NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Tutorial #7 – Observer Pattern and Factory Pattern

Dr. Wang Wenhan

Observer Pattern – Design Problems

Defines a one-to-many dependency between objects so that when one object changes state, all of its dependents are notified and updated automatically.

Easy Subscription and Automatic Notification

Observer Pattern

How to Resolve these Problems?

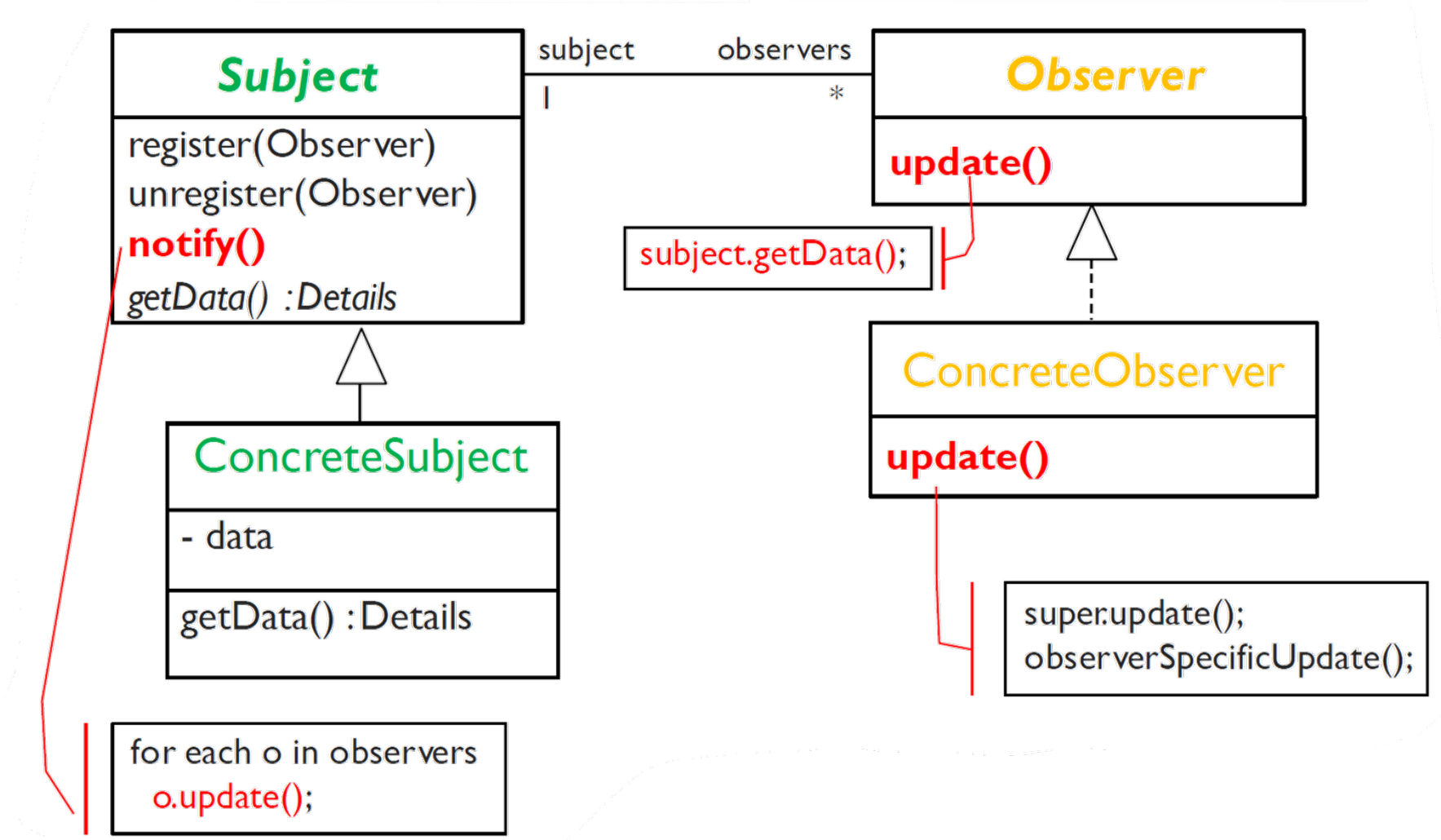
- **Subscription** mechanism

