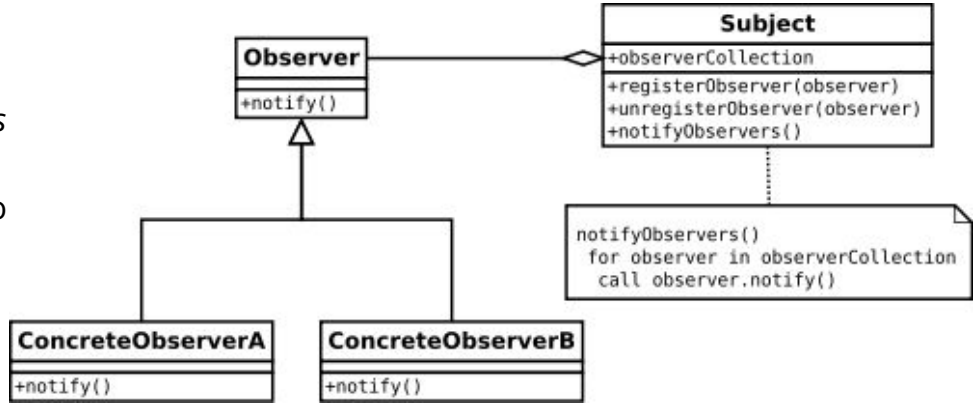
Padrão Observer

Padrão comportamental

Observer

- *"Definir uma dependência um-para-muitos entre objetos para que quando um objeto mudar de estado, todos os seus dependentes sejam notificados e atualizados automaticamente " (GoF)*
- Observar e notificar ações sobre um determinado sujeito
  - Interfaces gráficas, navegadores, redes sociais,
- Princípio
  - Interface observador com método para notificação
  - Objeto sujeito
    - Registra quem receberá os eventos
    - Dispara os eventos



Fonte: wikipedia

Definição e uso

```
interface IObserver {
   public void notify(Object source);
}
class MensagemConfirmaAcao implements IObserver{
     public void notify (Object source) {
         System.out.println("Você clicou no botão.");
     }
}
abstract class Subject {
   ArrayList<IObserver> observerCollection;
   abstract public void register Observer(IObserver observer);
   abstract public void unregister Observer(IObserver observer);
   abstract public void notifyObservers();
}
class Button extends Subject {
   public void registerObserver(IObserver observer){
       observerCollection.add(observer);
   }
   public void unregisterObserver(IObserver observer) {
      observerCollection.remove(observer);
   }
   public void notifyObservers(){
      for (IObserver obs : IObserver )
        obs.notify();
   }
   //deverá ser disparado ao clicar no botão
   public onClick () {
      notifyObservers();
   }
```

```
Button button = new Button();
IObserver observer = new  MensagemConfirmaAcao();
button.registerObserver(observer);
button.onClick();
```

Observer

- Considerações
  - Pode aumentar acoplamento pelo relacionamento bidirecional
  - Incluir parâmetros no *notify* -> passar o objeto

Relacionamento entre os padrões

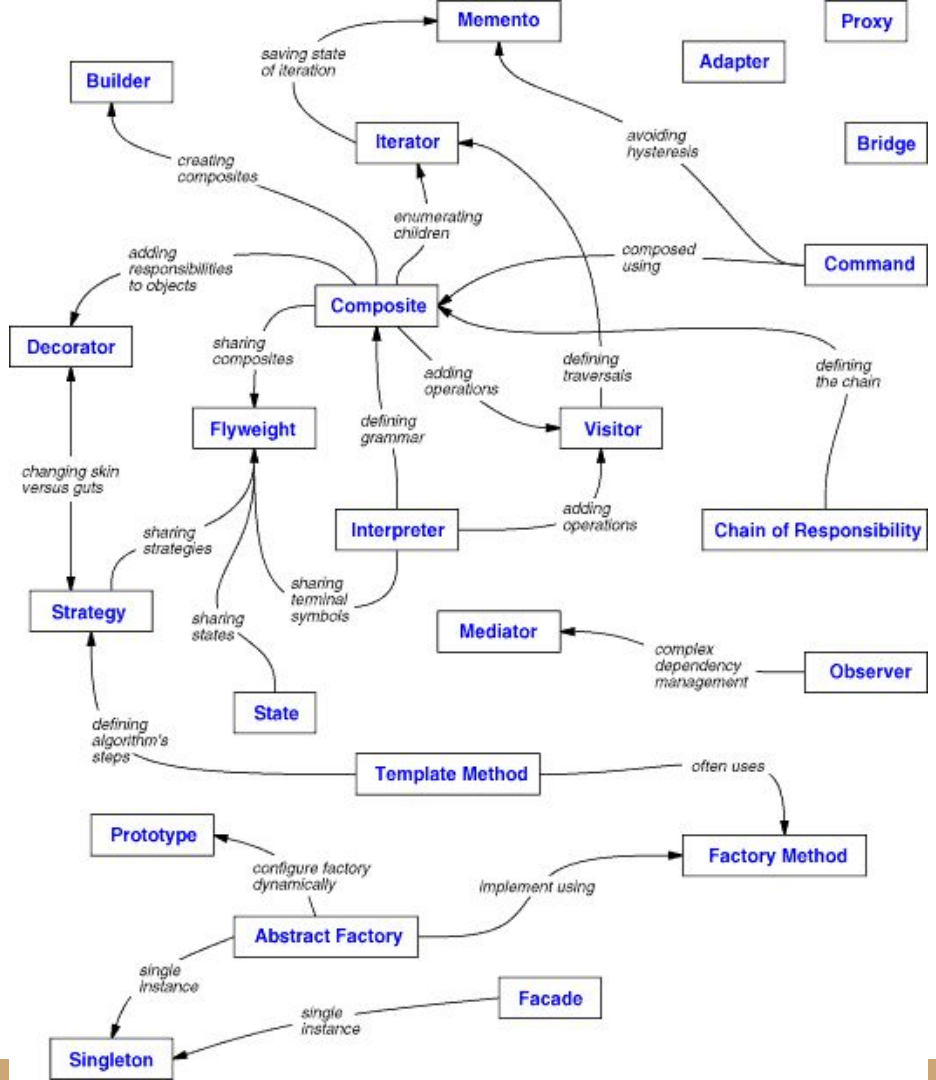


Figura 1.1 do livro