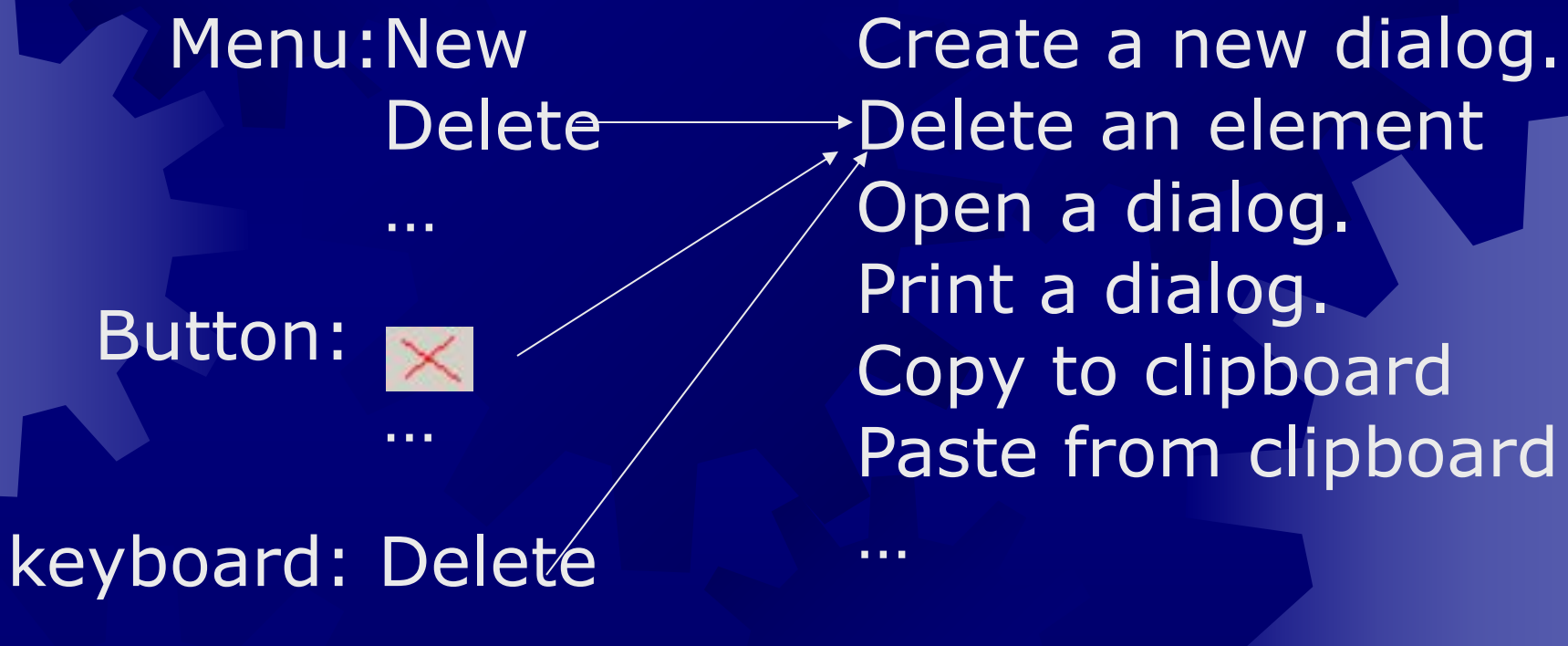
The Command Pattern

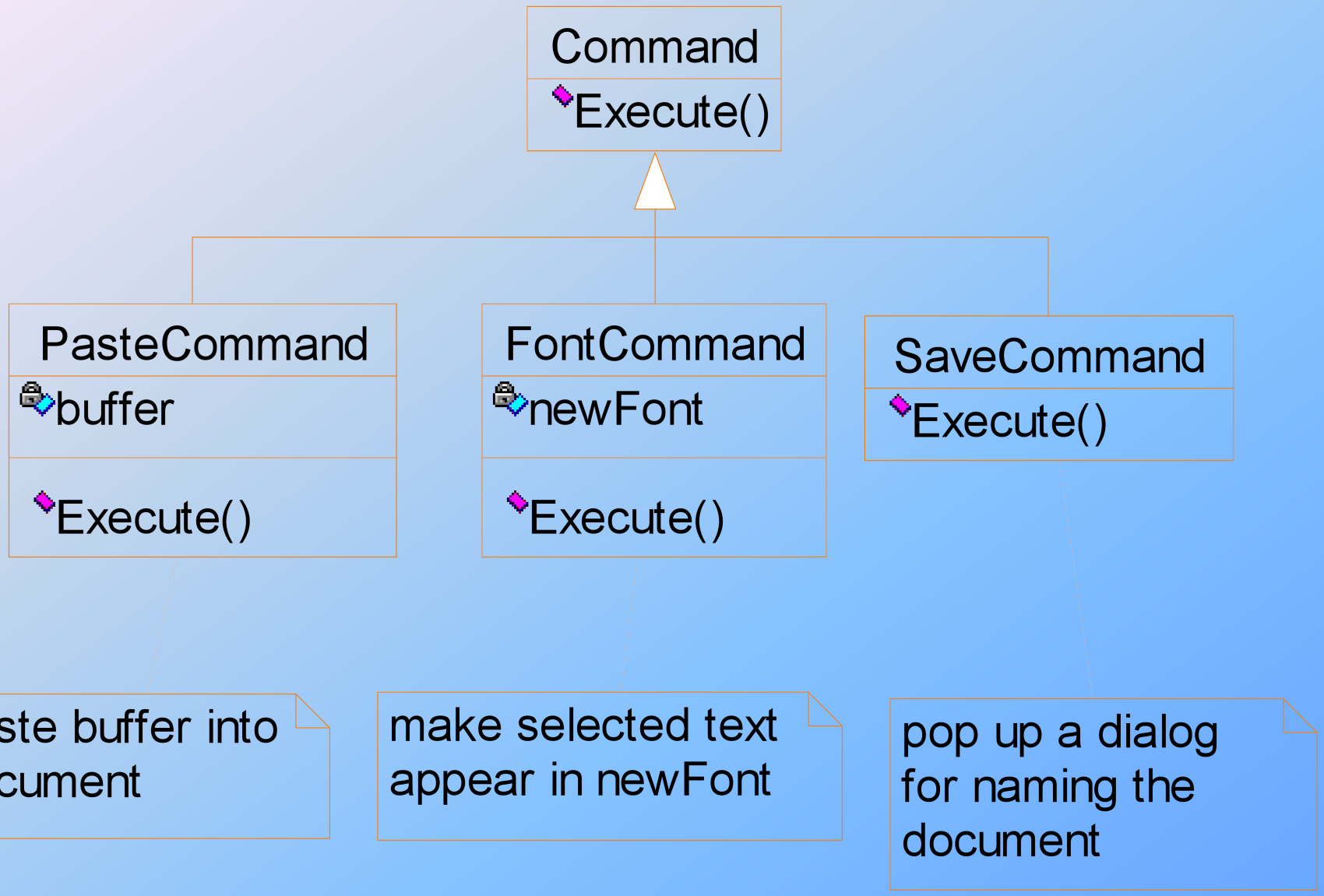


Requirements

- Multiple widgets can be mapped onto the same operation
- We do not want the close coupling between the classes that implement the operations and the interface classes.
- We want to support the Undo/Redo functionality

MFC solution

- ✓ MFC solved the first two requirements by the message mapping mechanism
  - How to implement the undo/redo? Operation ⇔ Function coupling can not solve the problem!
- ➢ We abstract the concept that varies to form a class.