  Observers freely register/unregister their interests in Subject

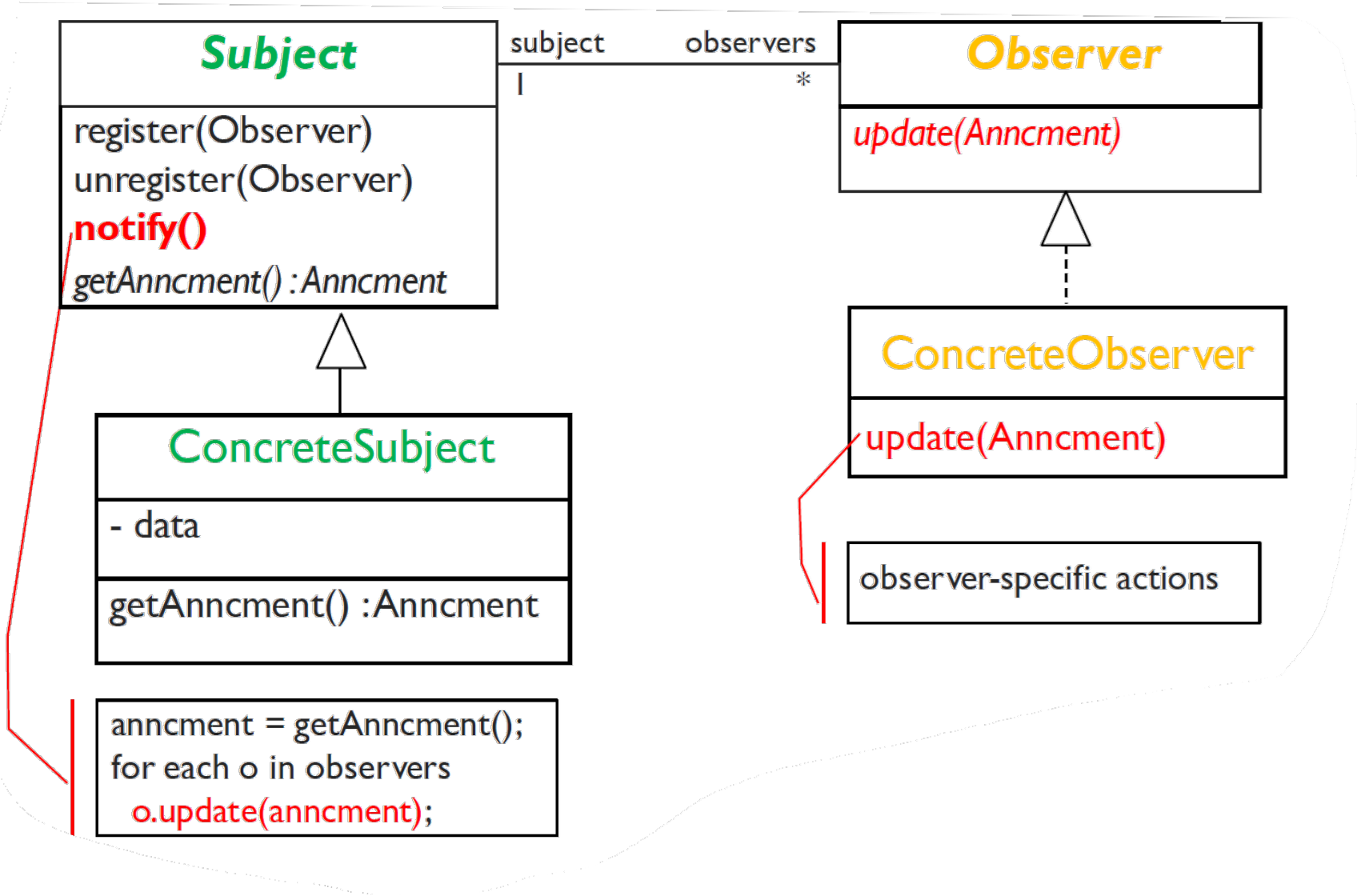
- **Notification** mechanism

  Subject propagates the change to Observer when the change occurs

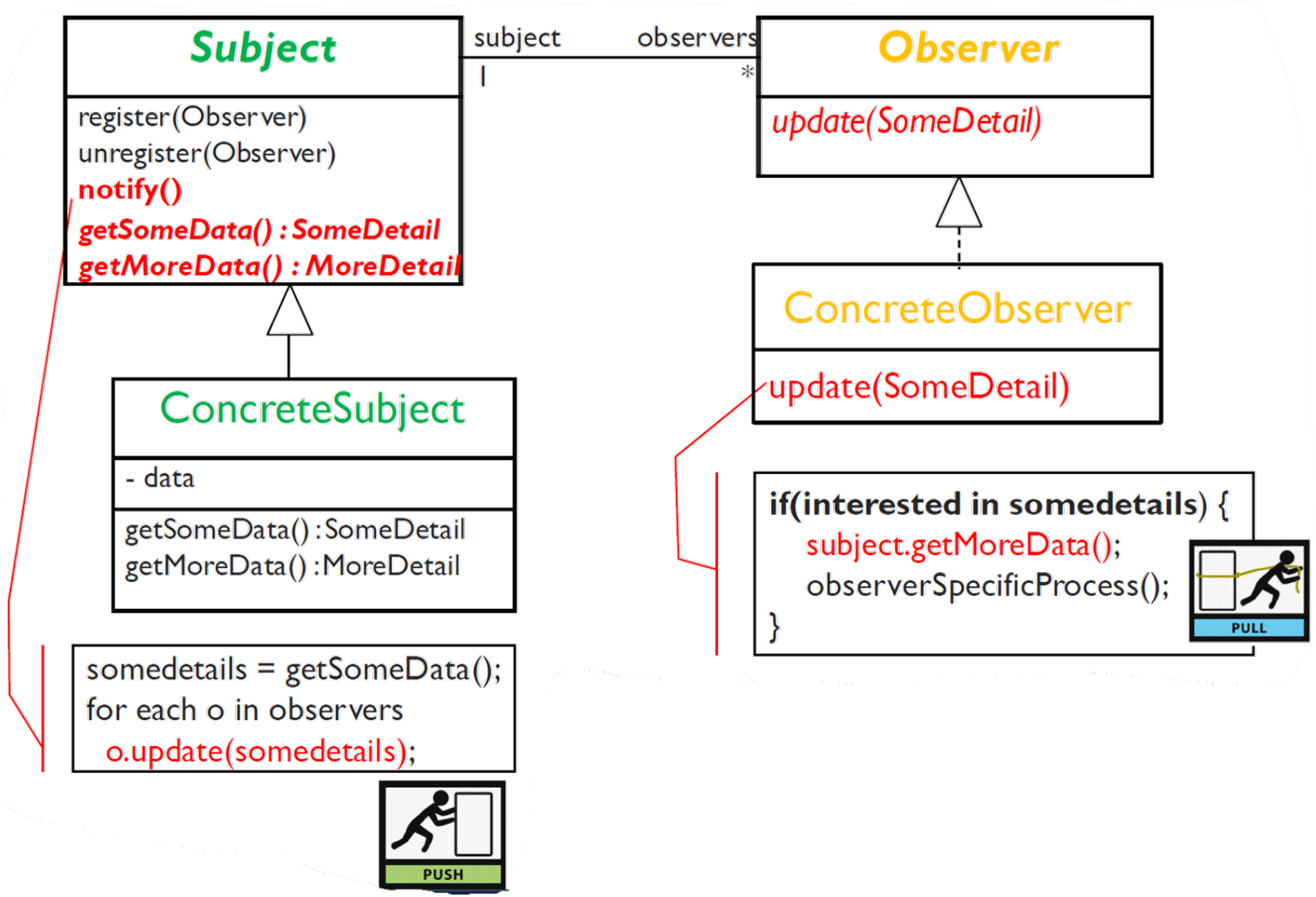
Observer Pattern (Pull)



