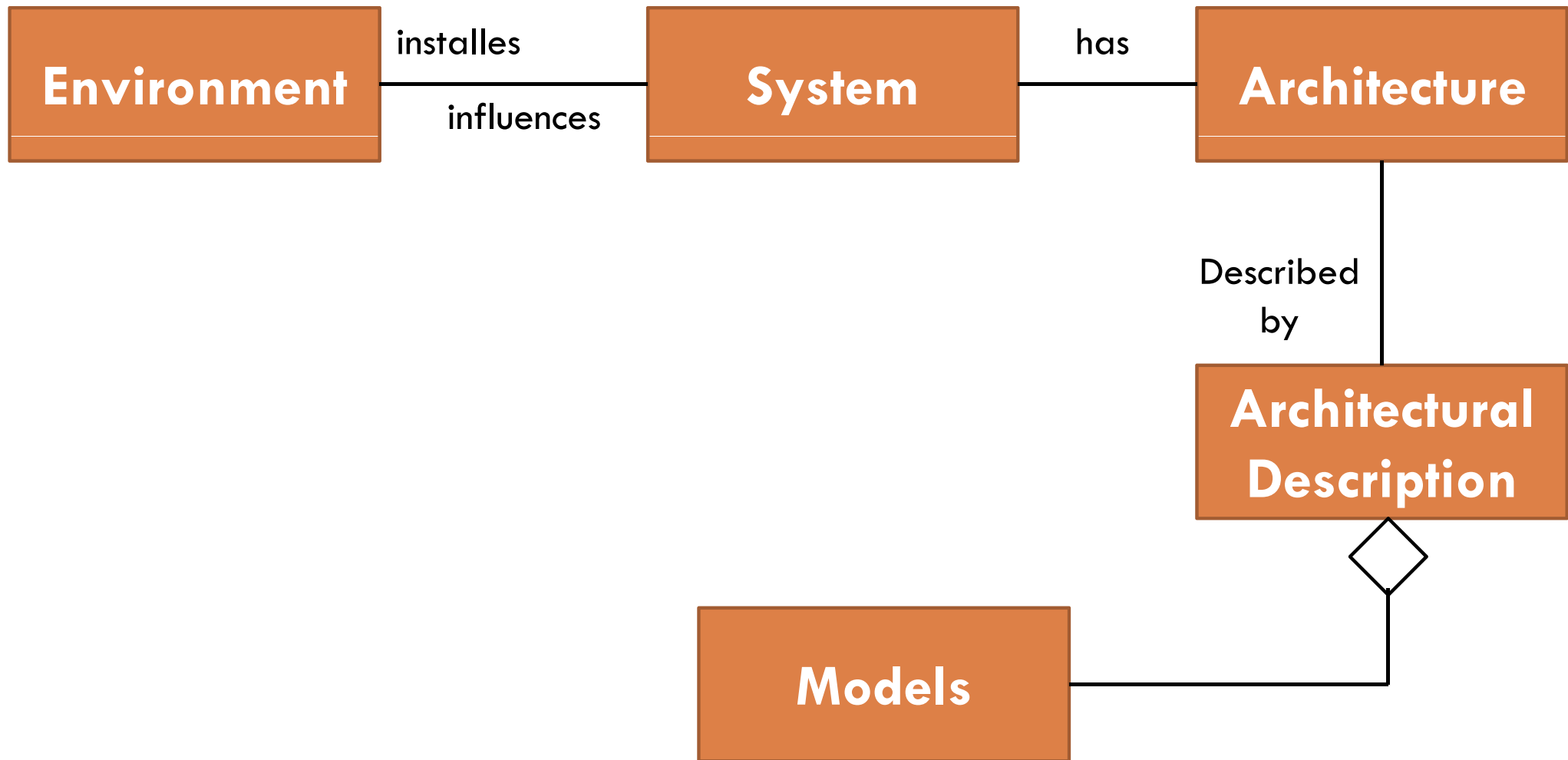
Every system has an architecture!

- Every system is composed of elements and there are relationships among them
- In the simplest case, a system is composed of a single element, related only to itself..



