

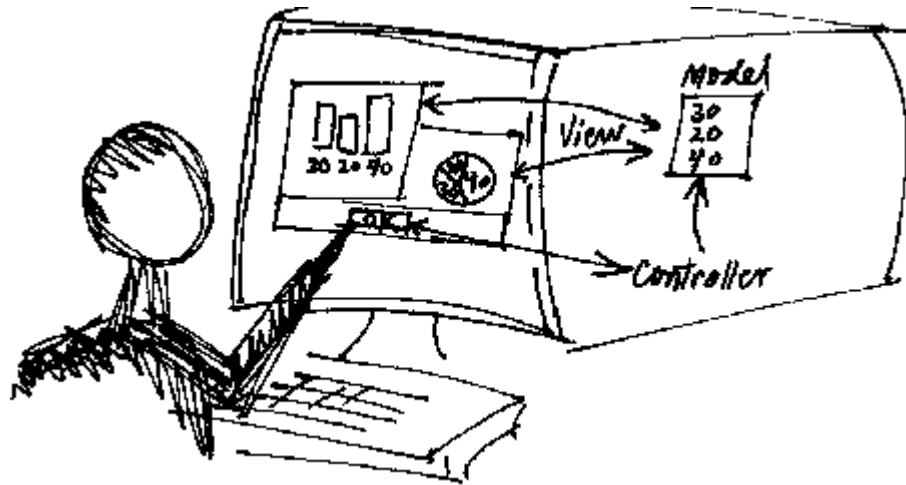
EIF206 - Programación 3

05-Arquitectura MVC

(Prof. José Sánchez, Ph.D.)

Model-View-Controller (MVC)

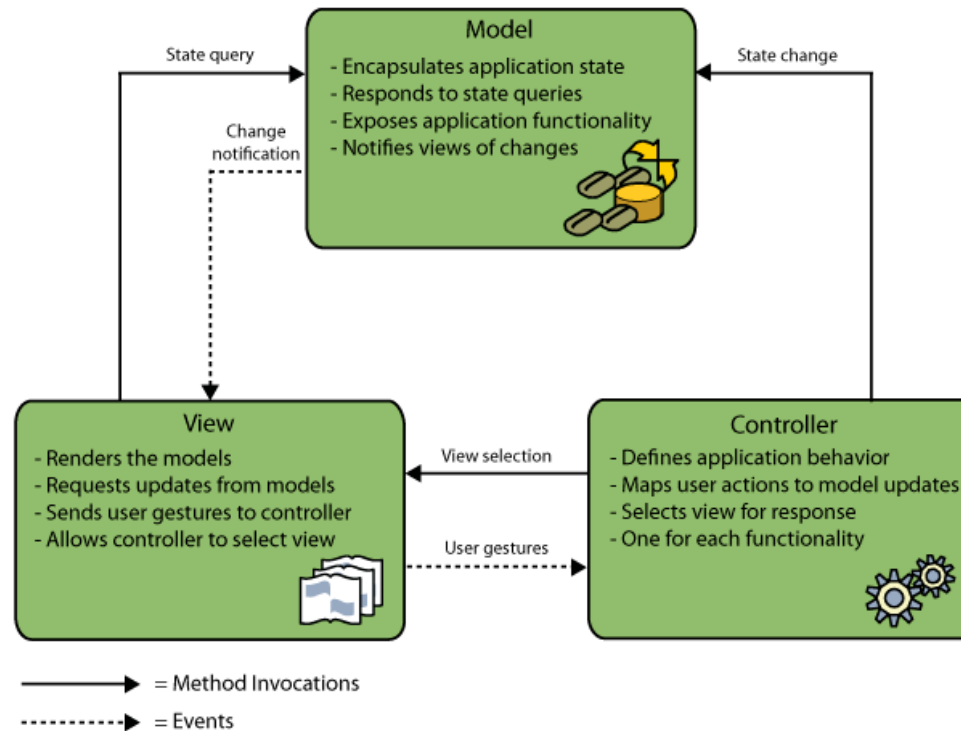
• Es un patrón arquitectural para crear interfaces de usuario que permitan ver y/o editar datos.



