Domain Modeling & Mapping Architectures to Design

Lecture 05

BIL428 Software Architectures

Asst.Prof.Dr. Mustafa Sert
msert@baskent.edu.tr

Department of Computer Engineering, Başkent University
Ankara 06810 TURKEY

Contents

- Domain Modeling
  - Domain Modeling Techniques
  - Feature Driven Modeling
- Mapping Architectures to Design
  - Application Frameworks

Domain Model

- **Explicit representation**
  - Of **common** and **variable** properties of the system in the domain

Domain Modeling Techniques

- **Domain Definition**
  - Describes the scope of the domain
  - What is in? What is out?
  - Examples
- **Domain Lexicon/Glossary**
  - Description of the basic, but important terms in the domain
- **Conceptual Models**
  - Describe concepts represented as OO diagrams, ER diagrams, or any other suitable modeling technique
- **Feature Models**
  - Describes common and variant properties of concepts

Example – Domain Definition

Driver Monitoring Systems

…

A driver monitoring system is a control feedback system in which the driver and the car performance is monitored..

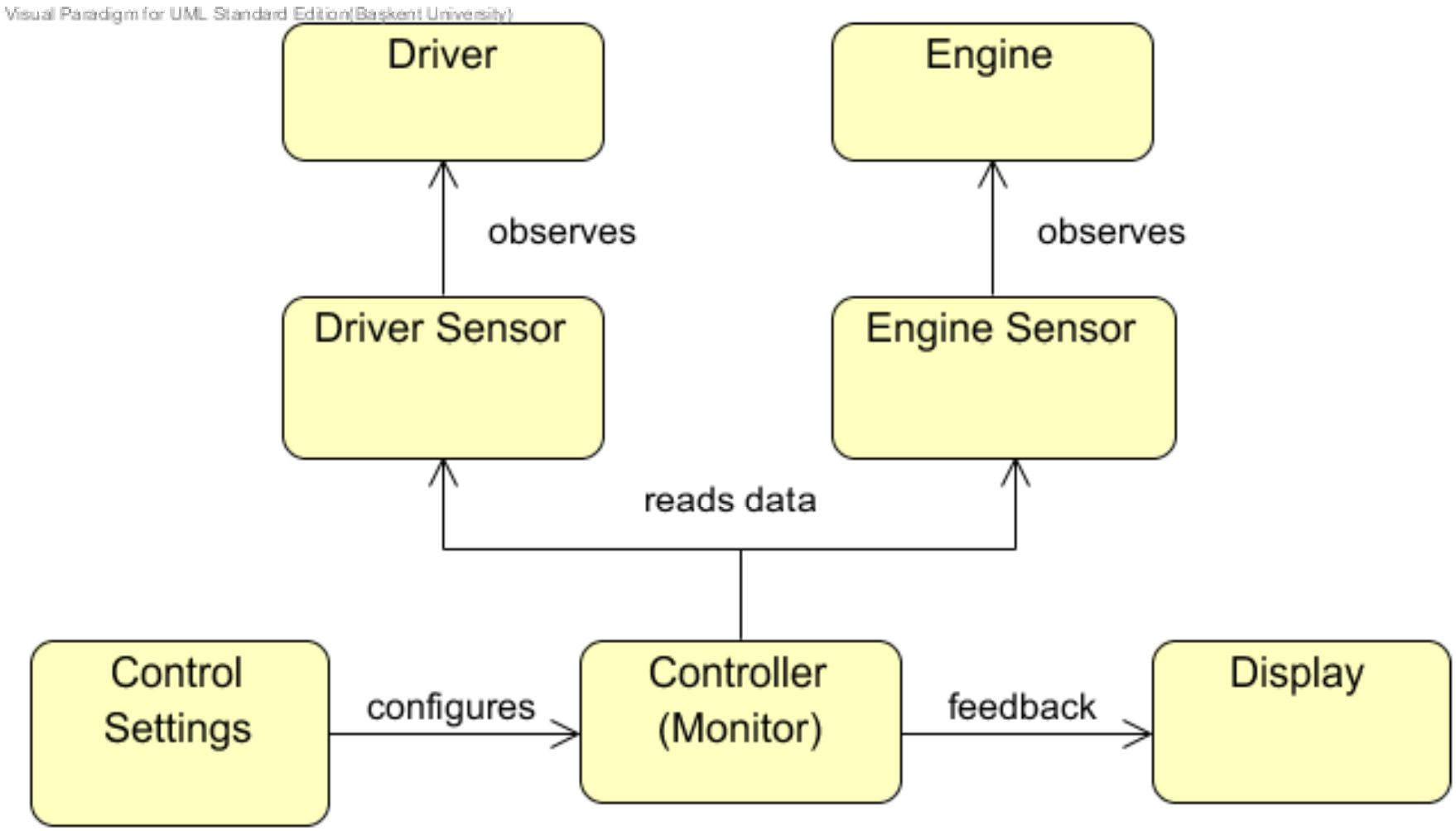
Examples:

