



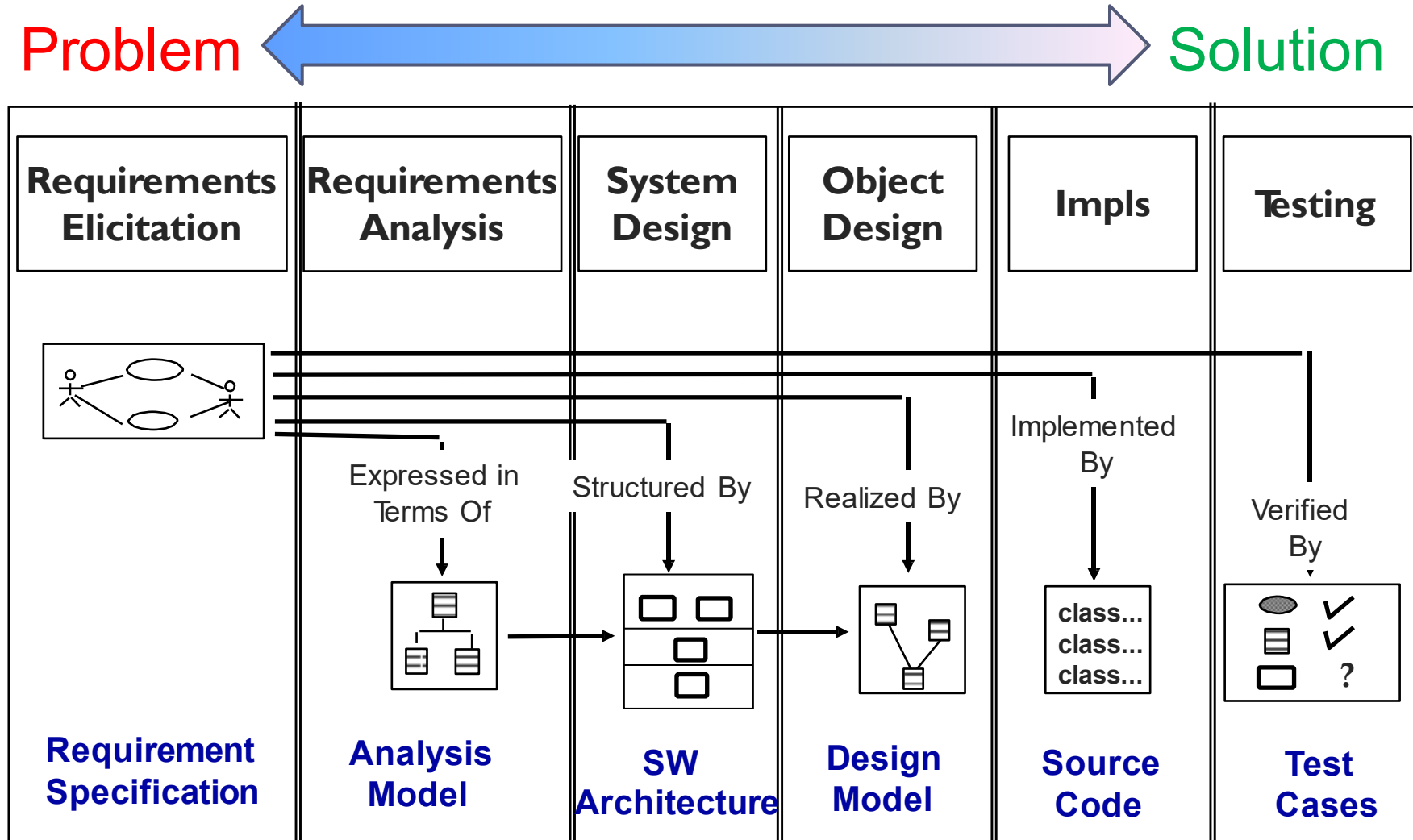
**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

Tutorial #6 – Software Architecture and Strategy Pattern

Dr. Wang Wenhan



Software Development Life Cycle (SDLC) Activities



Software Architecture

Software architecture is considered as a description of the high level structure of a software system in terms of architectural elements and the interactions between them.

