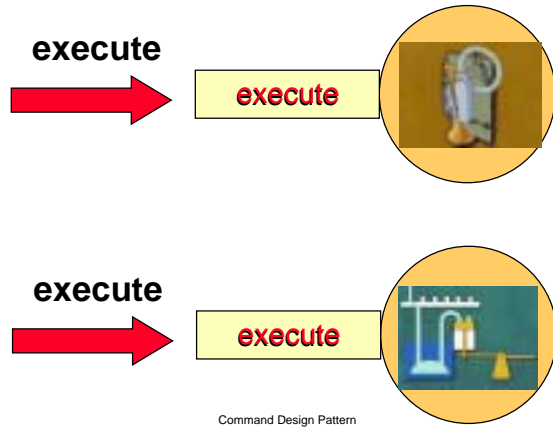


1: extending menus

Representing methods (operations) as objects

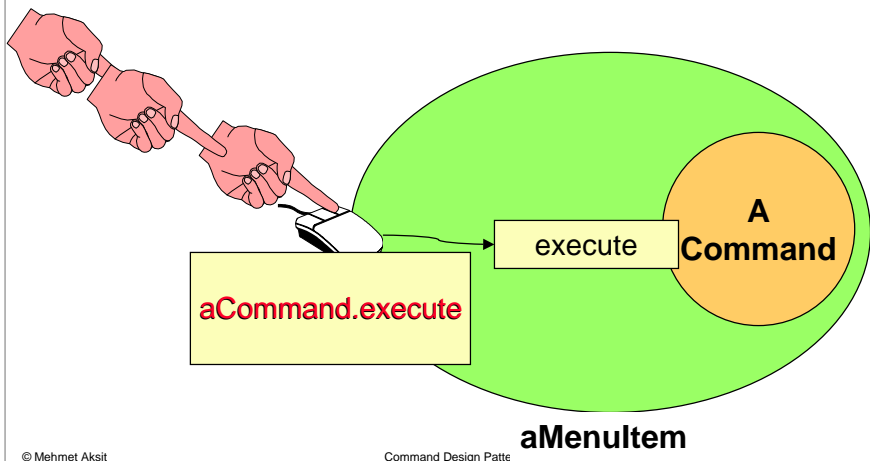


© Mehmet Aksit