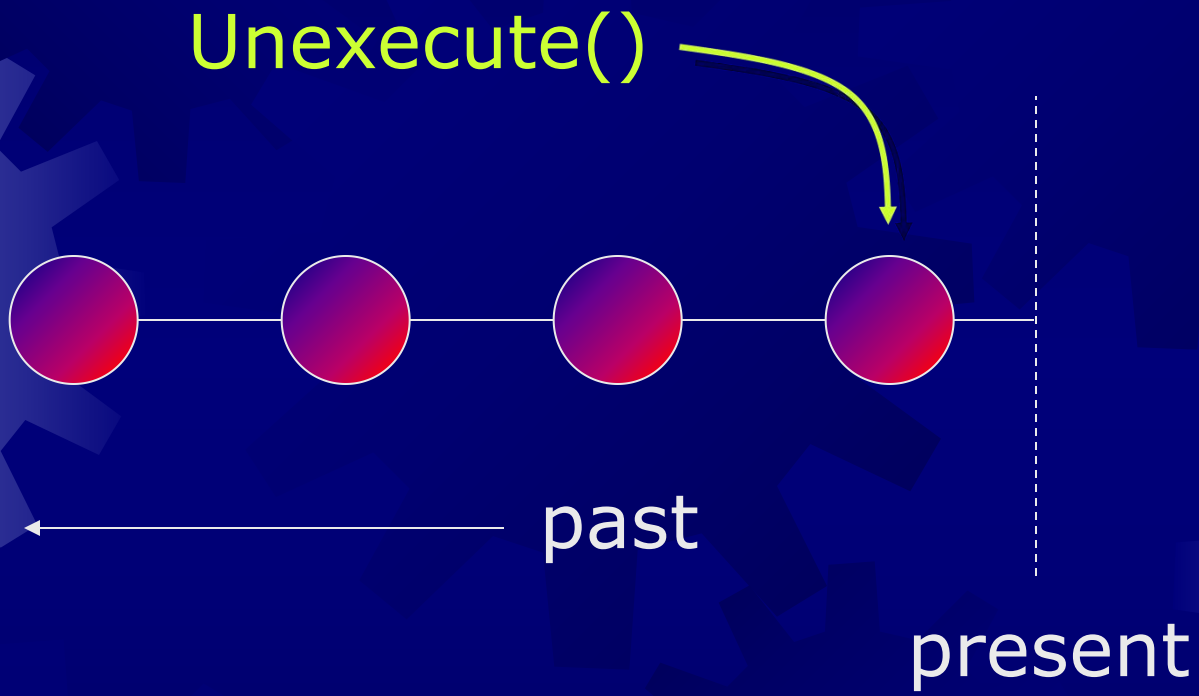


Operation details of the commands are recorded

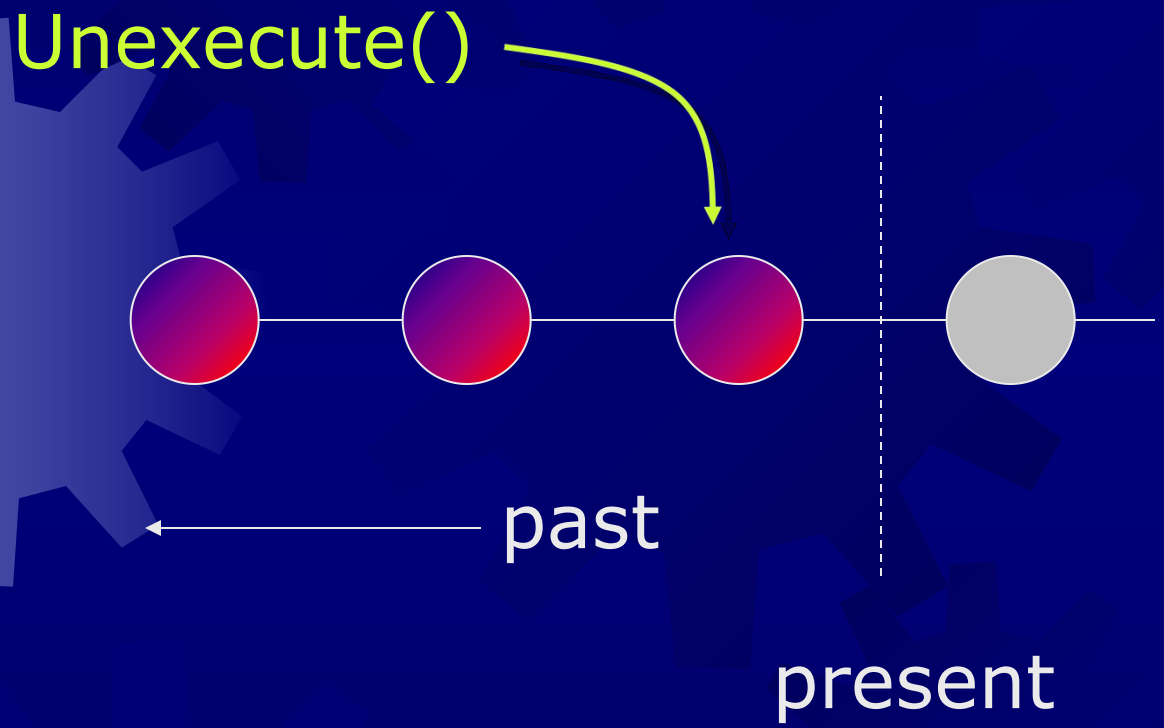
Undo

- Undo information must also be stored
  - E.g. changing font of the selected text
    range of the text
    original font(s) of the text
  - E.g. deleting the selected objects
    Information of all the objects should be stored!
- Undoability
  - Meaningless commands should not be undone
  - Whether a command is meaningless should be determined at run-time
  - Add Command::Reversible

