Software Design Techniques and Mechanisms

Topic: Behavioral Design Patterns

Presenter: Drumea Vasile

Chișinău 2021

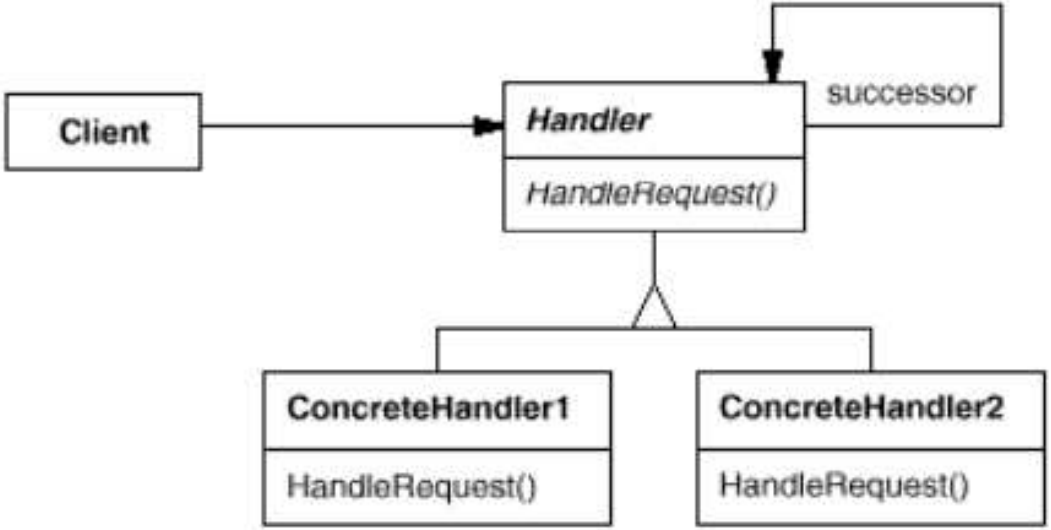
Overview

- The Behavioral Design Patterns are concerned with the communication between software entities and how it can be realized optimally.
- Based on the mechanism used there are 2 types:
  - Behavioral Class Patterns: Use inheritance to distribute behavior between classes.
  - Behavioral Object Patterns: Use object composition to have multiple objects perform common tasks.

Chain of Responsibility Pattern

- A chain of request handlers that links the sender to the receiver.
- The request can be handled partially by all the handlers, or by one specific handler from the chain.

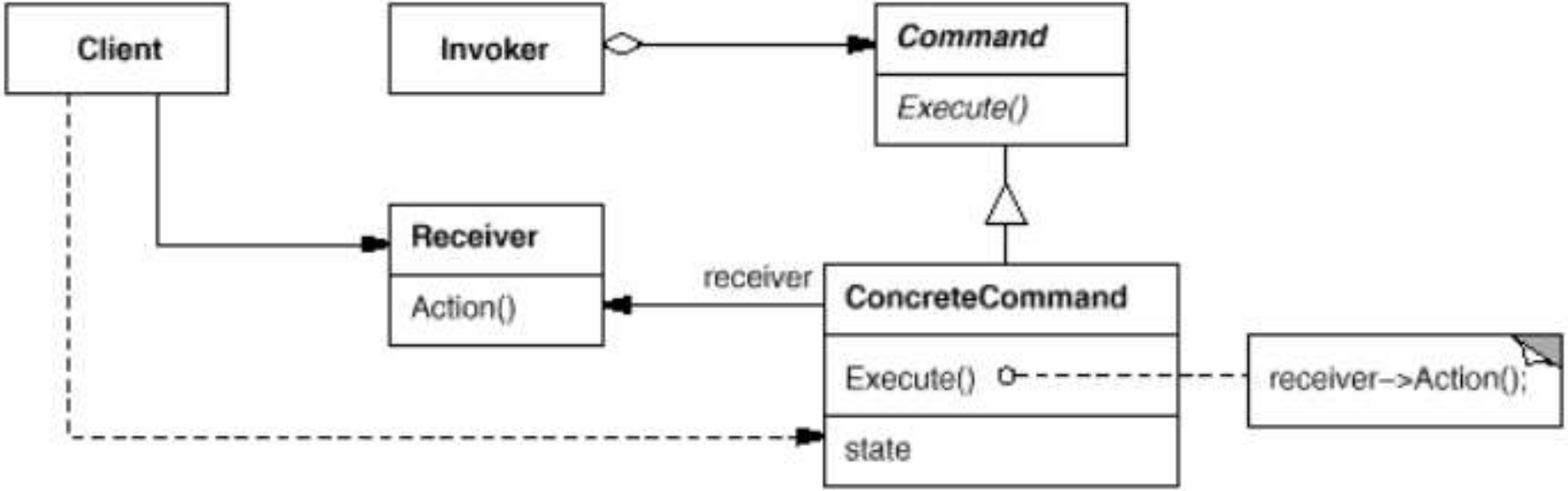
The UML Diagram for Chain of Responsibility