Software Architecture = {Components, Connectors}

Components

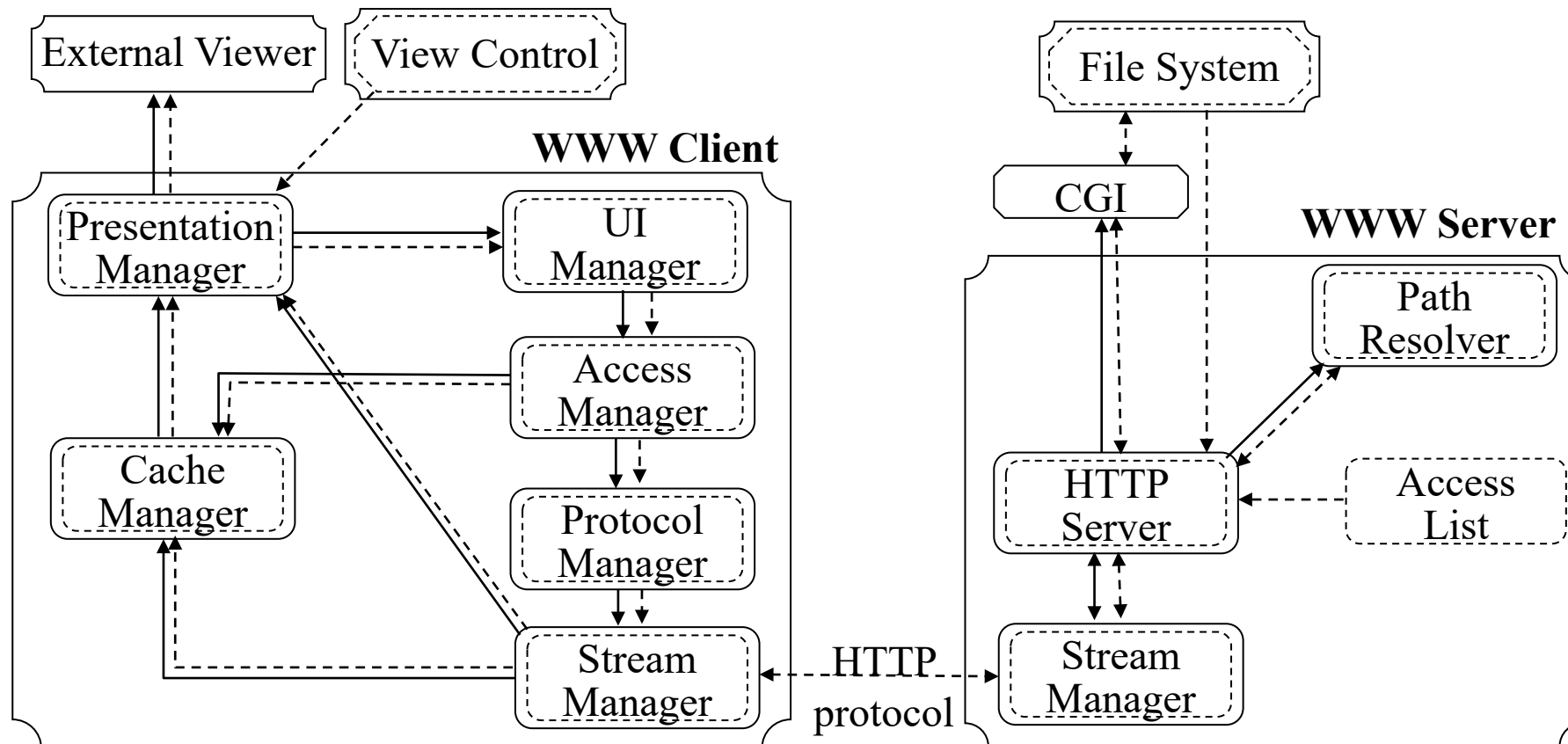
A component is a unit of software that performs some function at run-time.