Command Design Pattern

1: extending menus

Selecting a menu item

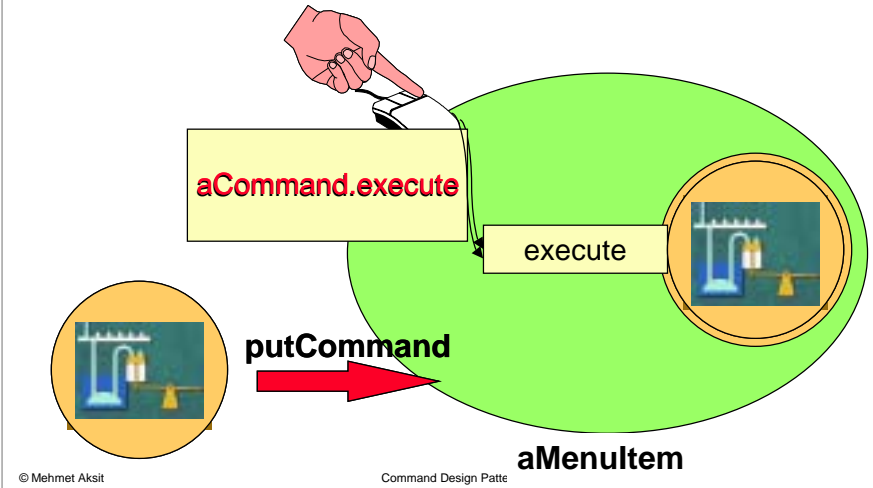


© Mehmet Aksit

Command Design Patte