Command Pattern

- Encapsulate a request as an object/class so that it can be used whenever it is needed.
- The command object provides a blueprint which is a set of method calls for a specific command.
- Aka Action, Transaction.

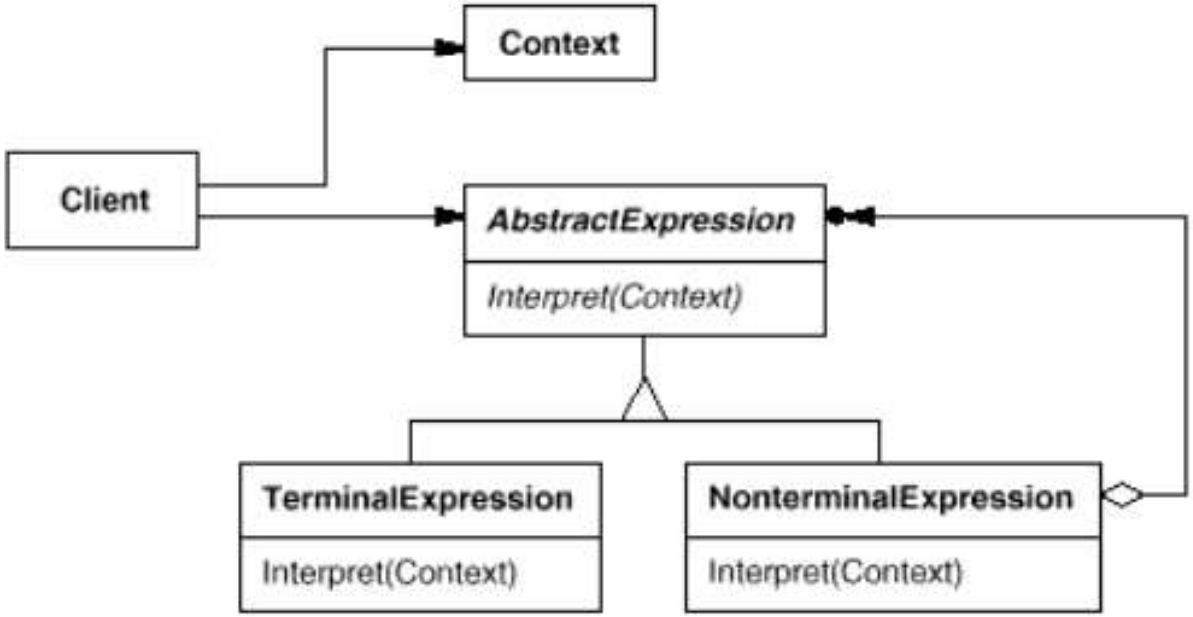
The UML Diagram for Command



Interpreter Pattern

- **Given a language, define a representation for its grammar along with an interpreter that uses the representation to interpret sentences in the language.**