<http://people.cs.ksu.edu/~schmidt/CIS200/ch10V12.html>

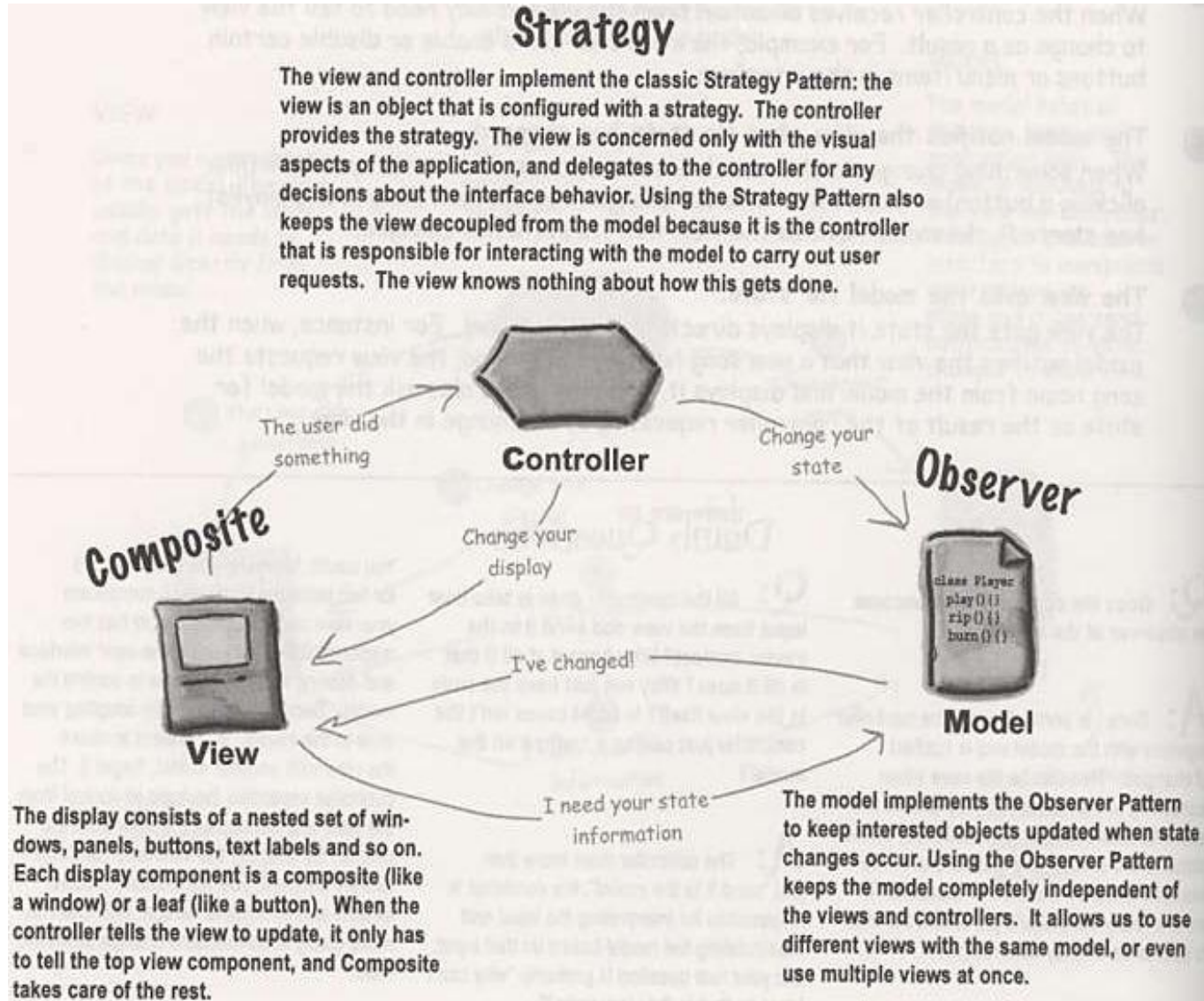
Componentes y Responsabilidades

- Modelo: encapsula los datos o estado.
- Vista: “pinta” los datos de forma conveniente al usuario.
- Controlador: define el comportamiento,actualiza el modelo

