

# Mediator Pattern

Benjamin Cooper

# Don't Let This Happen to Your Classes



# Mediator

- \* Provides a unified interface to a set of interfaces in a subsystem.

# What Does This Mean?



- \* Communication between classes can get complex and difficult to maintain
- \* Sometimes an object (mediator) can step in and handle this communication issue
- \* You don't want your planes to crash

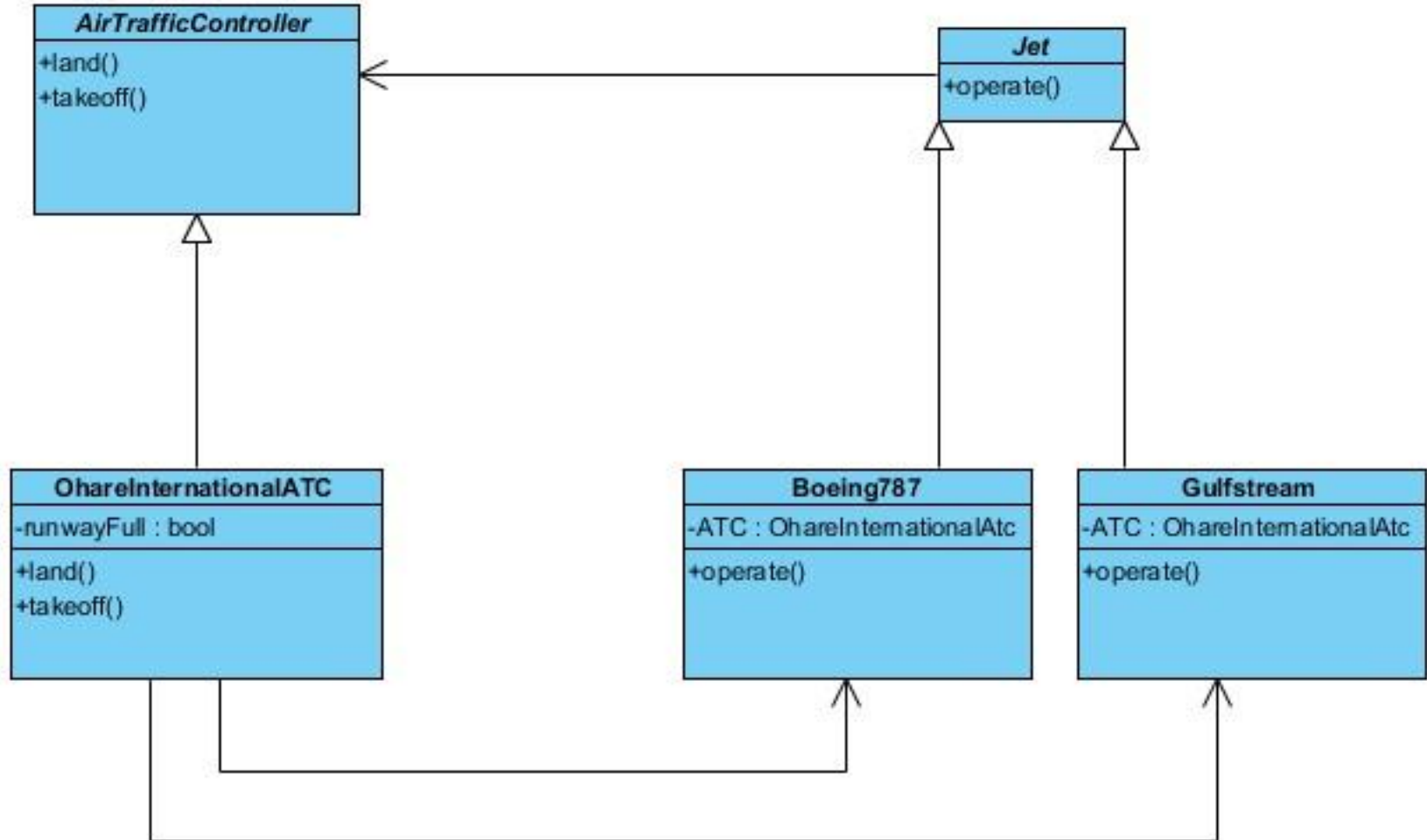
# Problem

- \* Breaking a system up into objects generally helps reusability
- \* Interconnections between the objects tend to reduce it.
- \* Can leave us with objects dependent on other objects and a mess if we need to change the system

# Solution

- \* Encapsulate behavior in a mediator object
- \* The mediator will control and coordinate interactions between the objects, promoting loose coupling
- \* Now the objects only know the mediator and don't have to deal with the other planes in the sky

# UML



# Considerations

- \* Limits subclassing
- \* Decouples colleagues
- \* Simplifies object protocols
- \* Abstracts how objects cooperate
- \* Centralizes control



# Examples

- \* GUI Library – Dialog class is a mediator for GUI controls
- \* Chat App – The Chatroom is a mediator for the participants



OR...

# Example

- \* O'Hare ... with one runway and two crazy pilots

# References

- \* Design Patterns, Gamma, et. al., Addison Wesley, 1995
- \* <http://johnlindquist.com/2010/10/13/patterncraft-mediator-pattern-vs-framework-mediators/>
- \* [http://sourcemaking.com/design\\_patterns/mediator](http://sourcemaking.com/design_patterns/mediator)
- \* [http://en.wikipedia.org/wiki/Mediator\\_pattern](http://en.wikipedia.org/wiki/Mediator_pattern)
- \* <http://www.oodesign.com/mediator-pattern.html>