Observer Pattern (Push)



Observer Pattern (Push+Pull Update)



Observer Pattern

- Pros
  - Abstracts coupling between Subject and Observer
  - Supports broadcast communication
  - Enables reusability of subjects and observers independently of each other

***Loose coupling** is a benefit for both sides!*

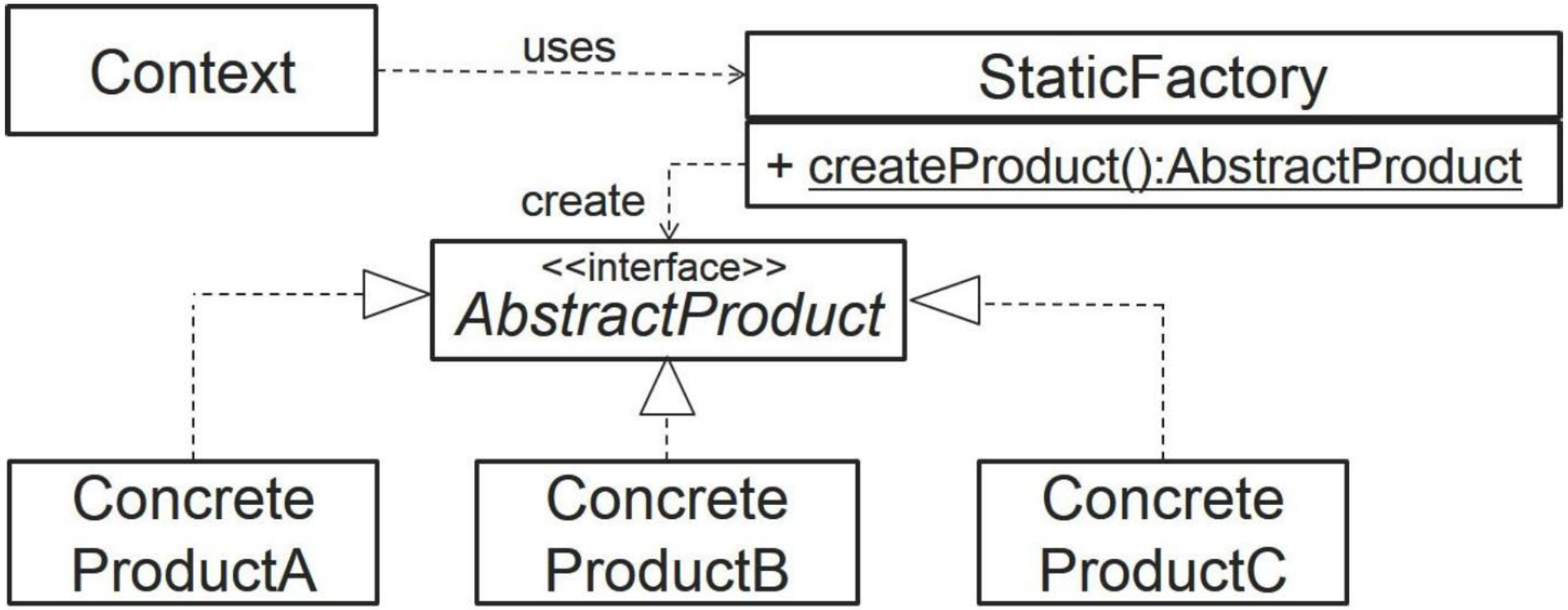
- Cons
  - Slower performance
  - If not used carefully the observer pattern can add unnecessary complexity

