

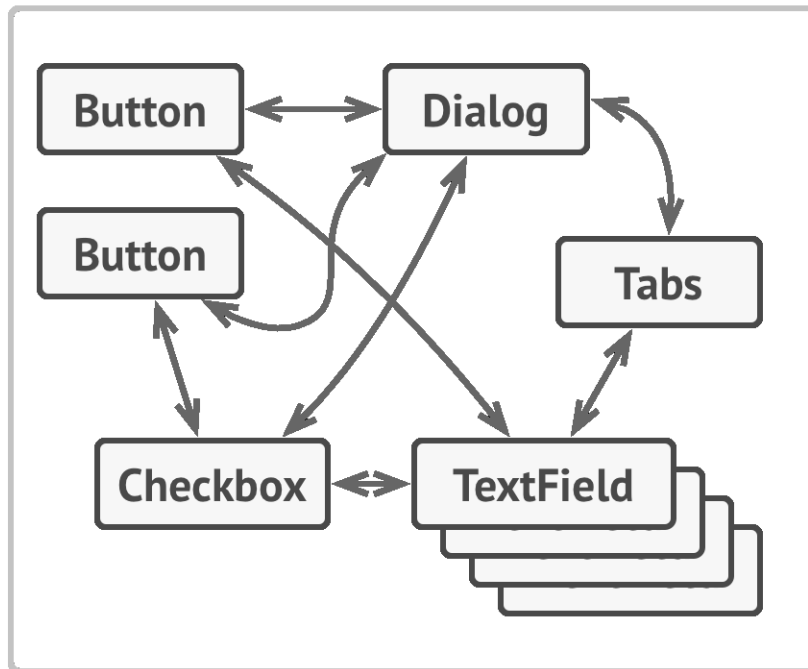
Mediator

also known as: Intermediary, Controller

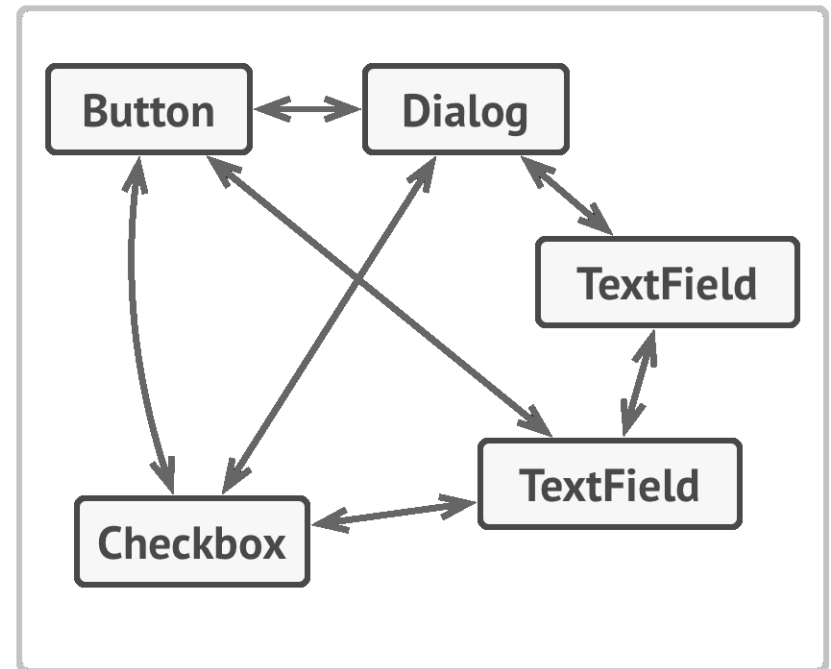
Mediator is a behavioral design pattern that lets you reduce chaotic dependencies between objects. The pattern restricts direct communications between the objects and forces them to collaborate only via a mediator object.