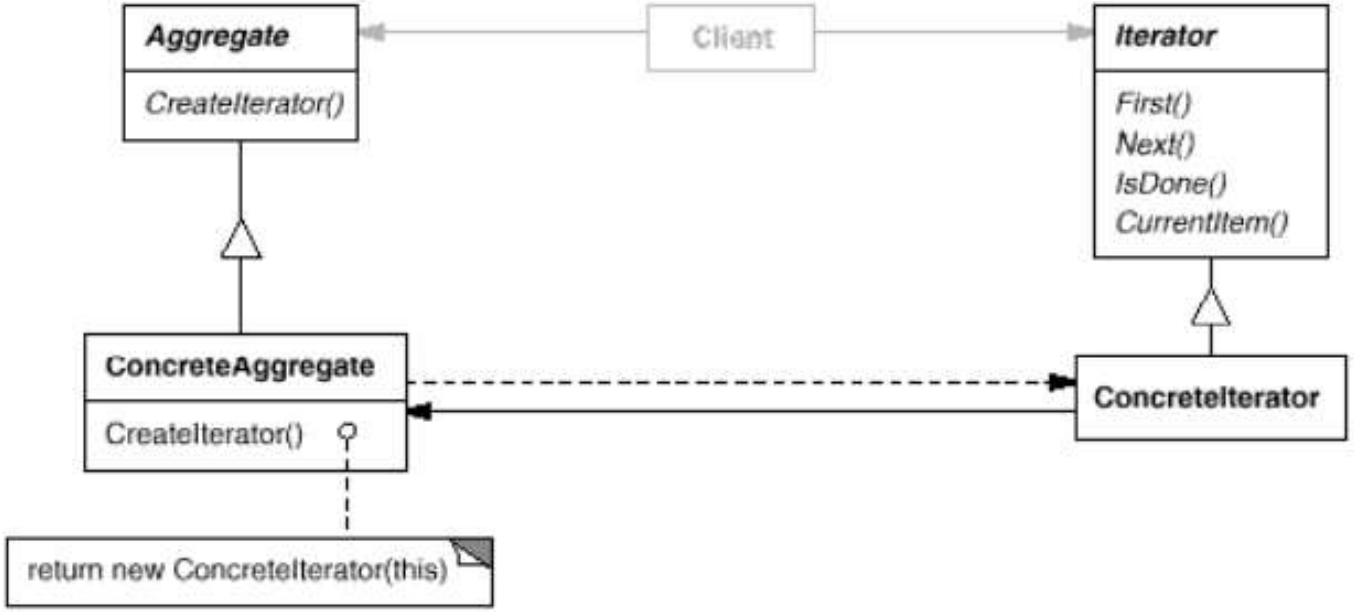
The UML Diagram for Interpreter



Iterator Pattern

- Provides a way to access sequentially the elements of an aggregate
  object without exposing its underlying representation.
- Aka Cursor.

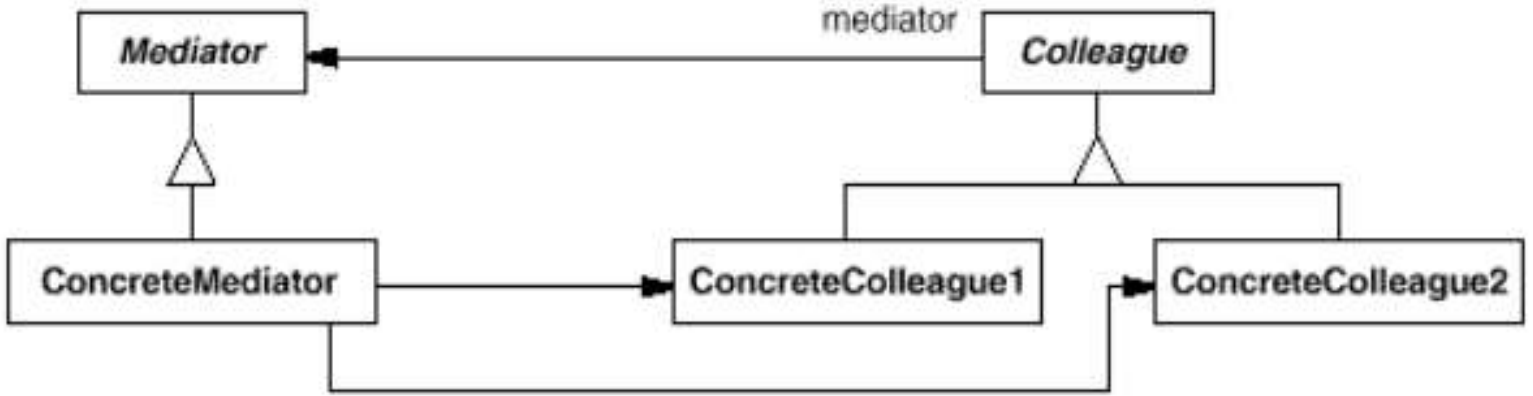
The UML Diagram for Iterator



Mediator Pattern

- Define an object that encapsulates how a set of objects interact.
- Mediator promotes loose coupling by keeping objects from referring to each other explicitly, and it lets you vary their interaction independently.