Factory Pattern – Design Problems

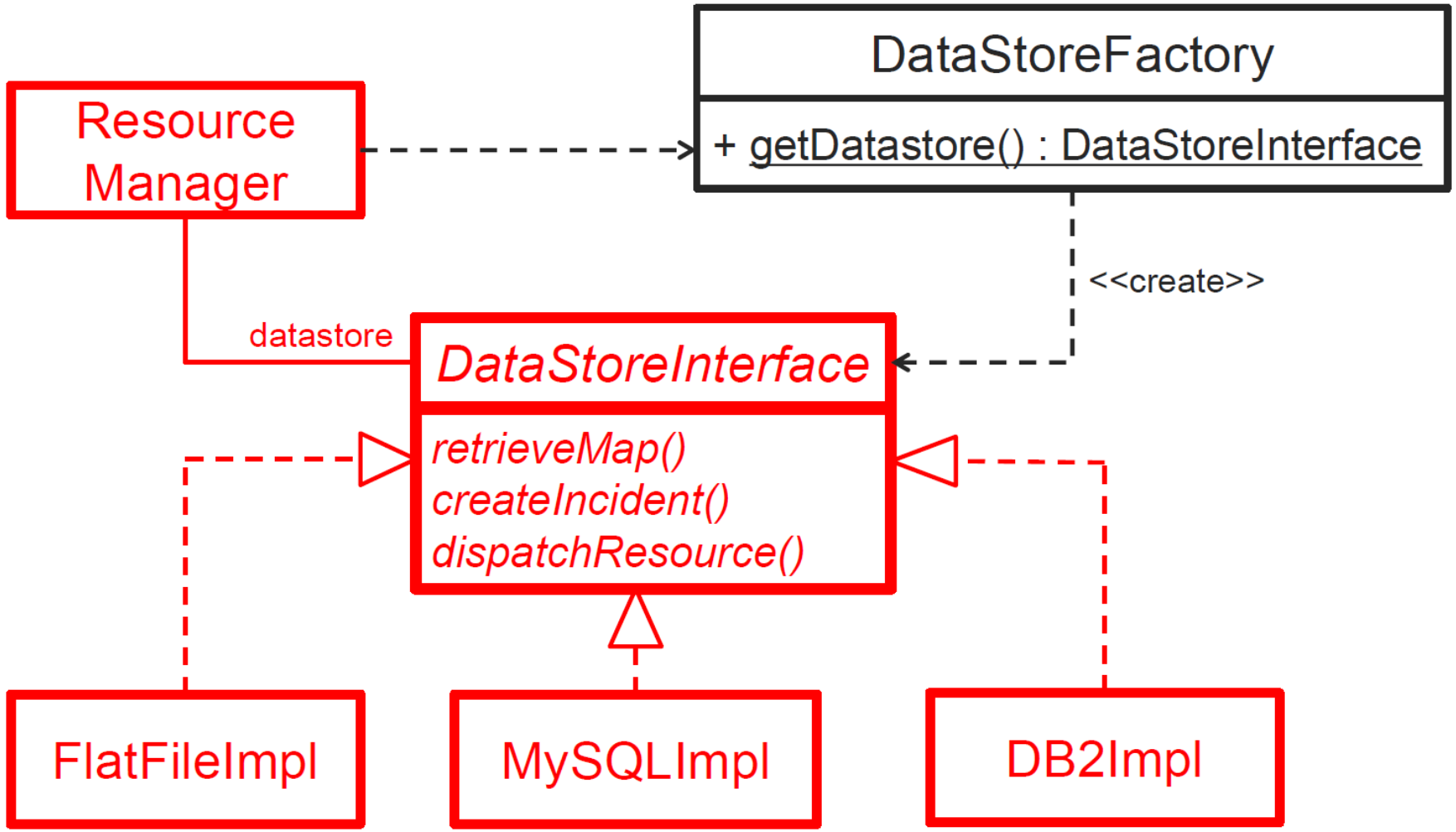
Decouple class selection and object creation (abstract instantiation process) from the client where the object is used allowing greater flexibility in object creation.

