# Command Pattern

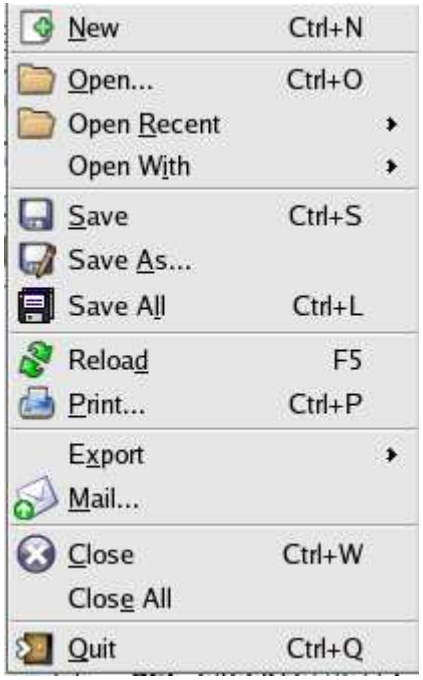
**Marc Provost**
**McGill University**
marc.provost@mail.mcgill.ca

**April 1, 2005**

# Command Pattern

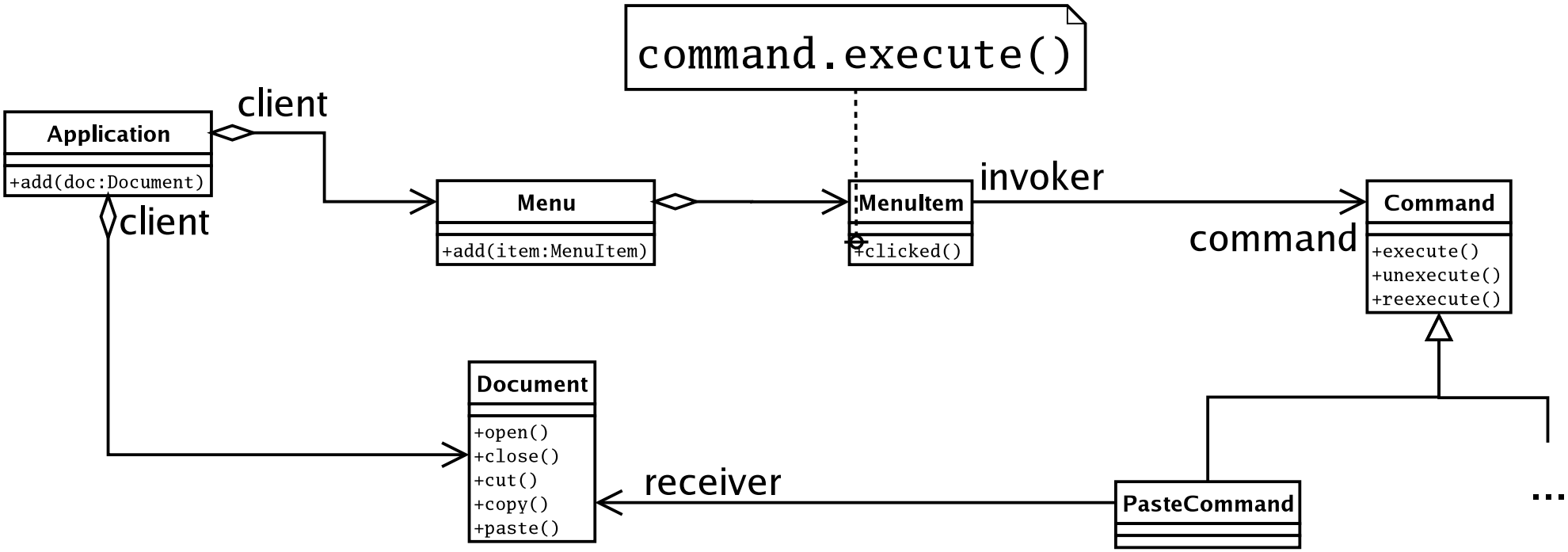
- Separates an operation from the object that executes it.
- With the Command Pattern, it is possible to parametrise an object with an operation.
- Support undo/redo
- Possible to execute the request at a different time. By passing the command object to another process.