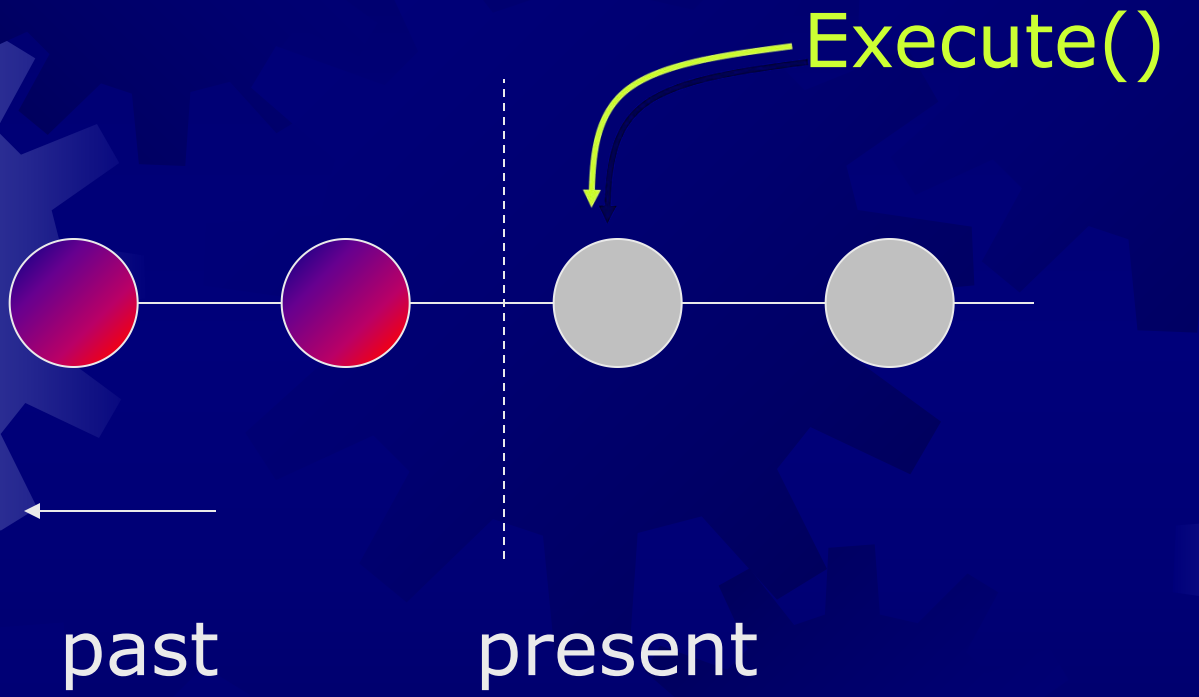
Command history



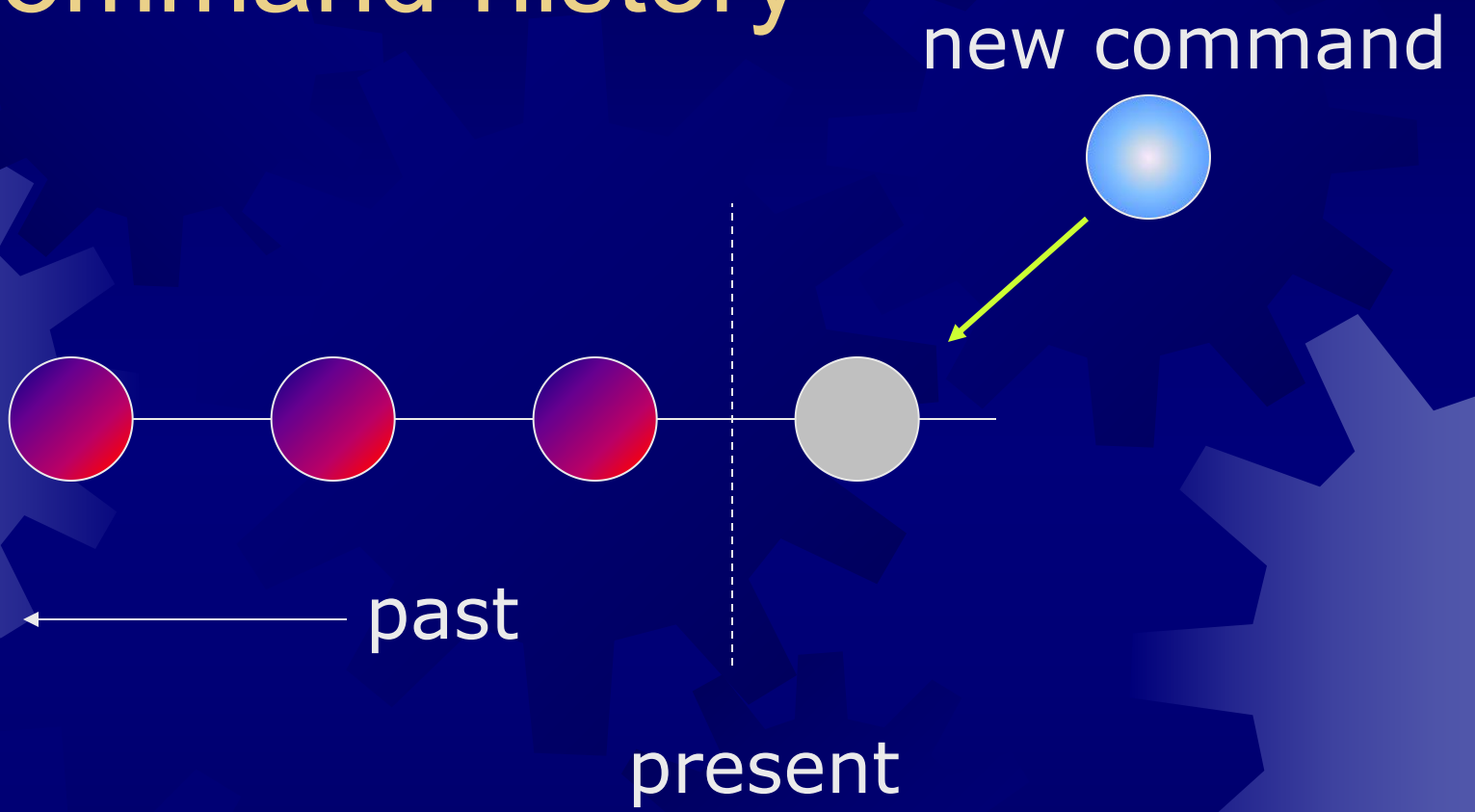
Command history



Command history