Connectors

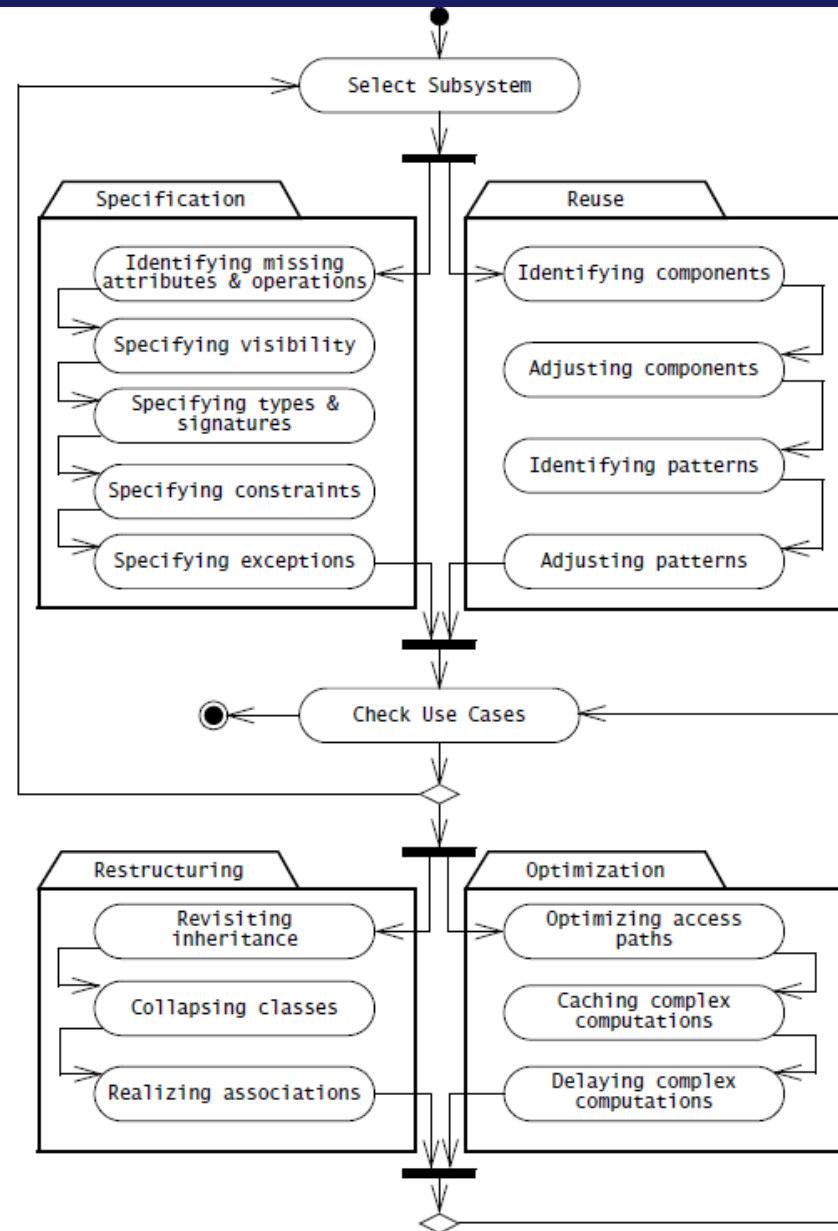
Interactions among components:

Architecture Diagram: an Example

WWW Client-Server Architecture



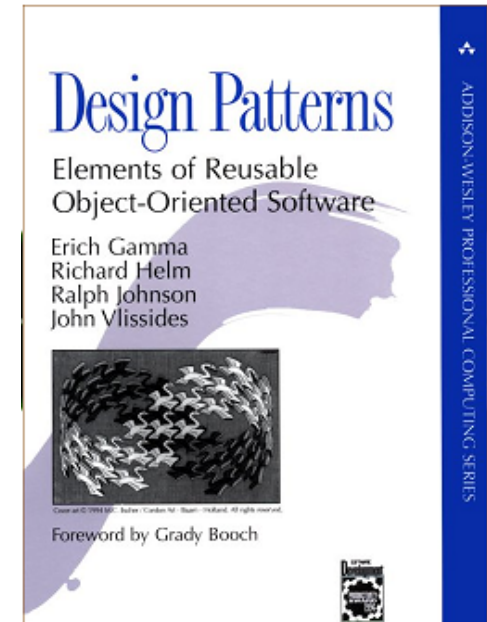
Object Design



What is a Design Pattern?