# Classical example



# Why?

- Each item in the menu is conceptually the same object.
- The only difference is with the action that is taken when pressed.
- Solution: parametrise the menu item object with a command.


command.execute()
client
Application
+add(doc:Document)
client
Menu
+add(item:MenuItem)
MenuItem
+clicked()
invoker
Command
command
+execute()
+unexecute()
+reexecute()
Document
+open()
+close()
+cut()
+copy()
+paste()
receiver
PasteCommand
...

# Supporting Undo/Redo

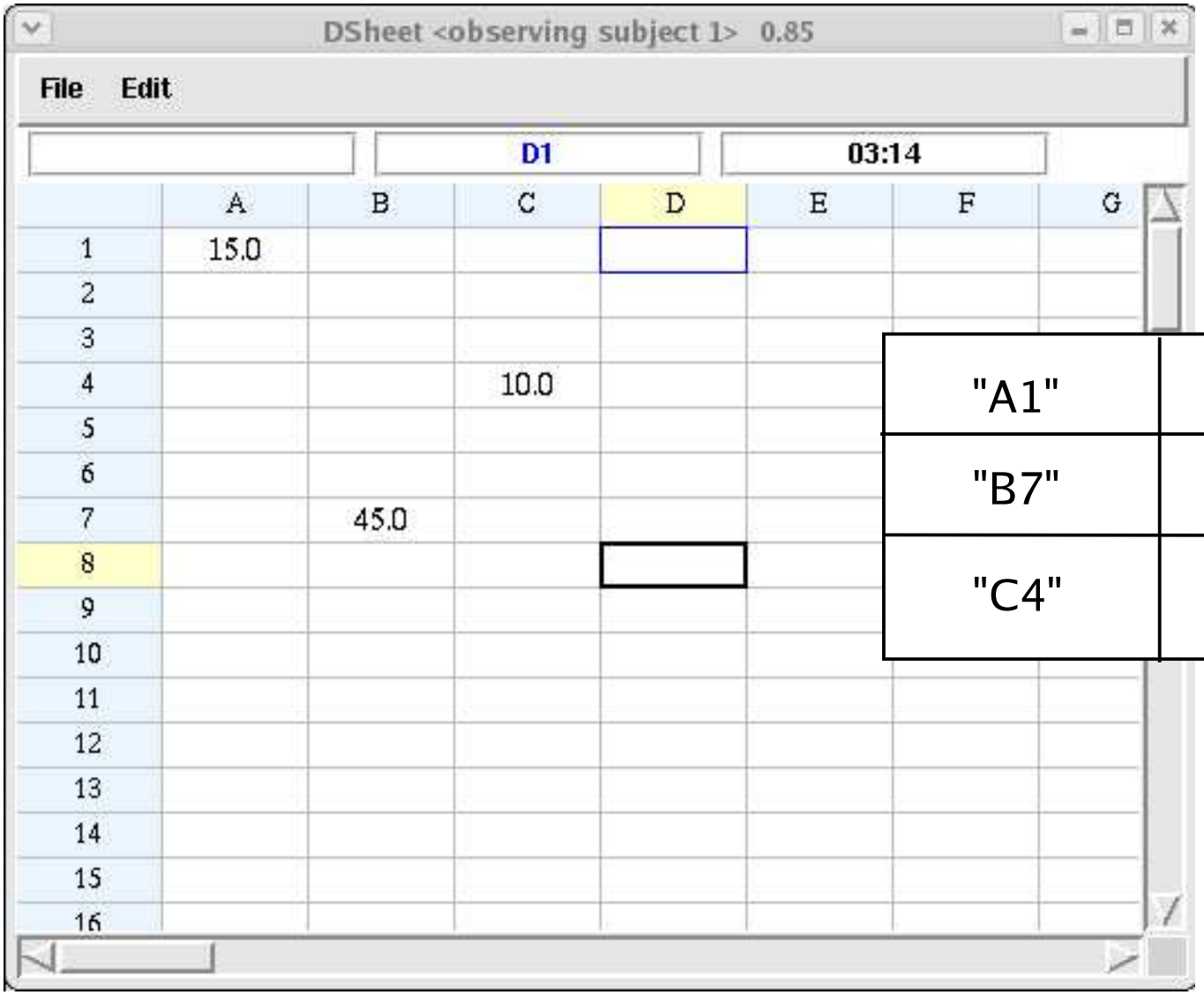
- Since a command is an object, it can hold a state.
- A command object could store the information required to undo or redo itself.
- Use an history list of commands objects.

