The Command Design Pattern

CSPP51023 - Object Oriented Architecture, Design, and Methodology

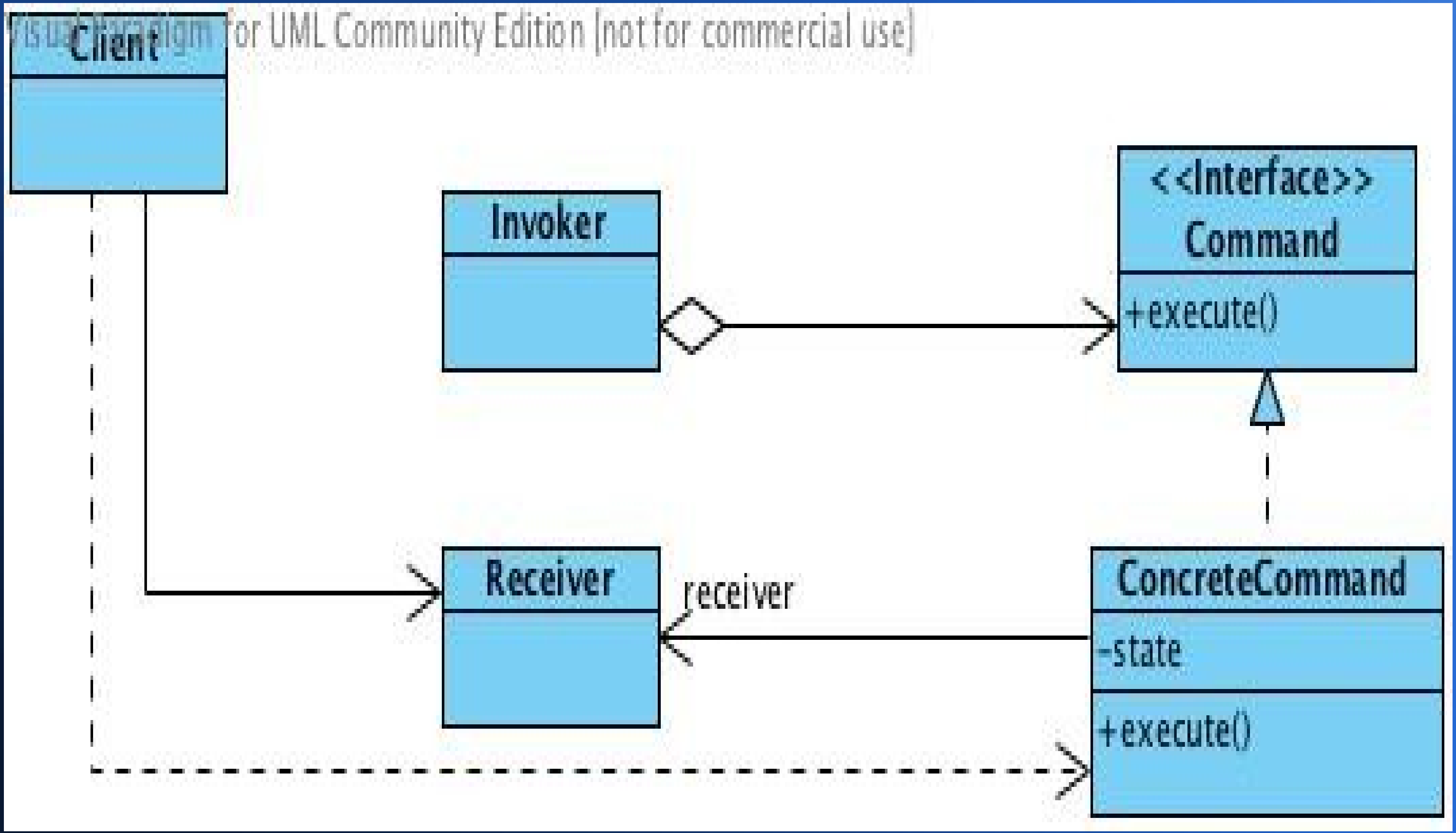
Why Use Commands?