…

Example – Domain Glossary

- **Monitor**
  - The entity that monitors the driver and the engine performance
- **Sensor**
  - Entity that observes the controlled entity
- **Control Data**
  - Data which represent the goal parameters
- **Feedback**
  - Reaction given to the driver by the monitor
- **Display**
  - Physical entity to represent the feedback of the monitor
- …

Example – Conceptual Model

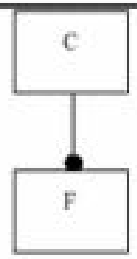
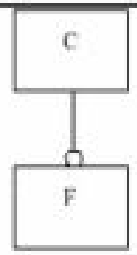
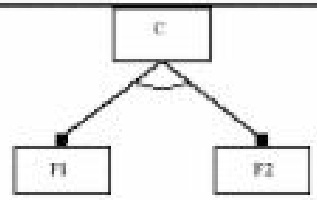
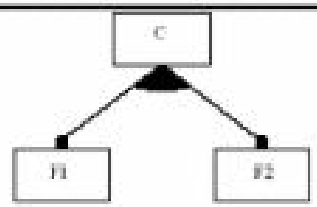


Feature-Oriented Domain Modeling

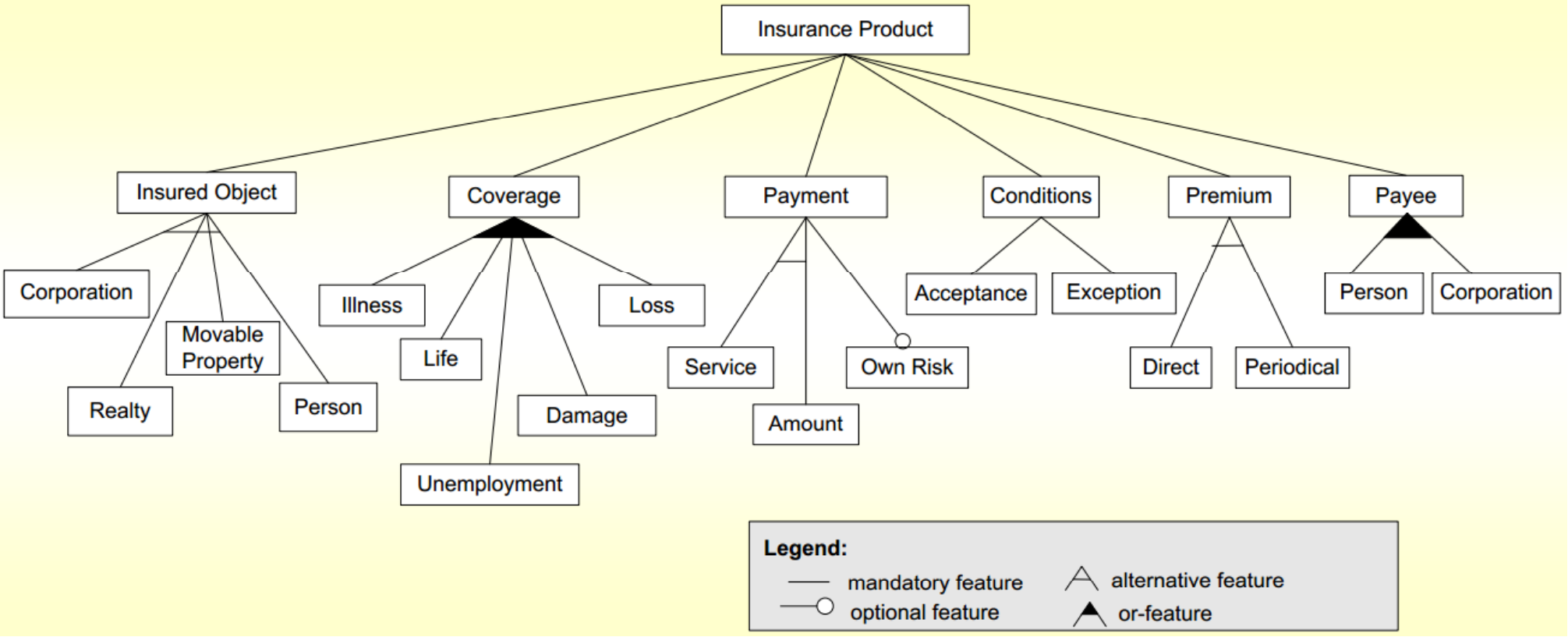
- A **feature model** represents the common and the variable features of the products and the dependencies btw the variable features
- **Feature**
  - A distinctive property of the concept (domain model)
  - User-visible characteristics of a system (requirements)
- A **feature diagram** consists of a set of nodes, a set of directed edges, and a set of edge decorations