- Existing subclasses can be replaced, or new subclasses can be added
- Process of selecting subclass to use and creating object can be complex
- Client does not know ahead of time which class will be used, and the class can be chosen at runtime

Factory Pattern - “Standard” Version (Template)



Data Access: Strategy Design + Factory Design



Question 1

Grain Elevator System (GES)

- Once grain arrives or leaves, GES will notify processing plants. The processing plants can register their interests in specific types of grains. GES will notify the registered processing plants using the plant preferred communication means, including email, SMS, or the combination of these means.
- Note that the interests and preferred communication means of the processing plants vary greatly, and the processing plants can change their interests and preferred communication means over time. Furthermore, GES should be easily extended to support new communication means once they become available.

- a) Please identify the design problem in this feature and suggest a design pattern for addressing this problem.

- b) Illustrate your solution in a class diagram and explain the roles of each class.

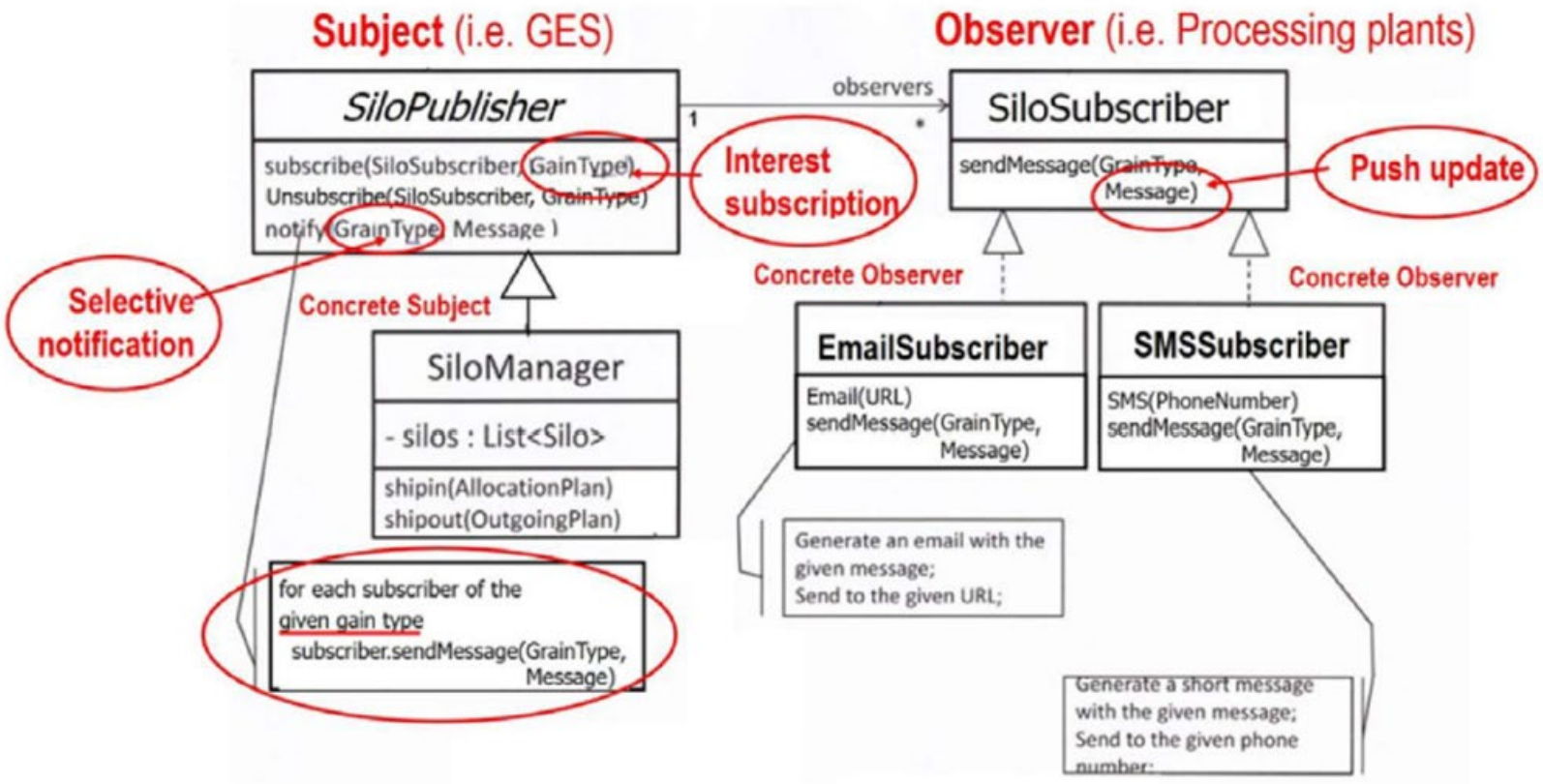
Question 1 (a) - Answer

Design problems:
There is tight coupling between an object (subject) and its dependent objects (observers):
There should be Loose coupling and immediate update between subject and its dependent observers:

- Subject wants to notify observers the change once it occurs, but observers can change their interests freely and constantly - loose coupling.
- Observers want to know the change once it occurs, but they do not know when the change will occur – immediate update.