A design pattern

- A proven solution to a problem in a context.
- Abstracts a recurring **design structure**
- A template with class and/or object
 - dependencies
 - structures
 - interactions
 - conventions



“Gang of Four” (GoF) Book, 1996

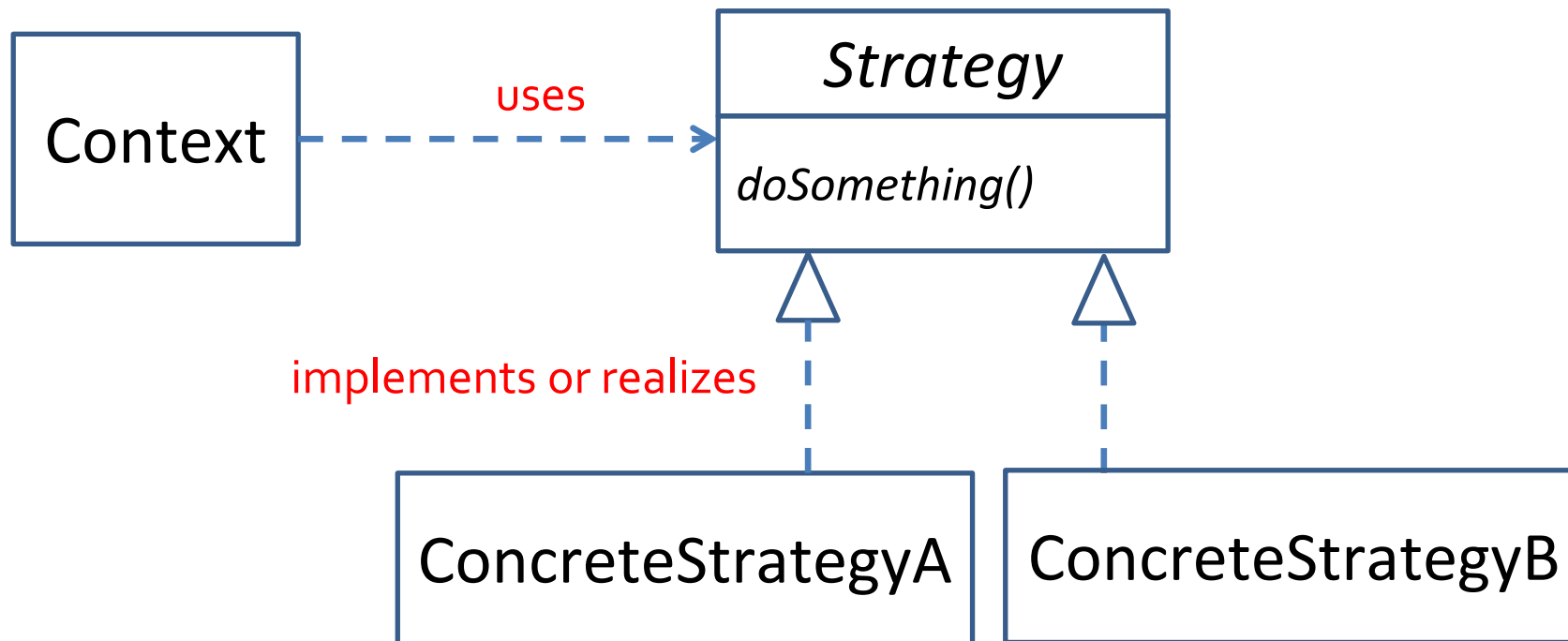
Elements of a Design Pattern

- Name
 - Describes the pattern
 - Adds to common terminology for facilitating communication (i.e. not just sentence enhancers)
- Problem
 - Describes when to apply the pattern
 - Answers - What is the pattern trying to solve?
- Solution
 - Describes elements, relationships, responsibilities, and collaborations which make up the design
- Consequences
 - Results of applying the pattern
 - Benefits and Costs
 - Subjective depending on concrete scenarios

Strategy Pattern

Design problem: A set of algorithms or objects should be **interchangeable**.