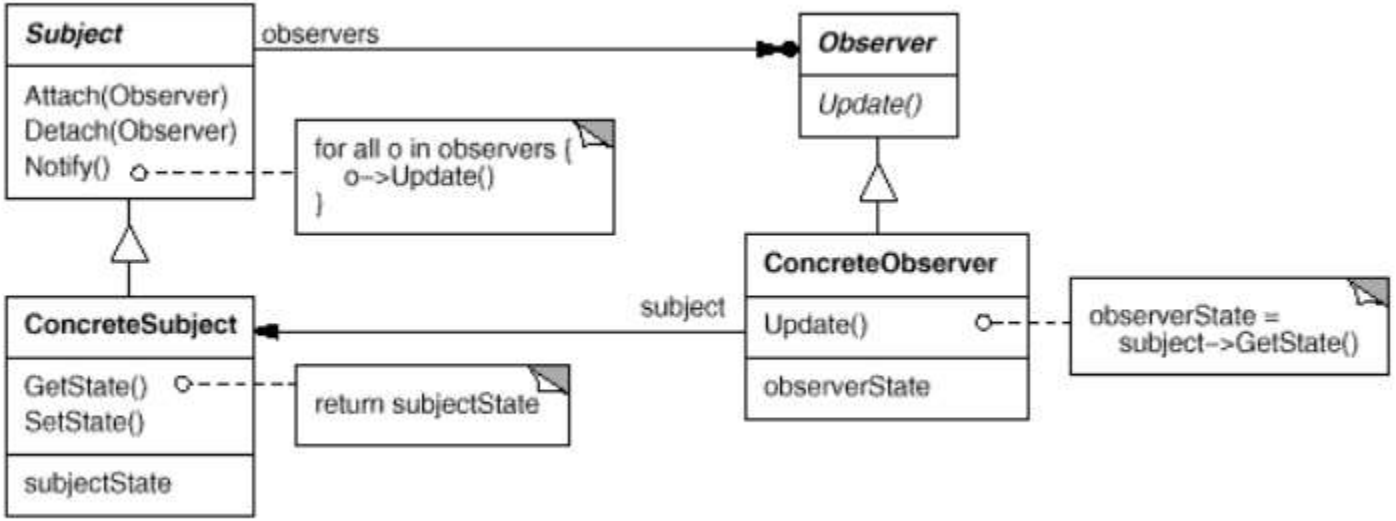
The UML Diagram for Mediator



Observer Pattern

- Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.
- Aka Dependents, Publish-Subscribe.