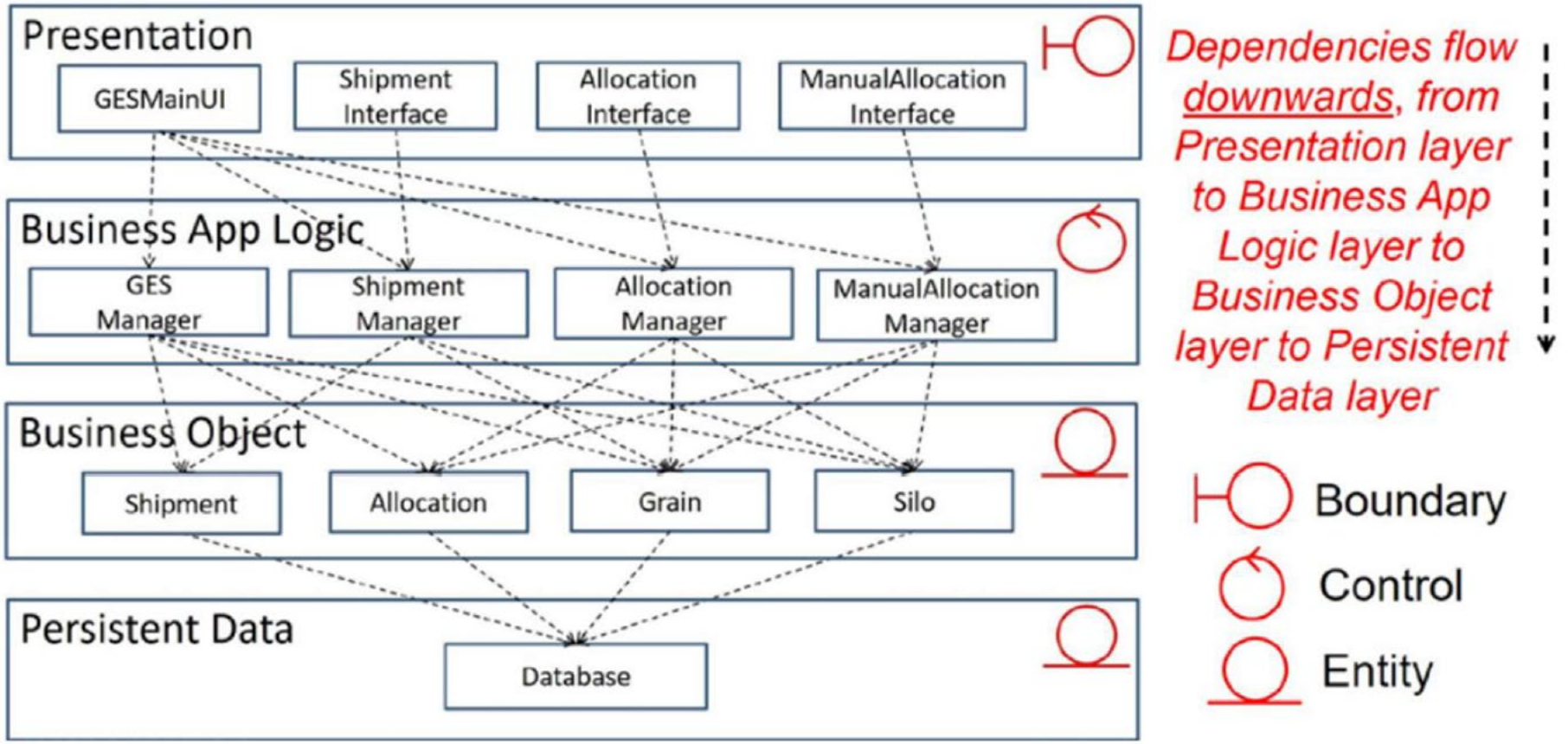
Solution: Observer pattern

Question 1 (b) - Answer



Question 1 - Answer

Layered architecture (4 layers)



Question 2

Refer to the design pattern solution developed in Question 2 of Tutorial#6.

a) Discuss what is missing in the solution

b) Add a relevant design pattern to the design solution to complete the design