Basic Feature Types

- **Mandatory Features**
  - Each application must have
- **Optional Features**
  - Each application can have or NOT
- **Alternative Features**
  - Each application must have one of (XOR)
- **OR Features**
  - Each application must have one of or multiple

Type	Notation
Mandatory	
Optional	
Alternative	
Or	

Example – Feature Model

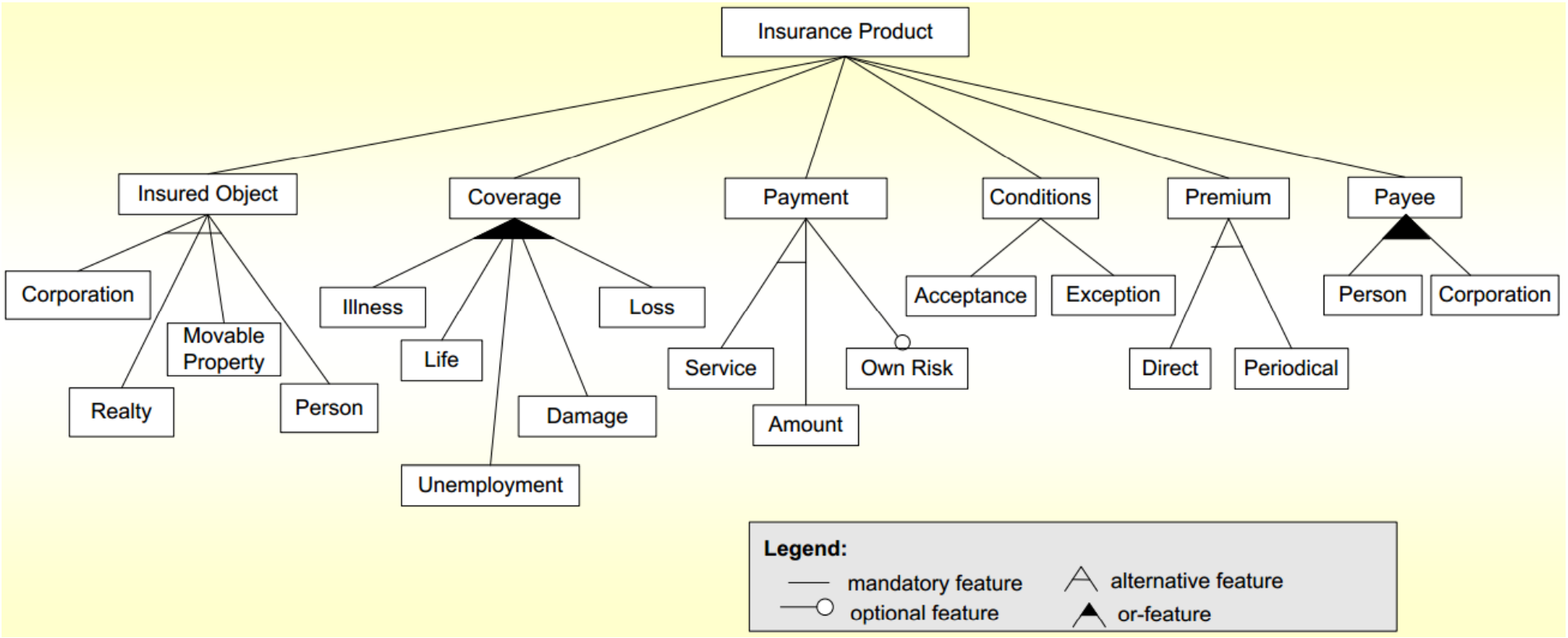


Composition Constraints

- Two types of compositions
  - **Mutex-with** rule
    - Defines a mutual exclusion relation btw two concepts or features
  - **Requires** rule
    - Defines which features the selected feature requires (interdependent relations)