# Supporting Undo/Redo

- Each command should know how to undo and redo itself (one level).

- A command manager hold the history list of commands:
  - $[commandA, commandB, commandC, ...]$
  - Moving backward: undoing commands
  - Moving foward: redoing commands

- Let's go over an example...

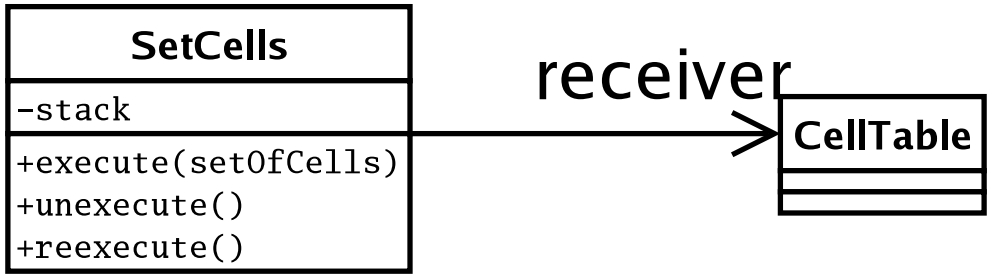



HashTable:

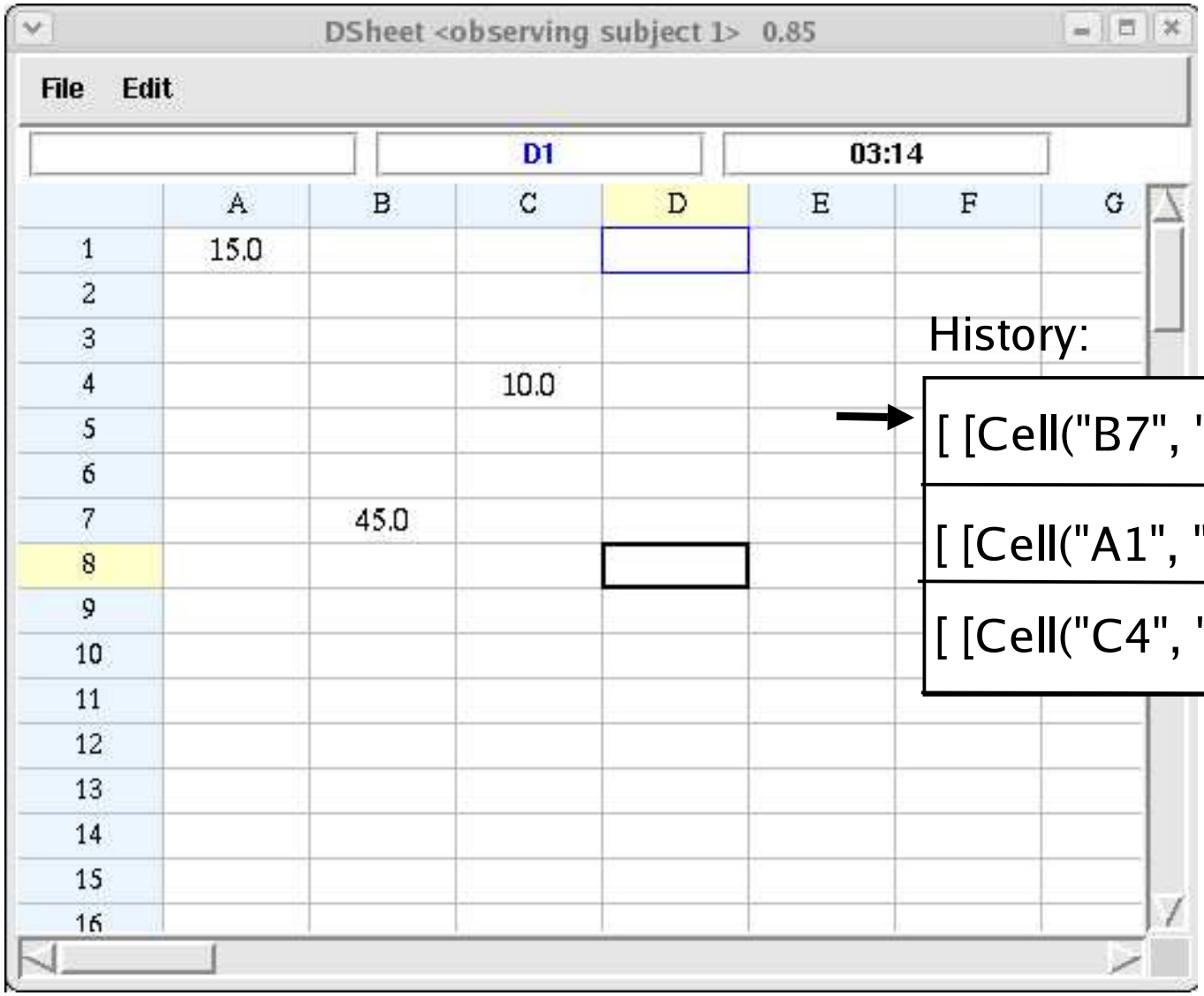
| | |
|---|---|
| "A1" | Cell("A1", "=5+C4") |
| "B7" | Cell("B7", "=45") |
| "C4" | Cell("C4", "=10") |

# SetCells Command

- 'SetCells' command, which acting on the previous hashtable is used to support undo/redo
- The history list is stored directly in the setcells command. (Unique command)




- Each time a set of cells is modified by the user, the *difference* between the previous state and the next state is added in the history list.