Question 2.a - Answer

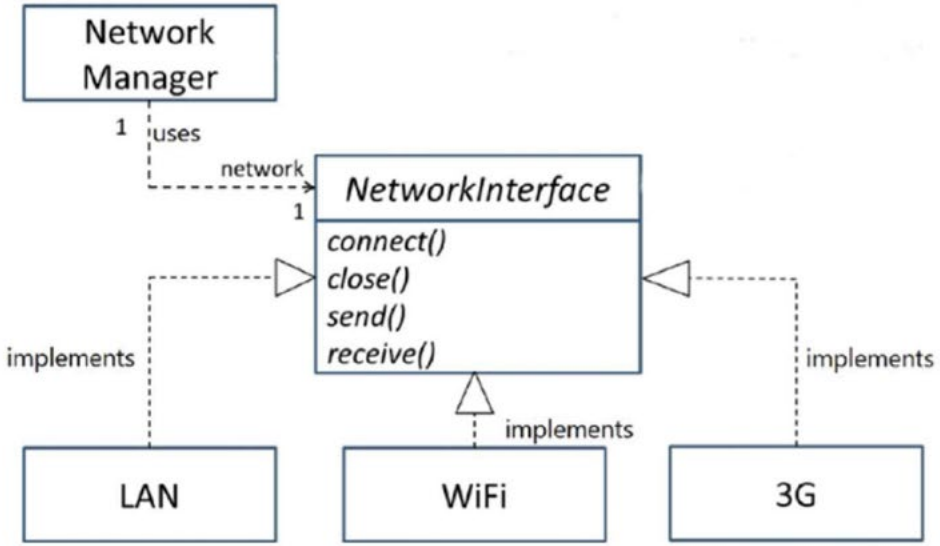
What is missing?

How to handle the object creation of different network interface objects?

Combine factory pattern with strategy pattern.

Question 2.b - Answer

Tutorial6: strategy pattern



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

Combine factory pattern with strategy pattern