Solution: Strategy Pattern



Strategy Pattern

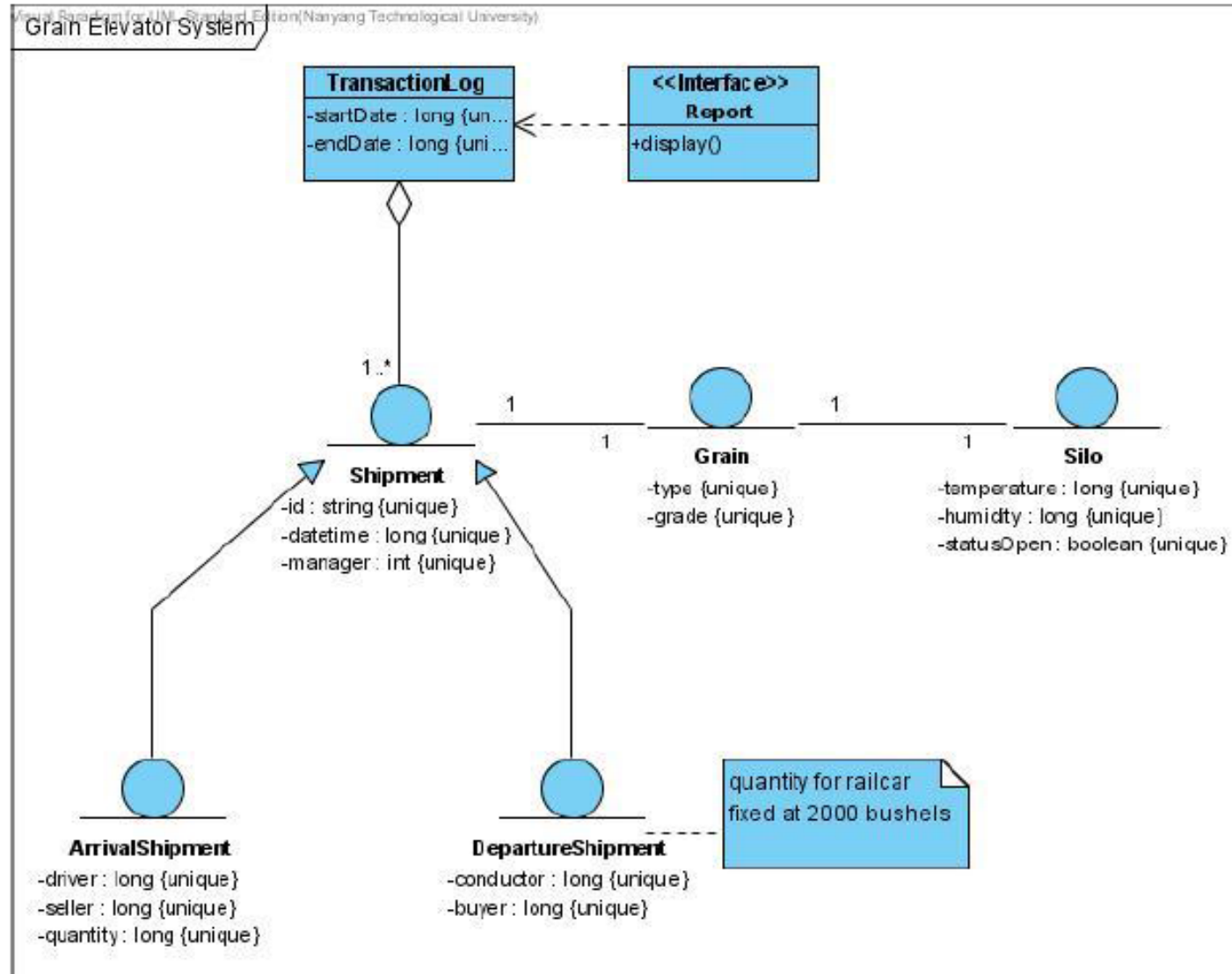
- Pros
 - Easy extension for new strategies
 - Provides encapsulation, hides implementation
 - Allows behavior change at runtime
- Cons
 - Results in complex, hard to understand code if overused

