

# Design Defects & Restructuring

Week 6: 08 Oct 22

Rahim Hasnani

# Observer Pattern

- ▶ Walk through of Chapter 2 of Head First Design Patterns

# Observer Pattern from GOF

## ► Intent

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

## ► Also Known As

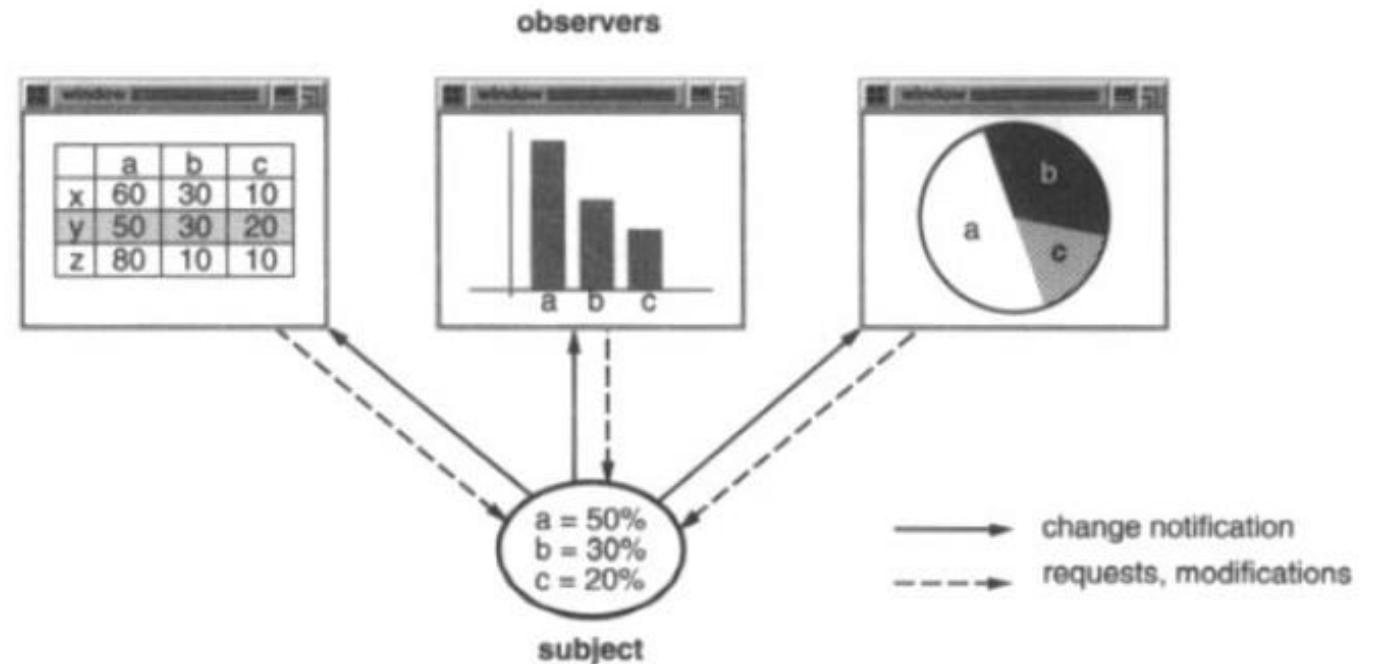
- Dependents, Publish-Subscribe

## ► Motivation

- A common side-effect of partitioning a system into a collection of cooperating classes is the need to maintain consistency between related objects. You don't want to achieve consistency by making the classes tightly coupled, because that reduces their reusability.
- For example in a GUI based system, classes defining application data and presentations can be reused independently. They can work together, too. Both a spreadsheet object and bar chart object can depict information in the same application data object using different presentations. The spreadsheet and the bar chart don't know about each other, thereby letting you reuse only the one you need. But they behave as though they do. When the user changes the information in the spreadsheet, the bar chart reflects the changes immediately and vice versa.

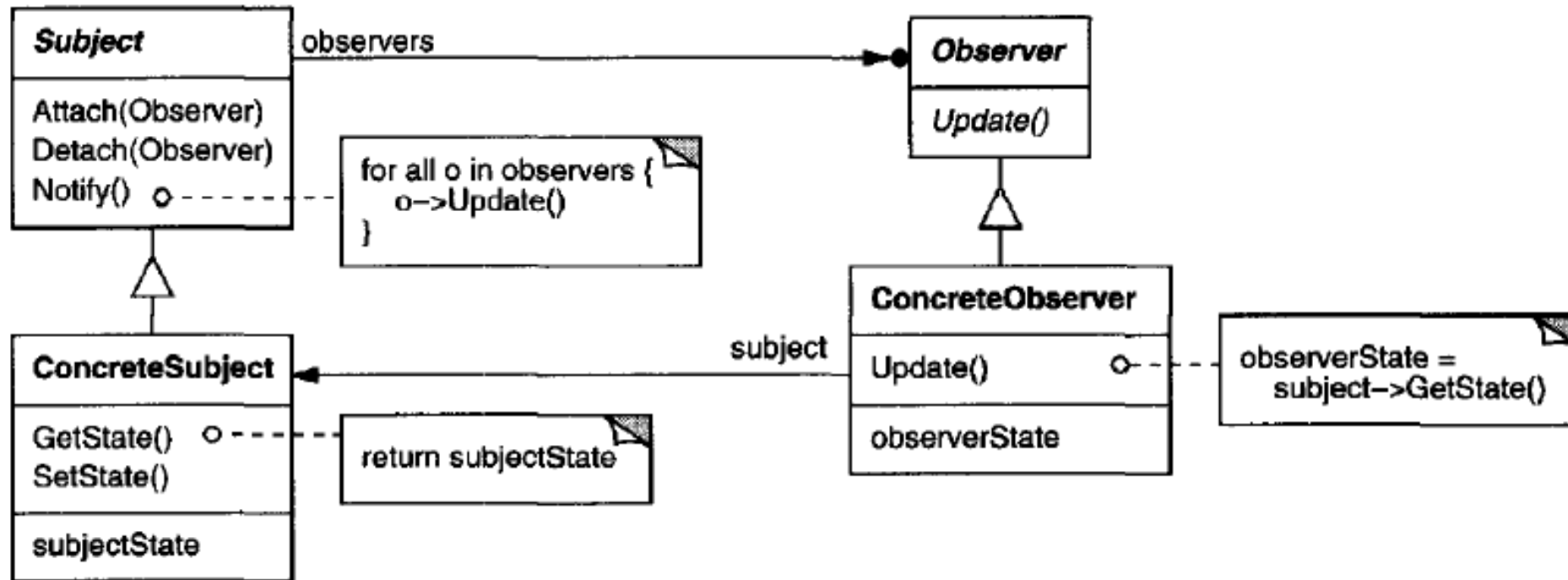
# Observer Pattern from GOF

- The Observer pattern describes how to establish these relationships. The key objects in this pattern are subject and observer. A subject may have any number of dependent observers. All observers are notified whenever the subject undergoes a change in state. In response, each observer will query the subject to synchronize its state with the subject's state.
- This kind of interaction is also known as publish-subscribe. The subject is the publisher of notifications. It sends out these notifications without having to know who its observers are. Any number of observers can subscribe to receive notifications.



# Observer Pattern from GOF

## Structure



# Exercise

- ▶ Maze Code is shared
- ▶ Provide a better design with an additional design goal: flexibility to add more ways to solve the maze, rather than limiting itself to back-tracking only...