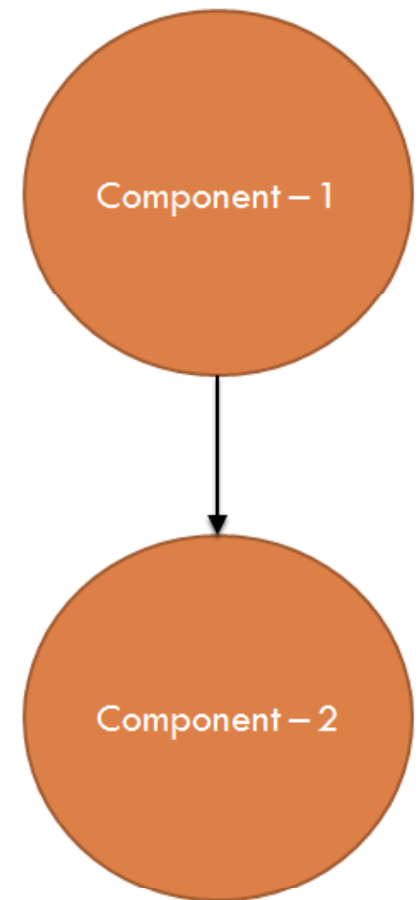
Architecture Description

28



What is wrong with this definition ?

- Ambiguous and often leading to confusion
- What are the components?
 - ▣ Class, Runtime entities (components), Processes, Data stores, ... ?
- What are the relations?
 - ▣ Data flow, Control flow, Dependency link, Allocation, ... ?

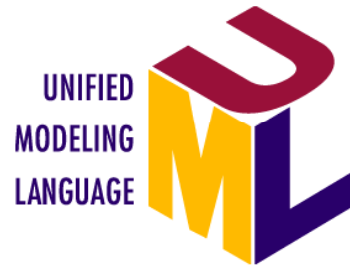


Architecture Modeling

30

□ Visual

- UML
- Others



□ Textual

- Architecture Description Languages (ADLs)
 - The SAE Architecture Analysis and Design Language – AADL
 - Systems Modeling Language – SysML
 - ...

Why UML?

- Is a de facto standard
- General purpose modeling language, supports diverse application areas
- Is based on experience and has a community (OMG)
- Wide-spread use
- Supported by many tools
- Supports the entire software development lifecycle

Ways of Modeling SWA in UML

32

1. Use existing modeling elements “as is”
2. Use UML’s built-in extension mechanisms to refine/extend existing models
3. Introduce new notations to represent architectural elements (change meta-model)

1st Approach

33

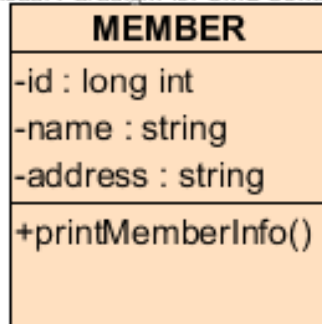
- What are the candidate UML models for representing architectural components?
- What are the candidate UML relations for representing architectural relations?

UML Structural Models

34

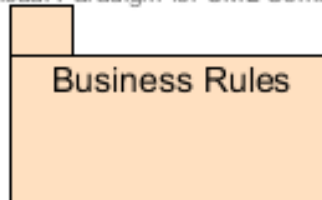
□ Class Models

Visual Paradigm for UML Commun



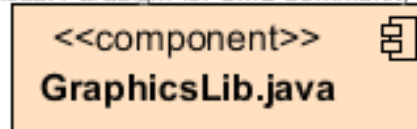
□ Package

Visual Paradigm for UML Commun



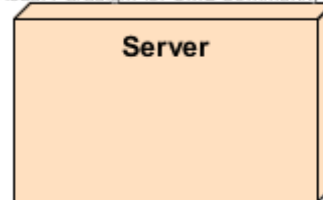
□ Component

Visual Paradigm for UML Community Edit



□ Node

Visual Paradigm for UML Community Edit



UML Relations

35

☐ Aggregation

Visual Paradigm for UML Community Edition [not for commercial use]



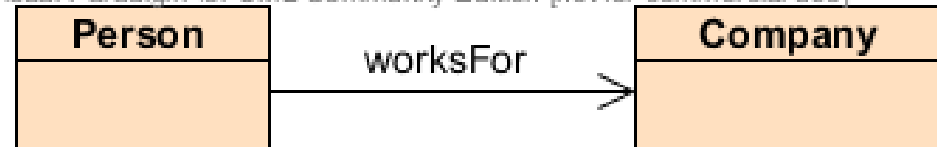
☐ Composition

Visual Paradigm for UML Community Edition [not for commercial use]



☐ Association

Visual Paradigm for UML Community Edition [not for commercial use]



☐ Dependency

Visual Paradigm for UML Community Edition [not for commercial use]