Question 1

Grain Elevator System (GES)

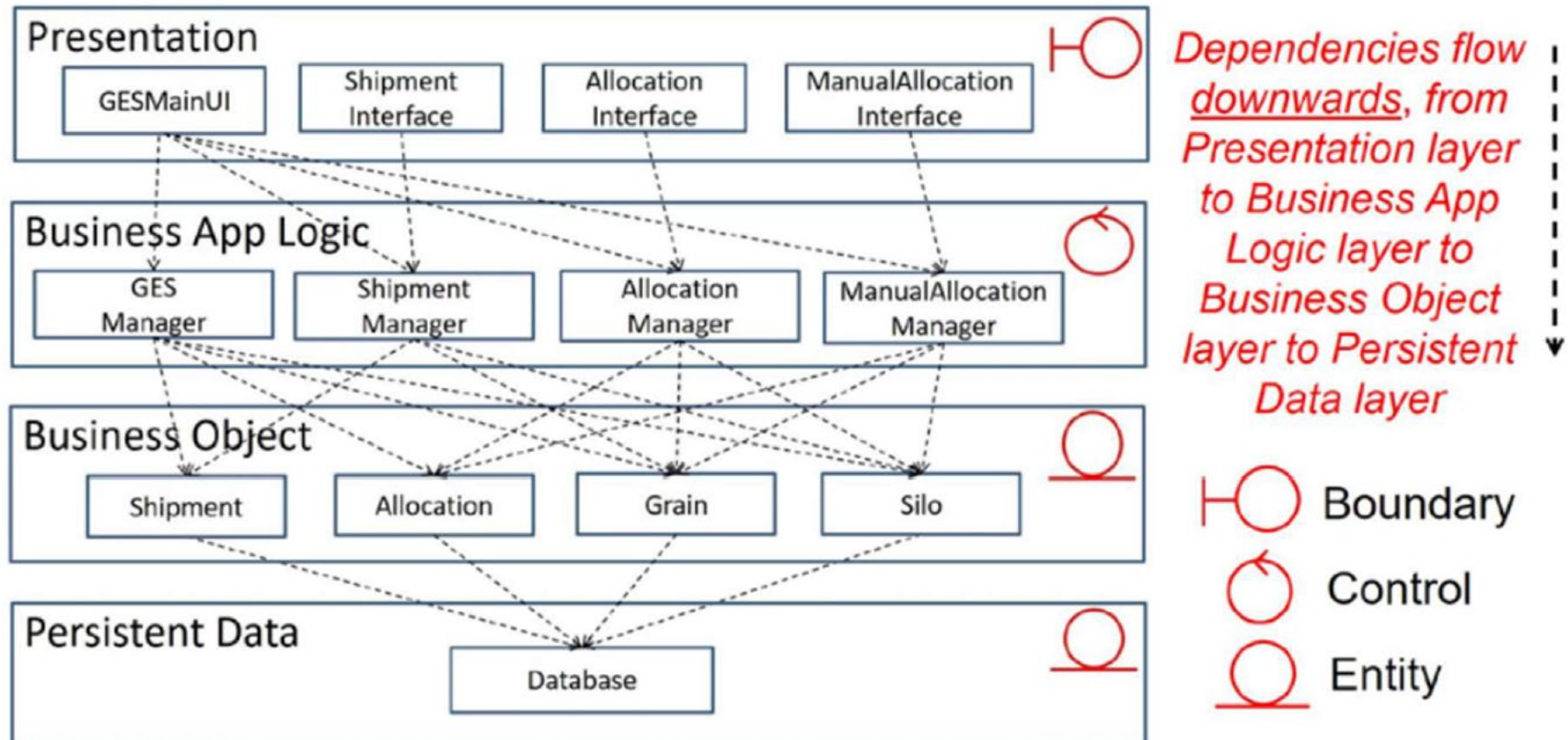
- Refer to the conceptual model developed in Tutorial#2.
 - Use proper architectural styles to model dependencies among boundary, control, and entity objects.