Exercise – Sample Domain

- Given sample domain, show an example **mutex** and **requires** constraint



Sample Constraints from the Exercise

- InsuredObject.Person **mutex-with** Coverage.Damage
  - If the insured object is a person, then the insurance product cannot include coverage of damage (for physical objects)
- Coverage.Loss **requires** InsuredObject.MoveableProperty
  - If the insurance product includes coverage for loss, then the insured object can only be a moveable property
- Coverage.Illness **mutex-with** InsuredObject.Corporation
  - If the insurance product includes coverage for illness, then the insured object cannot be a corporation
- InsuredObject.MoveableProperty **requires** Coverage.Damage
  - If you select MoveableProperty feature, then you should select Damage

Possible Insurance Systems from the Exercise..

- Alternatives from the exercise
  - Life insurance with service and periodical payment
  - Car insurance with coverage with damage, own risk and periodical payment
  - Health insurance that covers illness with own risks and direct premium
  - …
- How many insurance systems can you derive from the given model?
  - ~ 3000 alternatives !

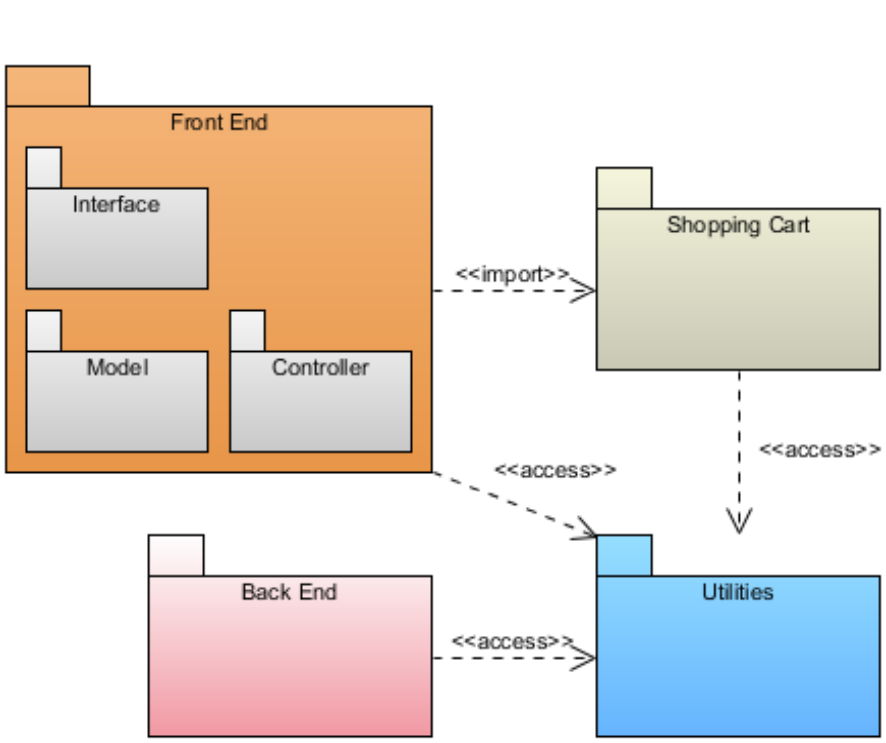
Exercise 2

- Define a feature diagram for the driver monitoring system including the following features
  - A display can be either red/green or multifunctional
  - Control data can be entered directly by the user or downloaded remote from a network
  - Driver performance must be monitored based on the physiological characteristics and the driving behavior
  - The physiological characteristics include eye movements, head movements, and optionally heart rate
  - Driving behavior includes tracking steering movements, brake maneuvers, and transmission maneuvers

