Command Pattern

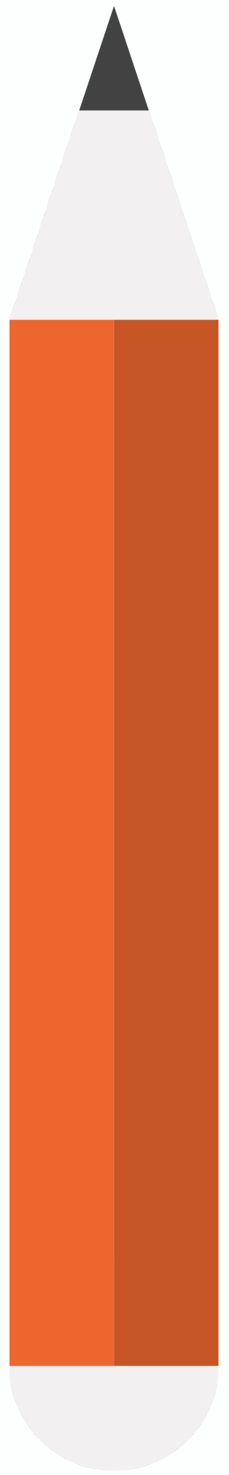
Bryan Hansen

twitter: bh5k | http://www.linkedin.com/in/hansenbryan

Concepts

- Encapsulate request as an Object
- Object-oriented callback
- Decouple sender from processor
- Often used for “undo” functionality
- Examples:
  - java.lang.Runnable
  - javax.swing.Action

Design



Object per command

Command Interface

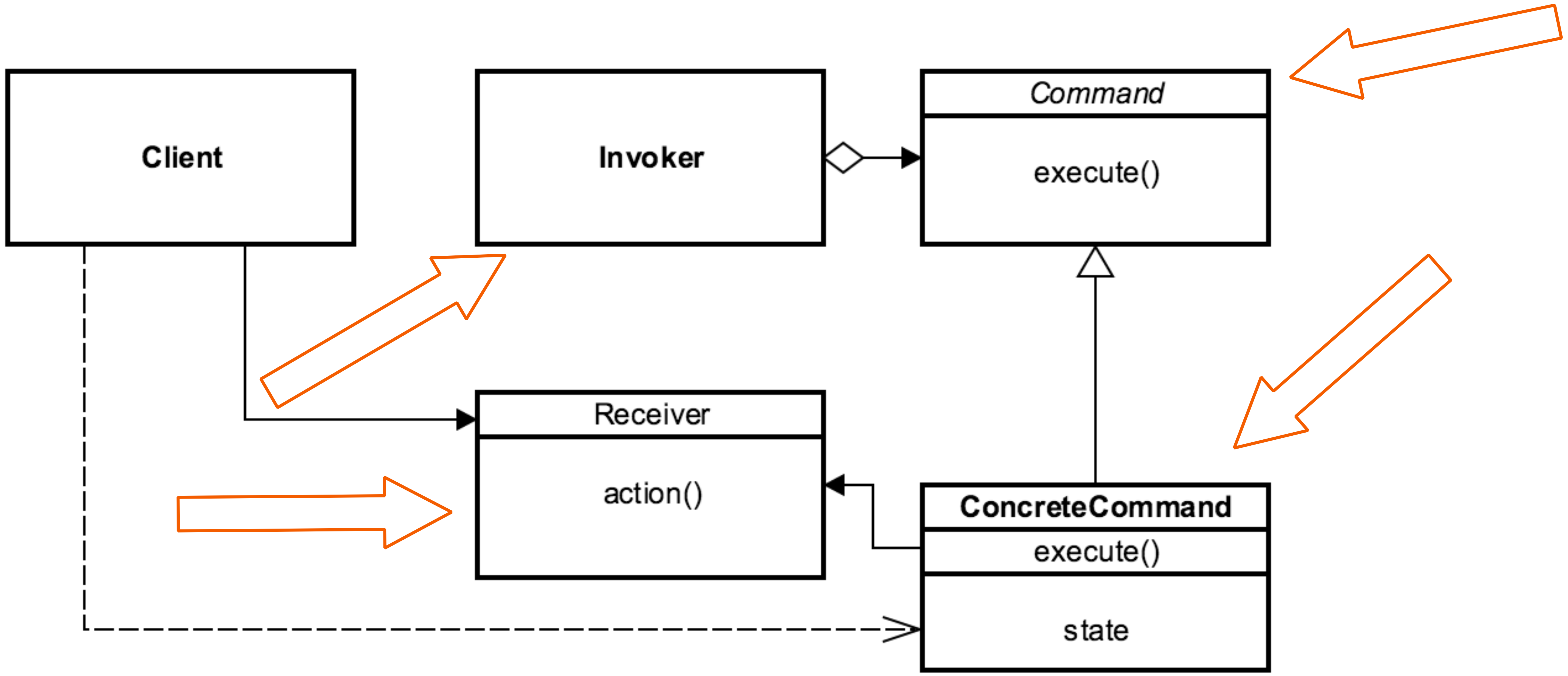
Execute Method

‘Unexecute’ method

Reflection

Command, Invoker, ConcreteCommand

UML


Client
Invoker
Command
execute()
Receiver
action()
ConcreteCommand
execute()
state

Everyday Example - Runnable

```
Task task1 = new Task(10, 12);
Task task2 = new Task(11, 13);

Thread thread1 = new Thread(task1);
thread1.start();

Thread thread2 = new Thread(task2);
thread2.start();
```

Exercise Command

Command, Invoker, ConcreteCommand, Receiver

Manage State

Macro Command

Pitfalls

- Dependence on other patterns
- Multiple Commands
- Make use of Memento
- Prototype for copies

Contrast

Command

- Object per command
- Class contains the ‘what’
- Encapsulates action

Strategy

- Object per strategy
- Class contains the ‘how’
- Encapsulates algorithm

Command Summary

- Encapsulate each request as an object
- Decouple sender from processor
- Very few drawbacks
- Often used for undo functionality