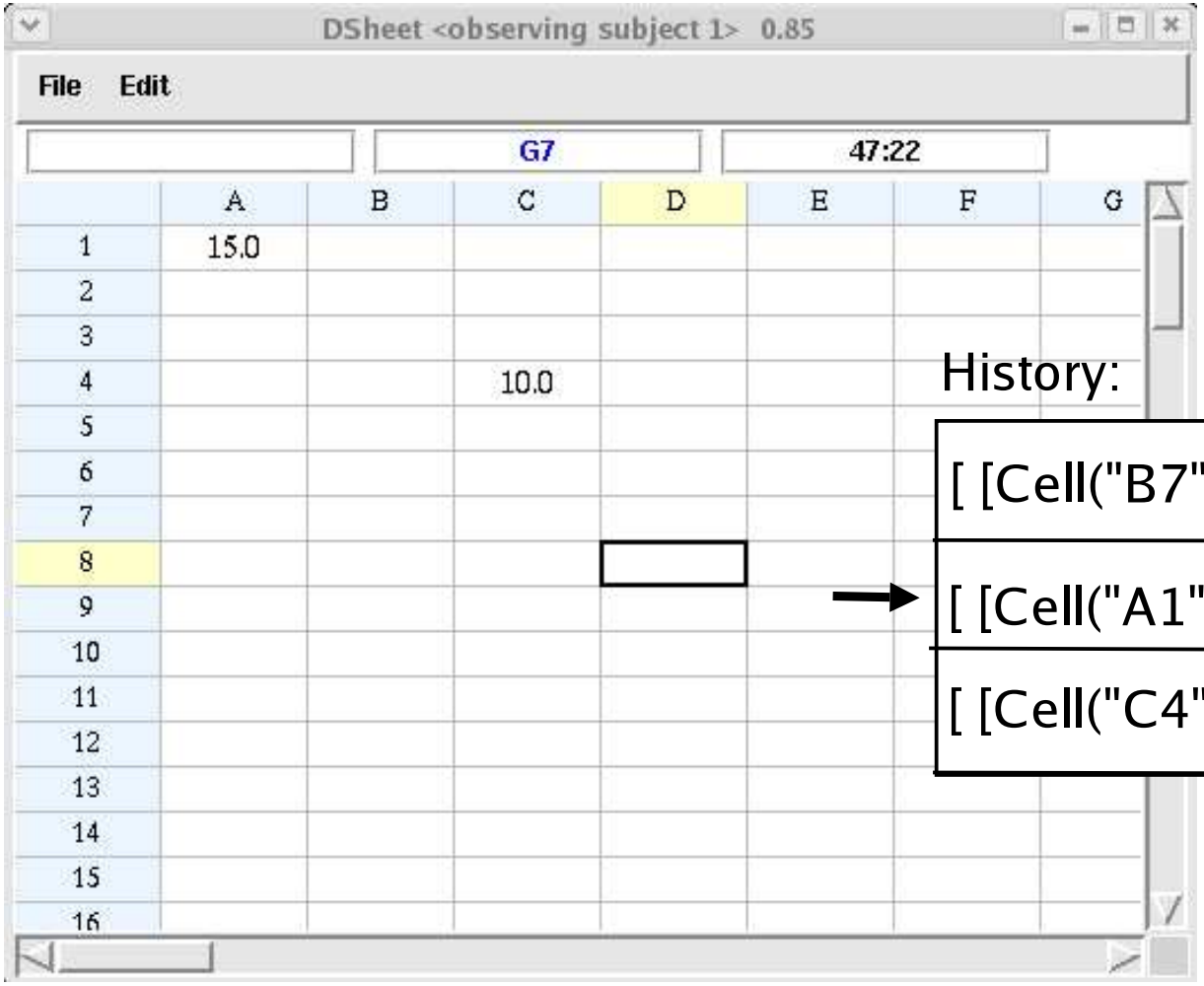



History:

[ [Cell("B7", ""), Cell("B7", "=45")] ]

[ [Cell("A1", ""), Cell("A1", "=5+C4")] ]

[ [Cell("C4", ""), Cell("C4", "=10")] ]