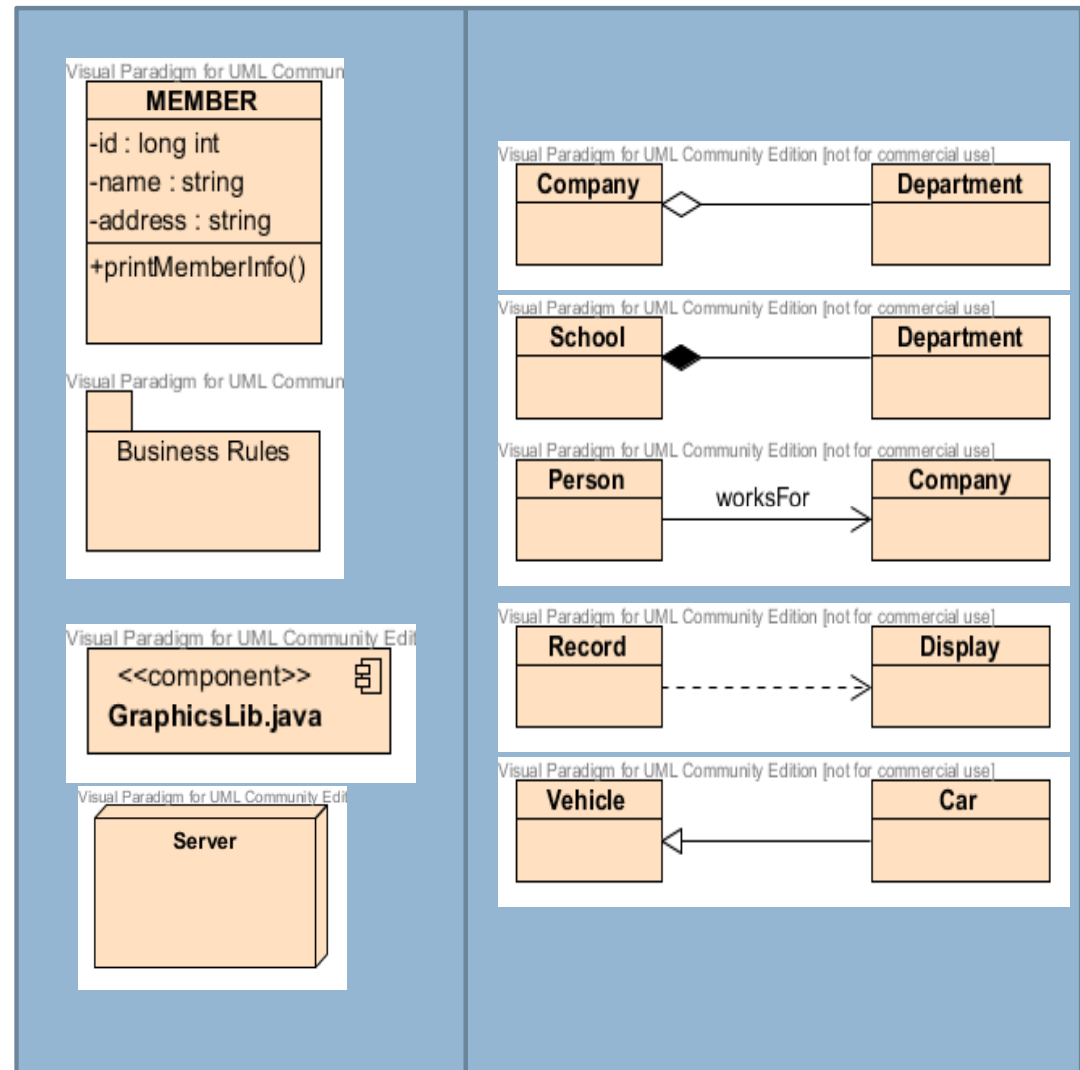
☐ Generalization

Visual Paradigm for UML Community Edition [not for commercial use]



1st Approach: Use UML model “as is”

- UML models for representing architectural components
 - ▣ Class, package, nodes, component
- UML models for representing architectural relations
 - ▣ Aggregation, association, inheritance, dependency