Question 1 - Answer



Question 1 - Answer

Layered architecture (4 layers)



Question 2

Grain Elevator System (GES)

The elevator manager uses a mobile device to process shipment and allocation. This mobile device must deal with a variety of network access protocols (LAN, WiFi, Bluetooth, 3G, 4G). Furthermore, we want to be able to deal with further network protocols with minimal impacts on the application.

- a) Identify the design problem and apply appropriate design pattern to address this problem
- b) Draw a UML class diagram depicting the classes in the design pattern and explain their roles

Question 2.a - Answer

Grain Elevator System (GES)

- a) Identify the design problem and apply appropriate design pattern to address this problem

Design problem:

- Different network protocols should be interchangeable and transparent from the other parts of the mobile device system. Which protocol to use should be easily interchangeable.
- Extensibility: future network protocols should be able to be added with minimal impacts to the application .

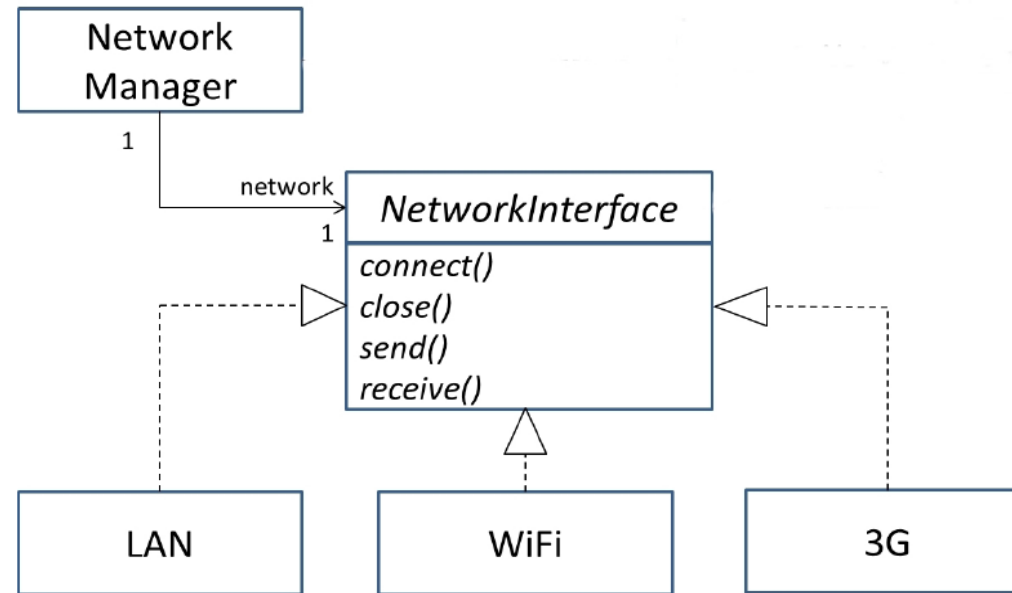
Solution: Strategy pattern

In computer programming, the strategy pattern (also known as the policy pattern) is a behavioral software design pattern that enables selecting an algorithm at runtime. Instead of implementing a single algorithm directly, code receives run-time instructions as to which in a family of algorithms to use.

Strategy lets the algorithm vary independently from clients that use it.

Question 2.b - Answer

Class diagram of the strategy pattern



NetworkManager: Context
NetworkInterface: Strategy interface
LAN, WiFi, 3G, etc.: Concrete strategy object