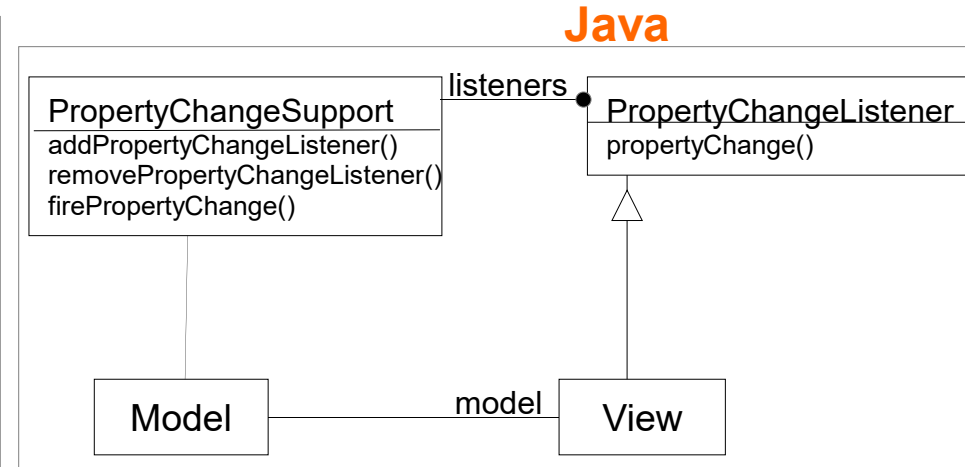
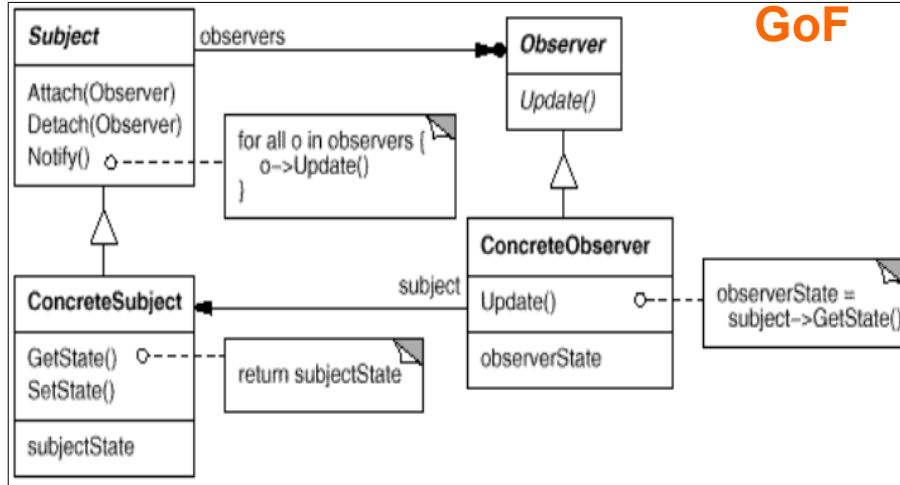


Model-View-Controller (MVC)

“un conjunto de patrones trabajando juntos” [Head-First Design Patterns,p.532]



Model-View: basada PropertyChangedListener



- .Usar clase *PropertyChangedSupport* e interface *PropertyChangedListener* de Java
- .*View* implementa *PropertyChangedListener* y *Model* tiene una instancia de *PropertyChangedSupport*.
- . *View* es registrada con un *PropertyChangedListener* de *Model*
- .Cuando el *Model* cambia usa el método *firePropertyChange* para que invoque el *propertyChange* de *View*

Model-View: basada PropertyChangedListener

Java



- Model hereda de `AbstractModel`, que agrega una instancia de `PropertyChangeSupport` hacia la que delega
- Cualquier propiedad que cambie en Model (`setX..`) invoca el método `firePropertyChange` del delegado
- Este a su vez invocará el método `propertyChange` de cada vista, para que ella se actualice, pidiendo al Model el nuevo valor de la propiedad que cambió.