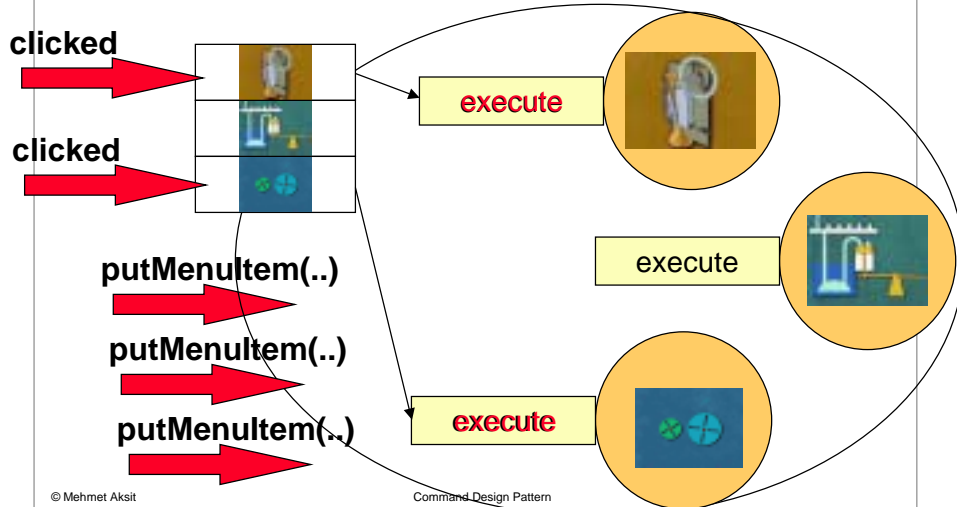
1: extending menus

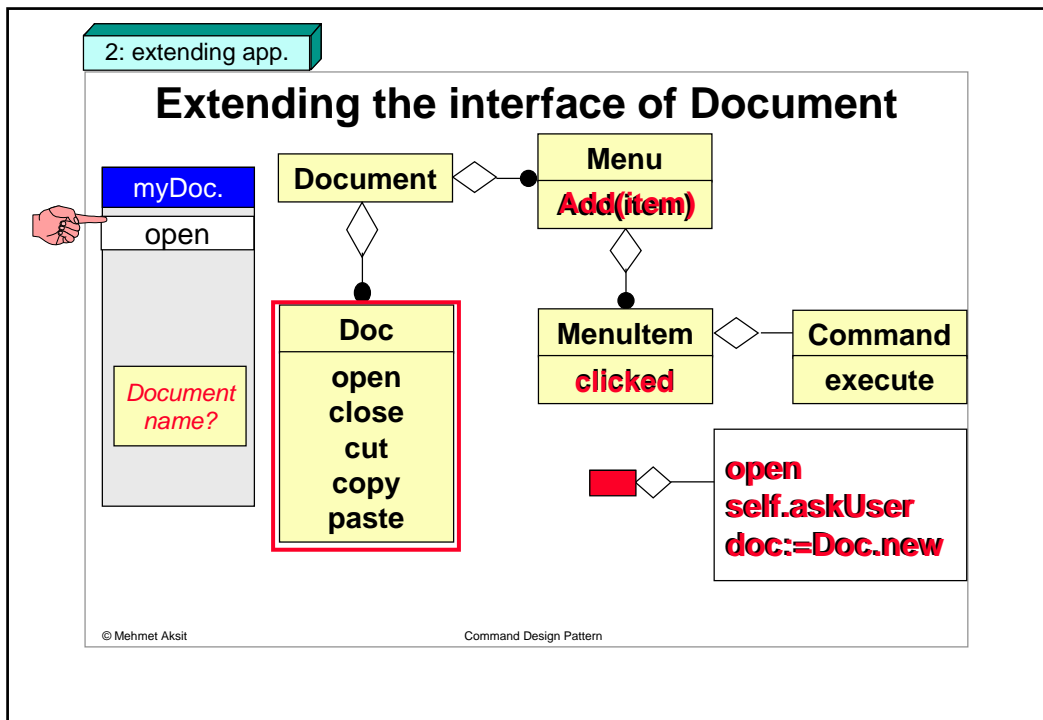
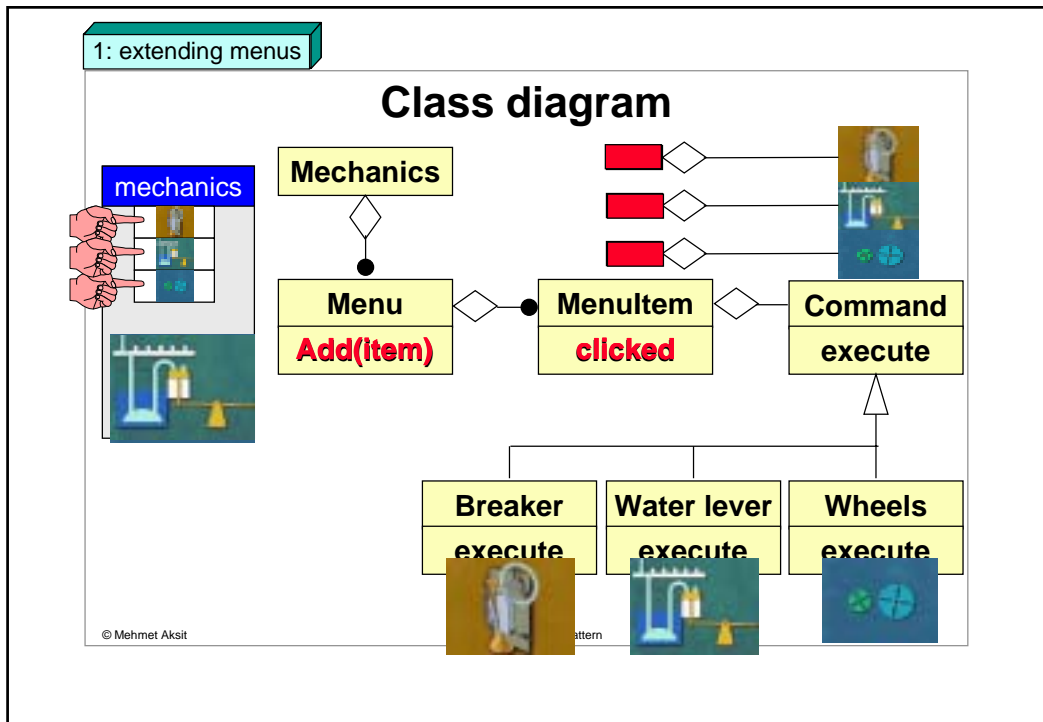
Uniform interface for every command