Problem

 Profile Dialog



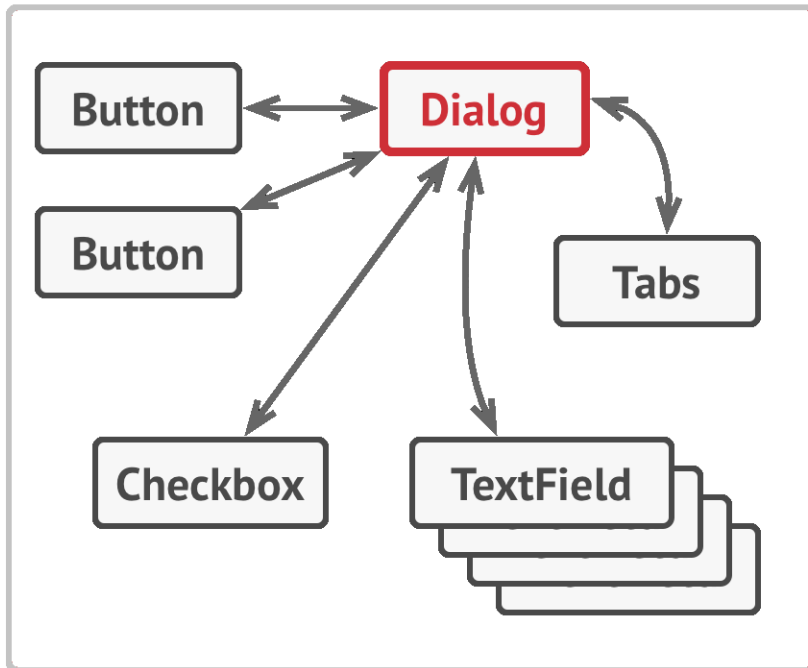
 Login Dialog



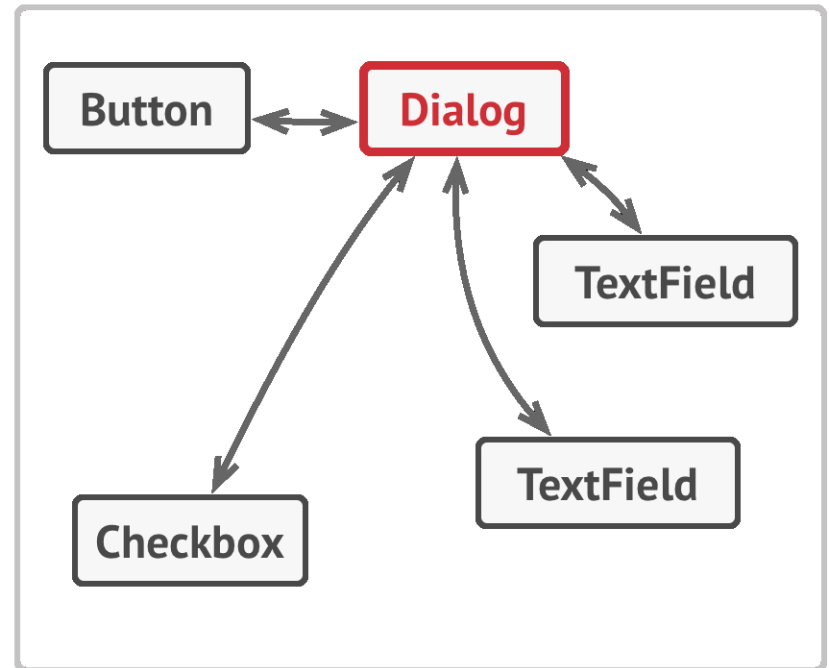
Relations between elements of the user interface can become chaotic as the application evolves.

Solution

 Profile Dialog

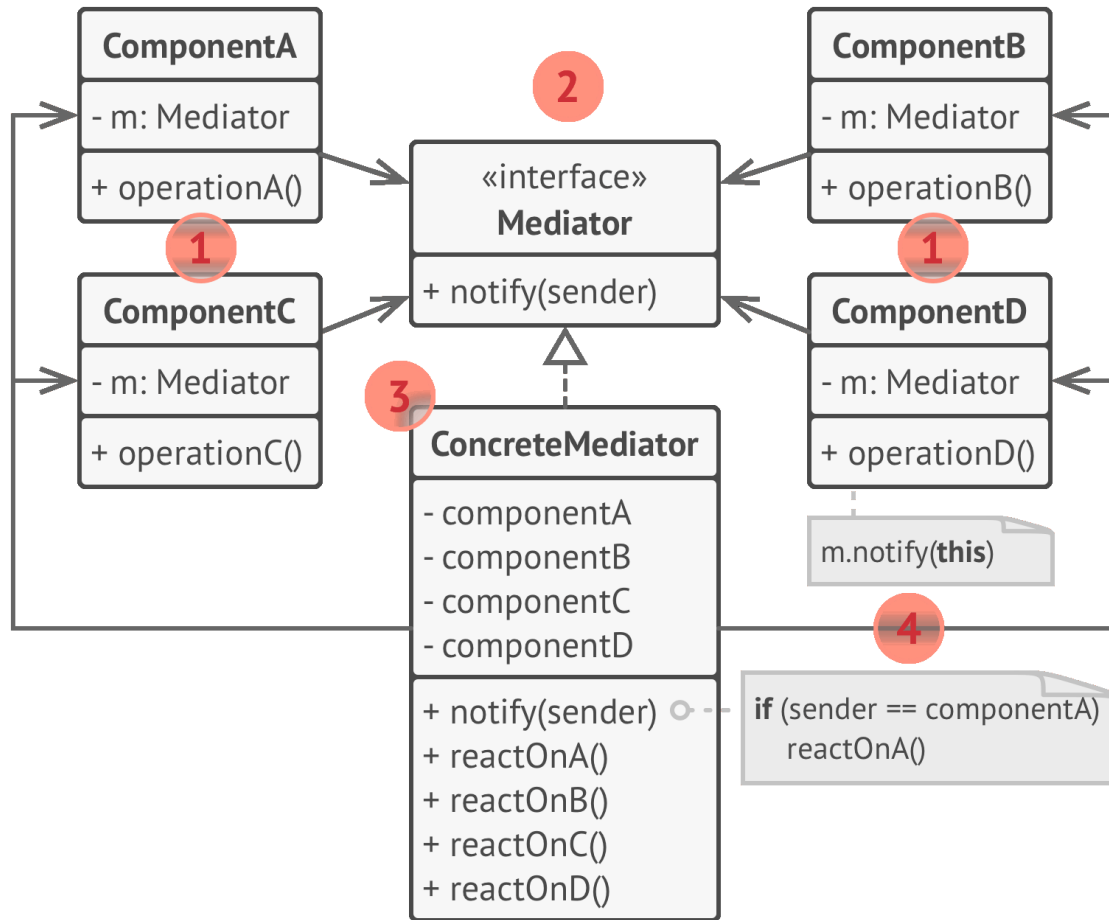


 Login Dialog



UI elements should communicate indirectly, via the mediator object.