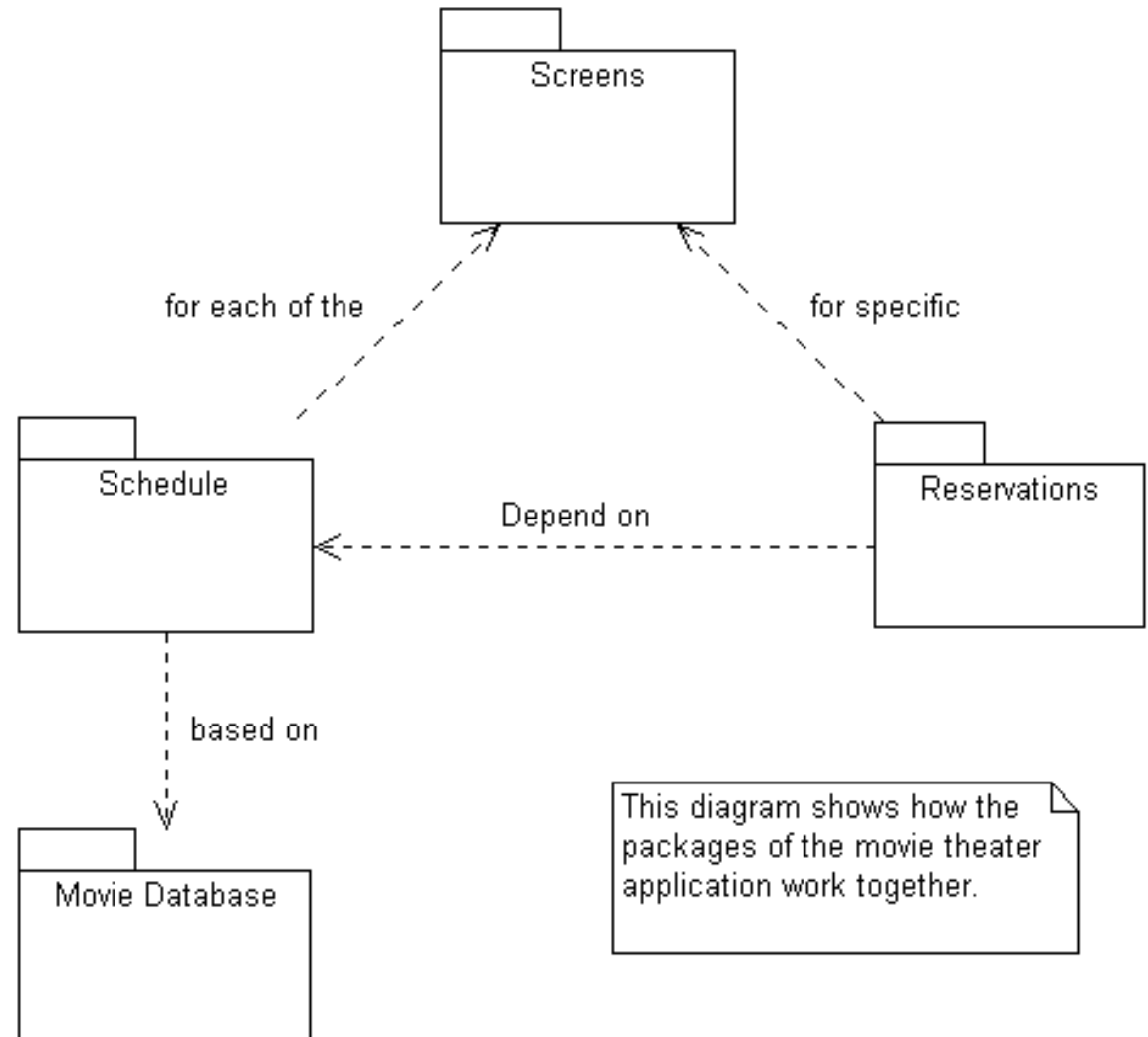


UML

Architectural Components as Packages

□ Example:

□ Video Theater Systems



□ Packages are only grouping constructs

□ Usefull for representing semantics

HW Assignment

38

1. Use existing modeling elements “as is”
2. Use UML’s built-in extension mechanisms to refine/extend existing models
3. Introduce new notations to represent architectural elements (change meta-model)

□ HW Assignment

- Demonstrate the 2nd and 3rd approaches above with an example and comment your assumptions and considerations..

Comparison

39

- **Use UML Model “as is”**
 - ▣ Compatible w/UML compliant tools
 - ▣ Violation of architectural component concept
- **Use built-in extension mechanism**
 - ▣ Compatible w/UML compliant tools
 - ▣ Requires complete style specification
- **Extended meta-model UML**
 - ▣ Provides “native” support for architectures
 - ▣ Requires backward tool compatibility
 - ▣ May result in incompatible UML versions

