

# Mediator

Baris Aktemur  
CS 534 | Ozyegin University

Contents are from “Design Patterns” by Gamma, Helm, Johnson, Vlissides

CS 534 | Ozyegin University

1

# Mediator

- Intent
  - Define an object that encapsulates how a set of objects interact.
  - Mediator promotes loose coupling by keeping objects from referring to each other explicitly, and it lets you vary their interaction independently.

CS 534 | Ozyegin University

2

# Mediator

- Motivation

- An OO system may end up with too many interconnected objects
- It can be difficult to change the system's behavior in any significant way, since behavior is distributed among many objects.
- As a result, you may be forced to define many subclasses to customize the system's behavior.

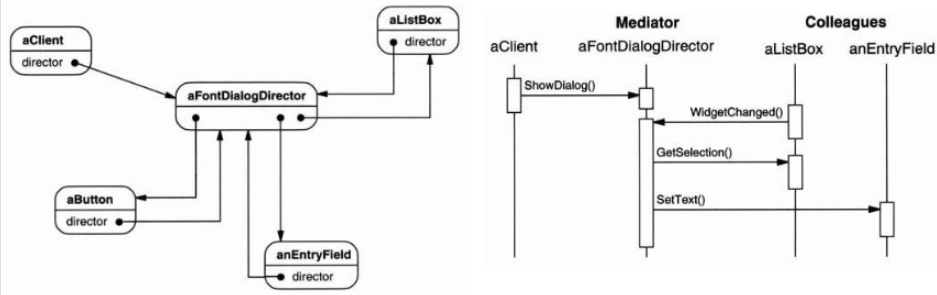
# Mediator

- Many dependencies between the widgets in the dialog.
- E.g: a button gets disabled when a certain entry field is empty.



# Mediator

- A hub of communication

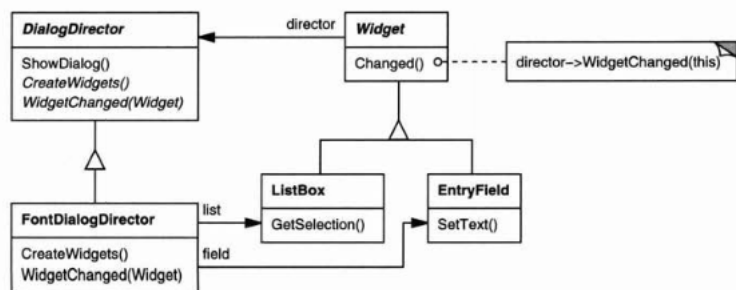


CS 534 | Ozyegin University

5

# Mediator

- Widgets communicate with each other only indirectly, through the director.
- They don't have to know about each other; all they know is the director.



CS 534 | Ozyegin University

6

## Applicability

- Use the Mediator pattern when
  - a set of objects communicate in well-defined but complex ways. The resulting interdependencies are unstructured and difficult to understand.
  - reusing an object is difficult because it refers to and communicates with many other objects.
  - a behavior that's distributed between several classes should be customizable without a lot of subclassing.

## Consequences

- A mediator promotes loose coupling between colleagues.
- It centralizes control. The Mediator pattern trades complexity of interaction for complexity in the mediator.