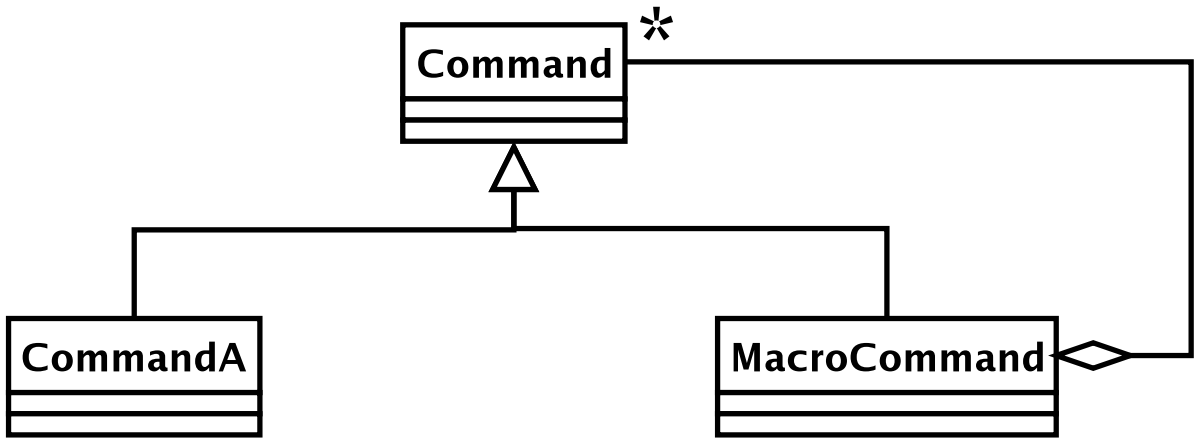


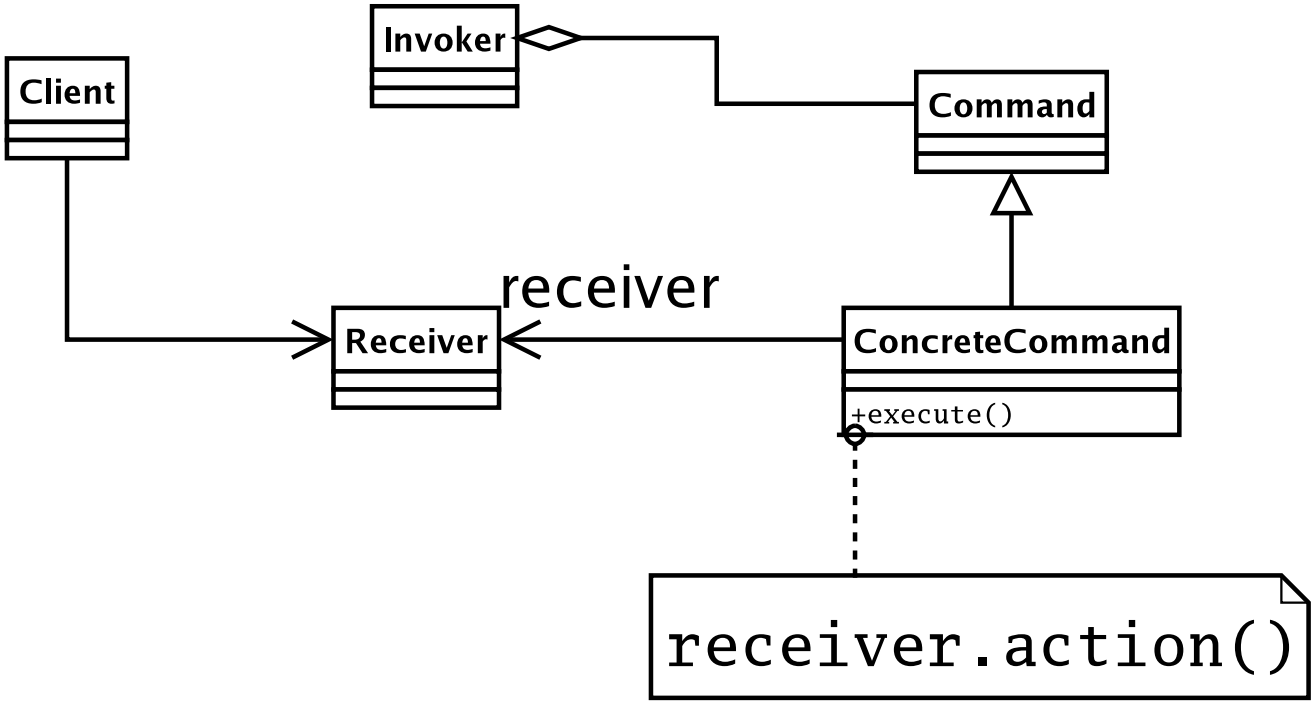

History:

[ [Cell("B7", ""), Cell("B7", "=45")] ]

[ [Cell("A1", ""), Cell("A1", "=5+C4")] ]

[ [Cell("C4", ""), Cell("C4", "=10")] ]


Command
*
CommandA
MacroCommand


Invoker
Client
Command
receiver
Receiver
ConcreteCommand
+execute()
receiver.action()