- Decouple request invoker from request receiver
- Parameterize clients with requests
- Log or queue requests (wizards)
- Support “undoable” operations

Roles In The Command Pattern

- Command – defines the command interface
- ConcreteCommand – realization of the Command interface
- Client - Application
- Invoker – asks command to execute
- Receiver – carries out the details of the request

Generic Structure

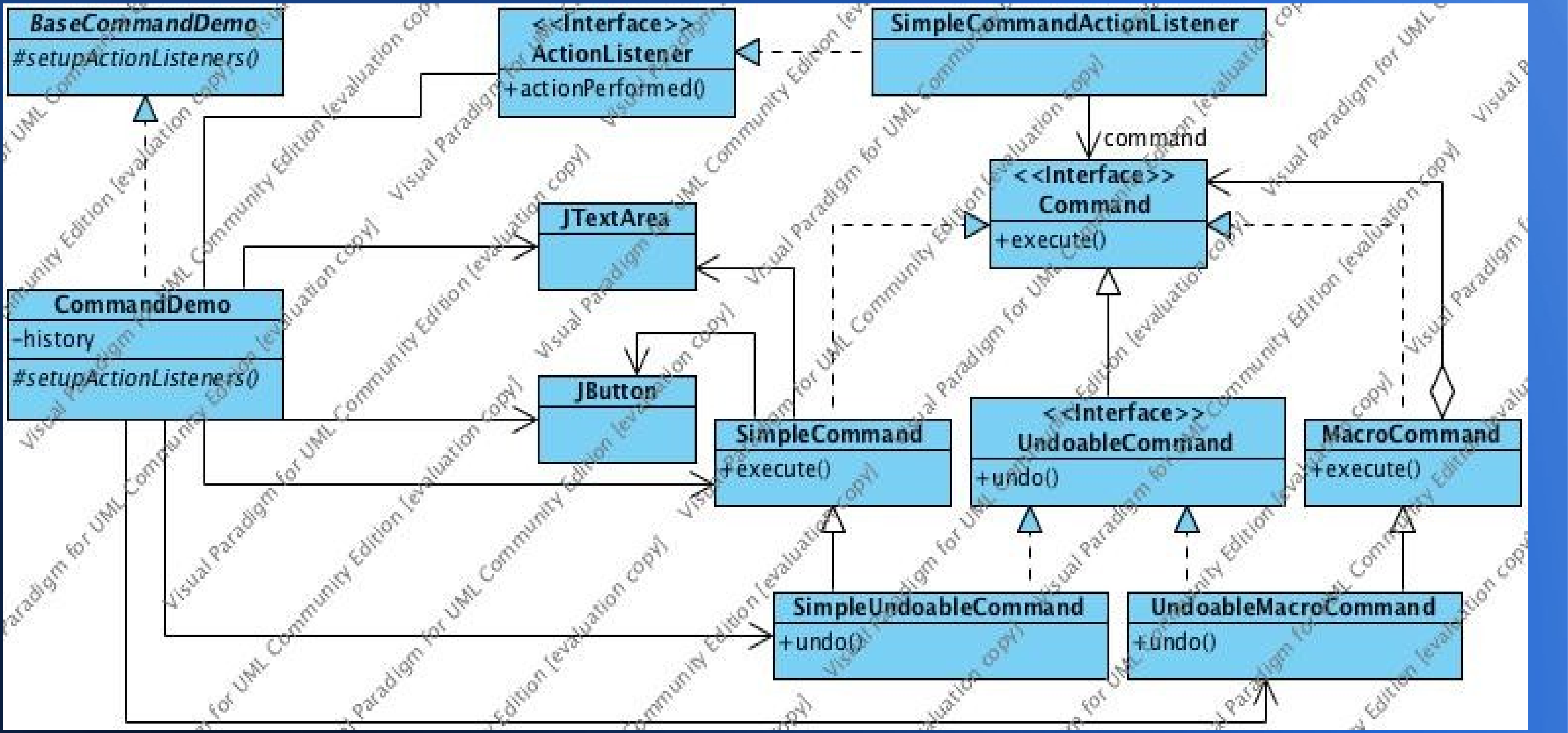


Example 1

- A simple Java Swing Application where buttons add text to specific text boxes, change button functions or undo operations.
- Main Actors

Actor	**Role**
CommandDemo	Client
SimpleCommandActionListener	Invoker
JTextArea JButton	Receiver
SimpleCommand SimpleUndoableCommand UndoableMacroCommand	Command