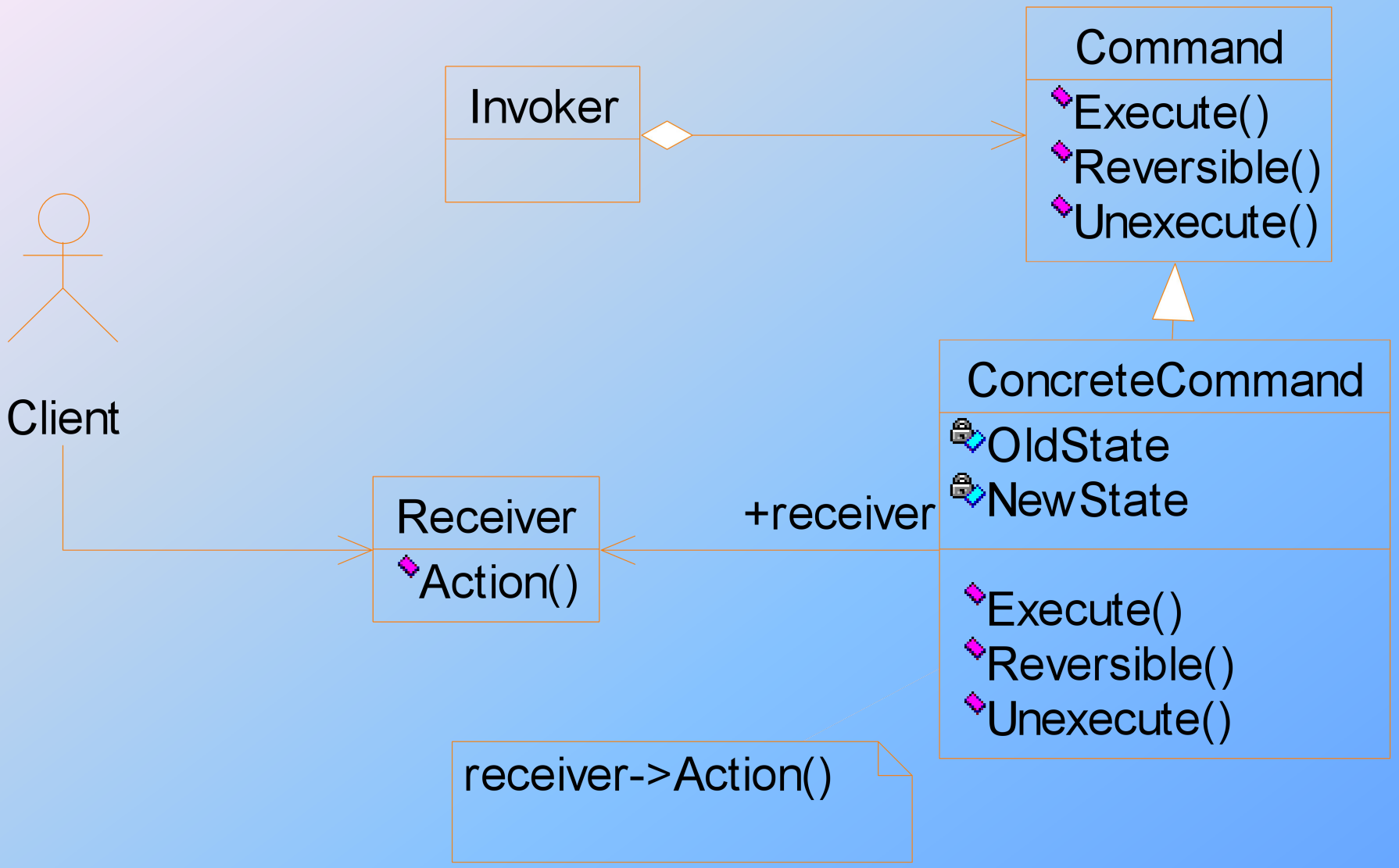
Command history



Command history

Applicability

- Support undo
- Support context-sensitive menus
- Support command macros
- Support logging changes to recover a crashed system
- Support the concept of *transaction* in an information system



Structure of the Command pattern