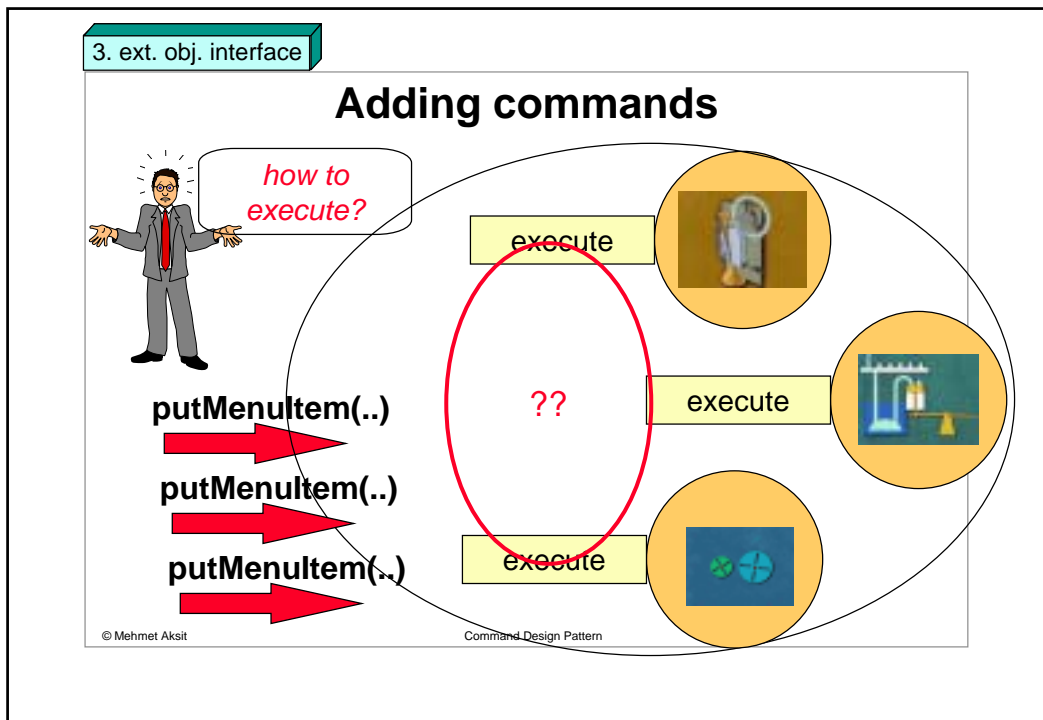
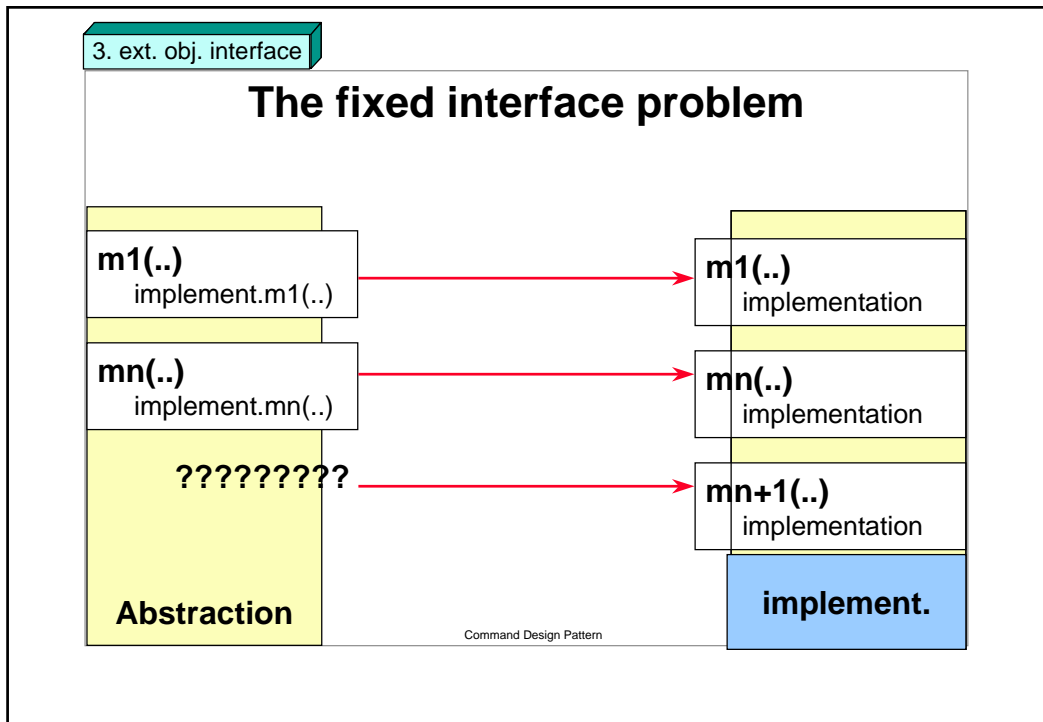