Example 1 UML

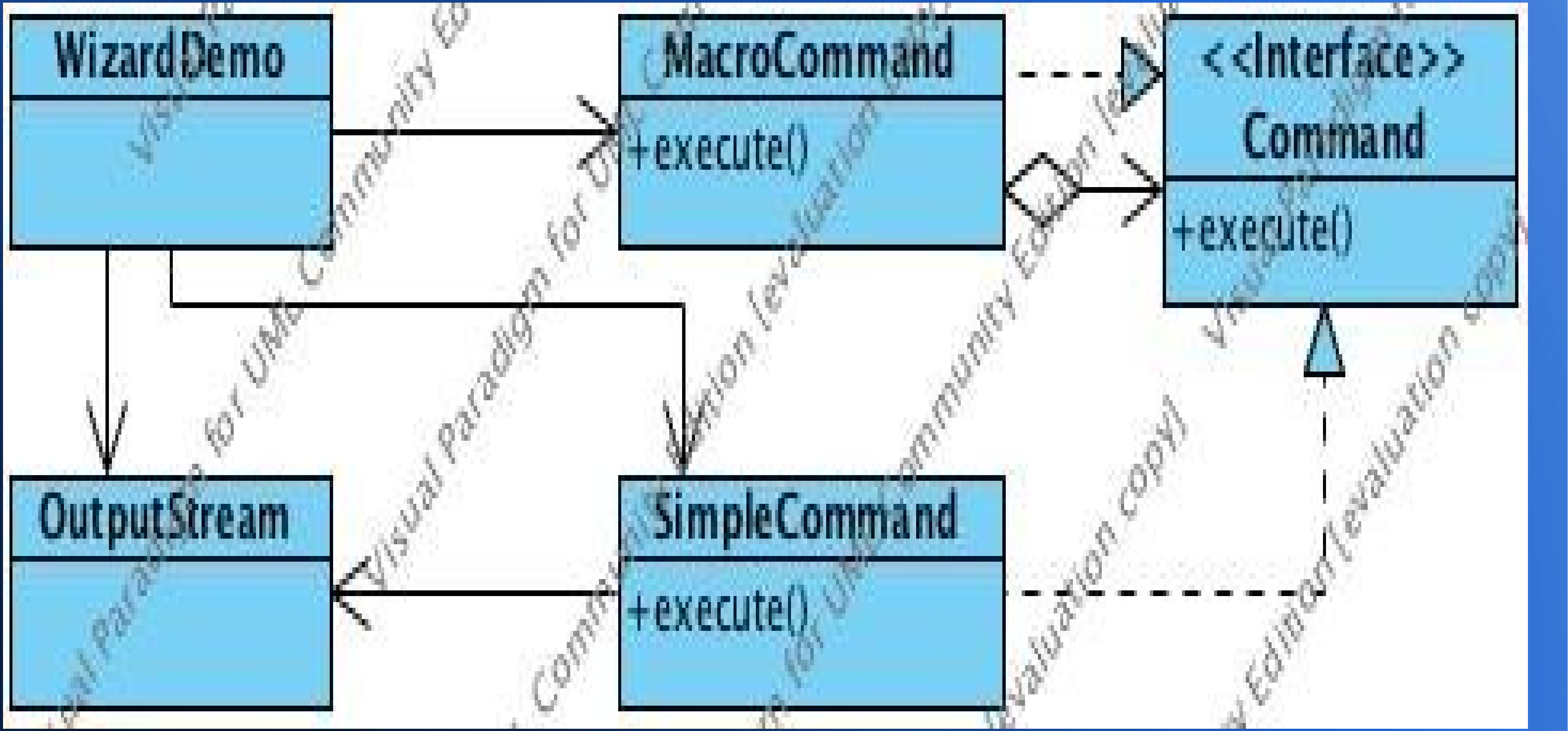


Example 2

- A simple Java command line application that batches commands like a wizard
- Main Actors

Actor	Role
WizardDemo	Client/Invoker
SimpleCommand MacroCommand	Command
OutputStream	Receiver

Example 2 UML



Summary

- Use the Command pattern when you need to issue requests to a receiver without knowing about the operation being requested or the receiver of the request
- Also know as: Action, Transaction