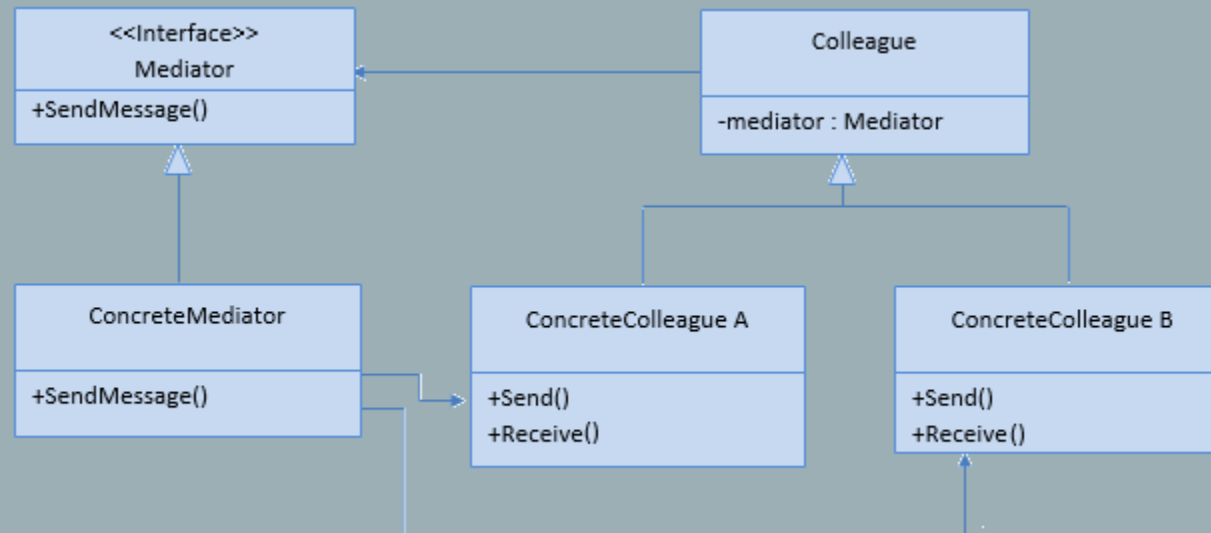


Mediator

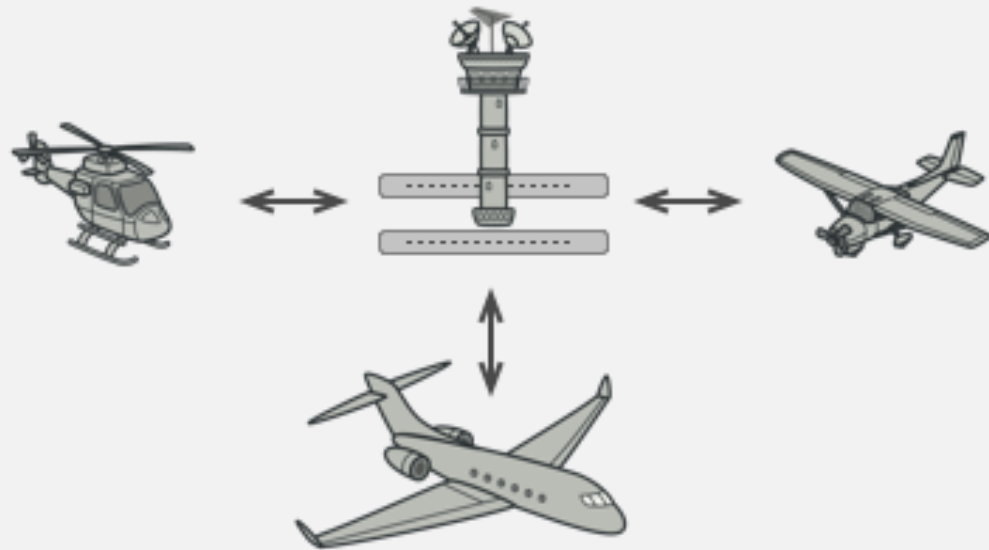
Design Pattern

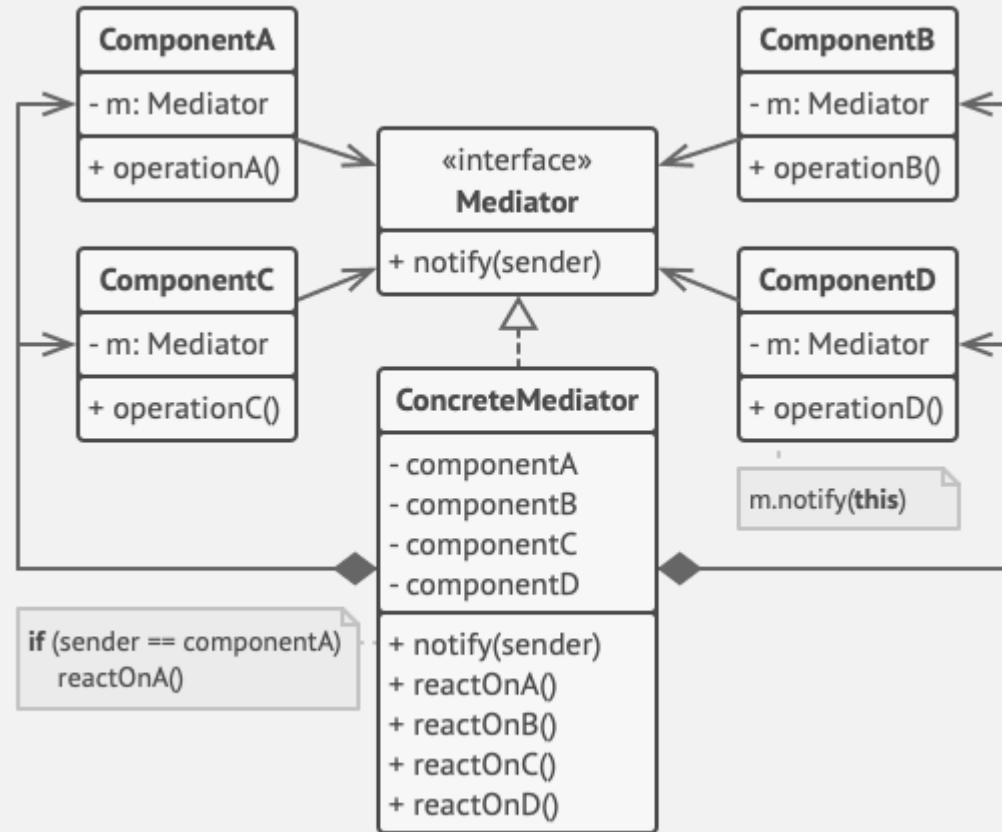




- Tight coupling
- Chaotic dependencies
- Direct communication
- Component reuse problem

- Loose coupling
- Reduce dependencies
- Restricts direct communication
- Component reuse
- Force objects to collaborate only via mediator





PROS AND 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".