Application Frameworks

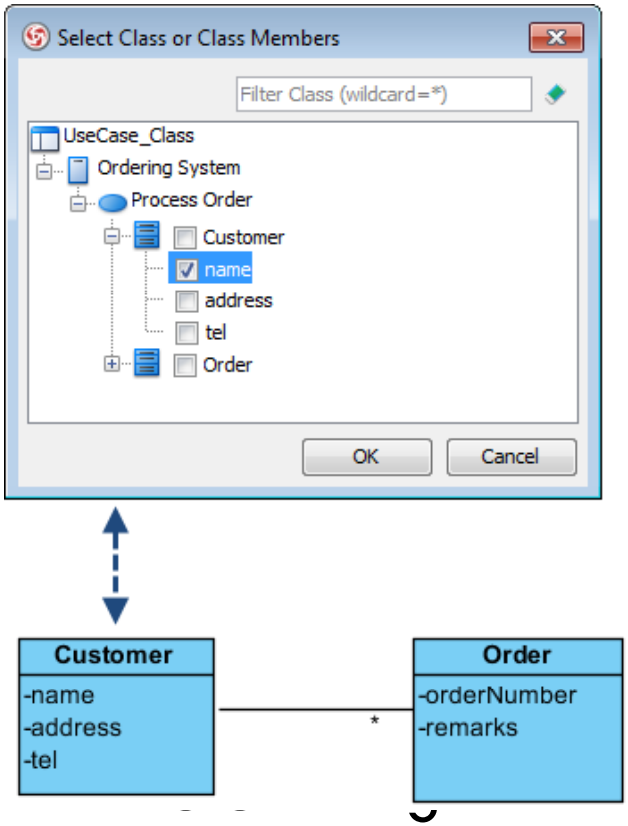
Map Architecture to Implementation

- Map Architecture to one complete implementation



ARCHITECTURE

MAP TO



(Implementation View)

Application Framework

- **DEFINITION**
  - A **reusable, "semi-complete"** application that can be **specialized** to produce **custom applications**
  - A set of classes that embodies an abstract design for solutions to a family of related problem

Users and Developers of Frameworks

Main roles associated with frameworks

- **Framework Developers**
  - Develop the original framework (based-on domain driven architecture)
- **Framework Users**
  - Also called framework clients or application developers
  - Use the framework to develop applications
  - Reuse and/or extend the framework for customized applications
- **Framework Maintainers**
  - Refine and develop the framework to meet new requirements

Key Points - Reuse

- Reuse methods – discussion..
  - OO-PL Mechanisms
  - Class Libraries
  - Software Components
  - Design Patterns
  - Application Frameworks

Classifying Application Frameworks

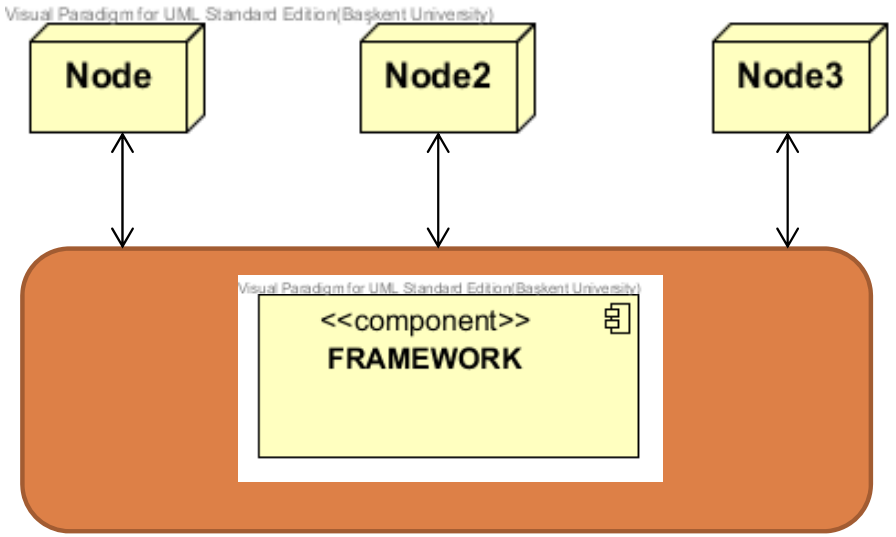
- System Infrastructure frameworks
- Middleware integration frameworks
- Enterprise application frameworks

System Infrastructure Framework

- Simplify portable and efficient system infrastructure development such as operating systems
- Are primarily used internally within a software organization and are not sold to customers directly

Middleware Application Frameworks

- Commonly used to integrate distributed applications and components
- Designed to enhance the ability of software developers to modularize, reuse, and extend their SW infrastructure to work seamlessly in a distributed environment
- Examples: CORBA, Message Oriented Middleware



Enterprise Application Frameworks

- Address broad application domains and are the centre of enterprise business activities
- Relatively expensive to develop and/or purchase
- However, enterprise frameworks can provide a substantial return on investment since they support the development of end-user applications and products directly