1: extending menus

Composing menu items

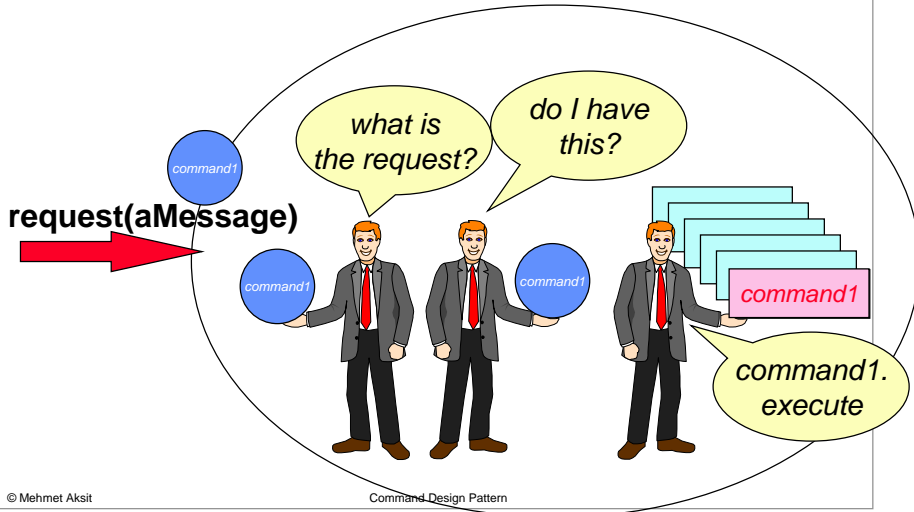






3. ext. obj. interface

Executing the command



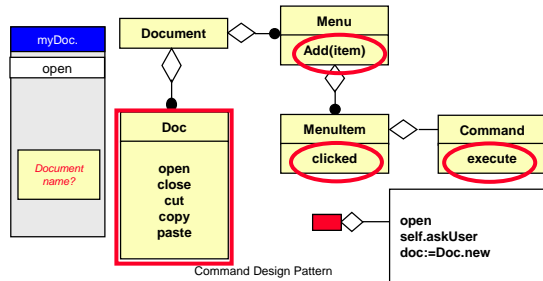
© Mehmet Aksit

Command Design Pattern

meta-model

Meta-model

reference	features		object-oriented techniques		
name	fixed	variable	coupling technique	changing technique	verification technique
command	clicked, execute, add, etc.				



© Mehmet Aksit

Command Design Pattern

meta-model

Meta-model (cont'ed)

reference	features		object-oriented techniques		
name	fixed	variable	coupling technique	changing technique	verification technique
command	clicked, execute, add, etc.	operations			

© Mehmet Aksit Command Design Pattern

meta-model

Meta-model (cont'ed)

reference	features		object-oriented techniques		
name	fixed	variable	coupling technique	changing technique	verification technique
command	clicked, execute, add, etc.	operations	polymorphic messages, encapsulation		

© Mehmet Aksit Command Design Pattern

meta-model

Meta-model (cont'ed)

reference	features		object-oriented techniques		
name	fixed	variable	coupling technique	changing technique	verification technique
command	clicked, execute, add, etc.	operations	polymorphic messages, encapsulation	composition	

© Mehmet Aksit
Command Design Pattern

meta-model

Meta-model (cont'ed)

reference	features		object-oriented techniques		
name	fixed	variable	coupling technique	changing technique	verification technique
command	clicked, execute, add, etc.	operations	polymorphic messages, encapsulation	composition	typing

© Mehmet Aksit

features

Features

- ☺ Decouples the invoker and executer objects;
- ☺ Commands are first class objects;
- ☺ Command can be composed;
- ☺ Easy to add new commands.

implementation

Implementation Issues and possible problems

- ☞ How intelligent should a command be?
- ☞ Supporting *undo* and *redo*: Additional state information may be needed: (1) from the executer (receiver) object (2) arguments of the operation and receiver must provide the necessary operations;
- ☞ Error accumulation must be avoided.