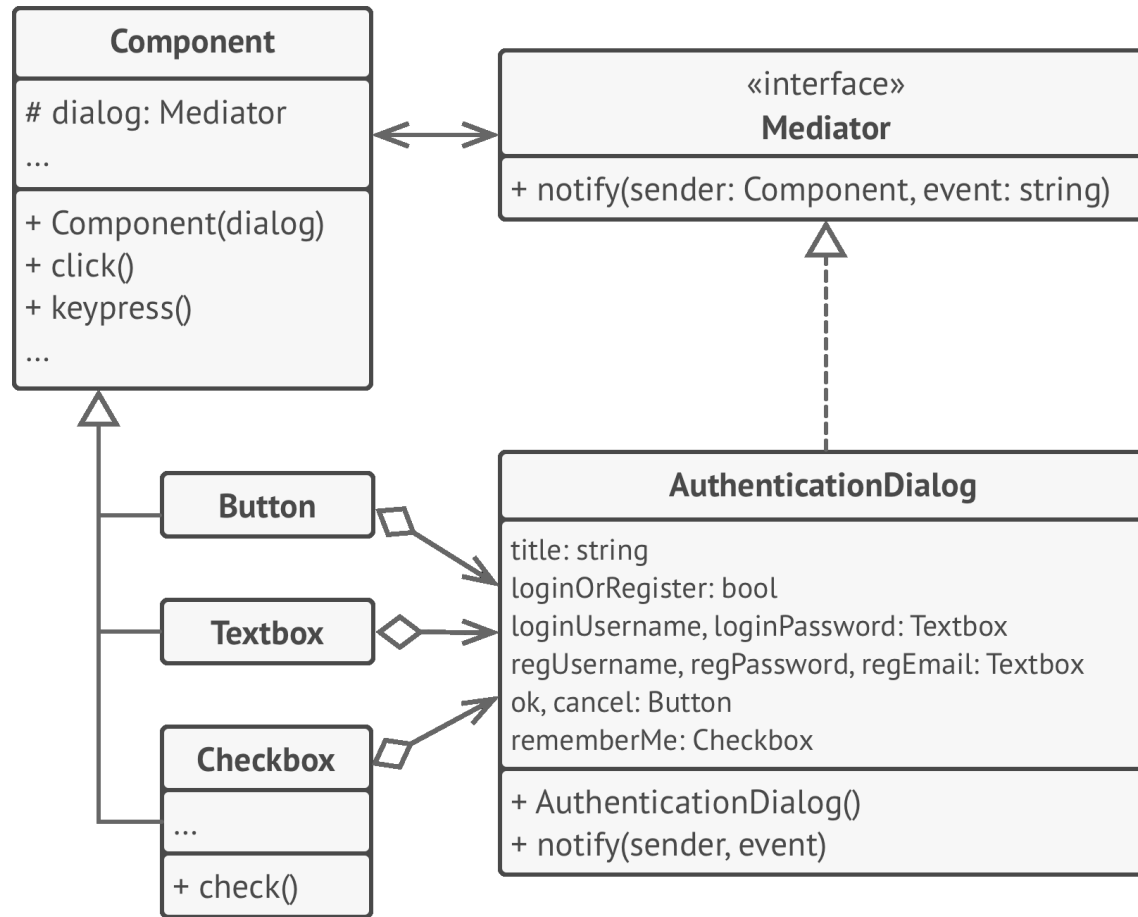
Structure



Contd.

- **Components** are various classes that contain some business logic. Each component has a reference to a mediator, declared with the type of the mediator interface.
- The **Mediator** interface declares methods of communication with components, which usually include just a single notification method.
- **Concrete Mediators** encapsulate relations between various components.
- Components must not be aware of other components. If something important happens within or to a component, it must only notify the mediator.
- From a component's perspective, it all looks like a total black box.

Implementation



Structure of the UI dialog classes.

Application

- Mediator pattern is applicable when it's hard to change some of the classes because they are tightly coupled to a bunch of other classes.
- This pattern is applicable when you can't reuse a component in a different program because it's too dependent on other components.
- Mediator is applicable when you find yourself creating tons of component subclasses just to reuse some basic behavior in various contexts.

Pros & Cons

- *Single Responsibility Principle.* You can extract the communications between various components into a single place, making it easier to comprehend and maintain.
- *Open/Closed Principle.* You can introduce new mediators without having to change the actual components.
- You can reduce coupling between various components of a program
- You can reuse individual components more easily.
- Over time a mediator can evolve into a **God Object**.