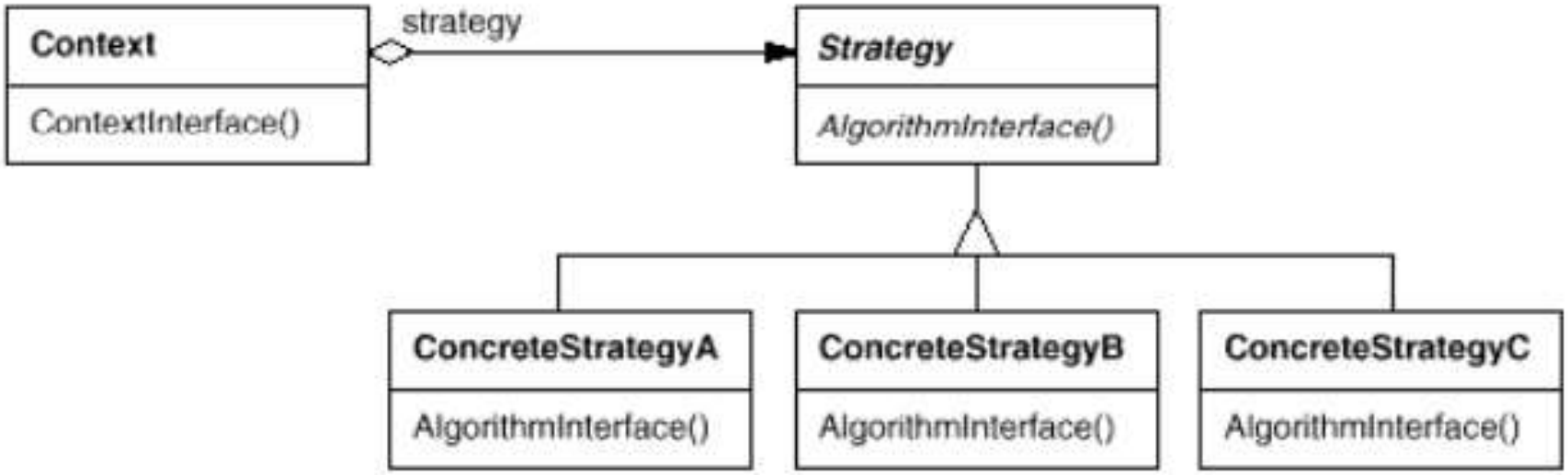
The UML Diagram for Observer



Strategy Pattern

- Defines a family of algorithms, encapsulates each one and makes them interchangeable.
- Aka Policy.

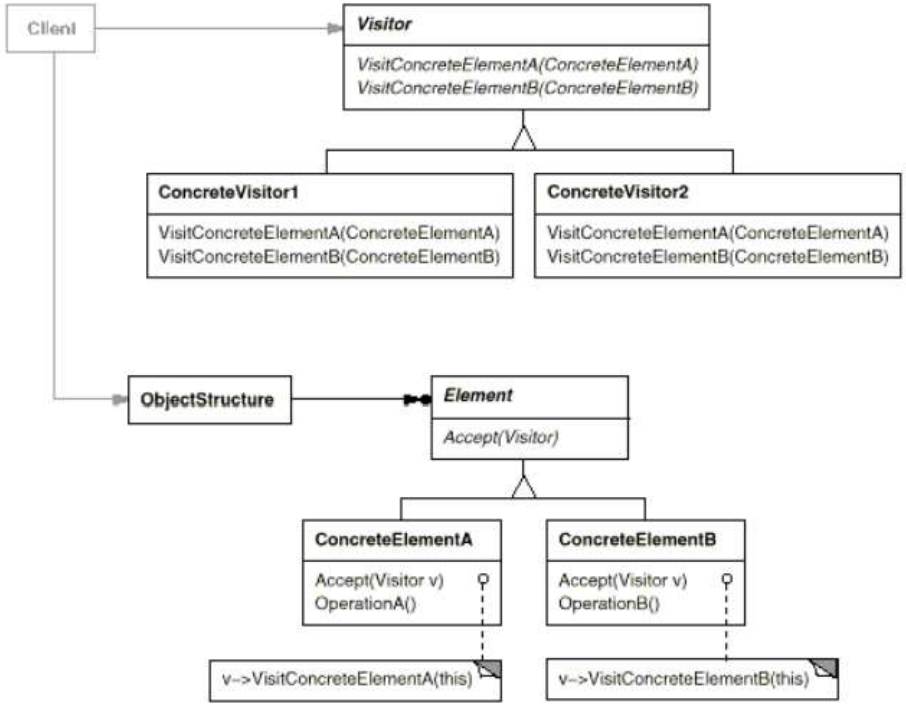
The UML Diagram for Strategy



Visitor Pattern

- **(Common definition)** Represent an operation to be performed on the elements of an object structure. Visitor lets you define a new operation without changing the classes of the elements on which it operates.
- Defines an operation to be performed on the elements of a common object structure (inheritance hierarchy/composite object).
- The operation is declared in the Visitor class and it will provide (if necessary) multiple implementations for different types of object.

The UML Diagram for Visitor



References

1. https://sourcemaking.com/design_patterns/behavioral_patterns
2. The “Gang of four”, 1994, *Design Patterns: Elements of Reusable Object-Oriented Software*
3. P.S. All the diagrams are from [2].

Thanks for your attention!
Questions?