What are Design Patterns?

Definitions and Template

AARHUS UNIVERSITET

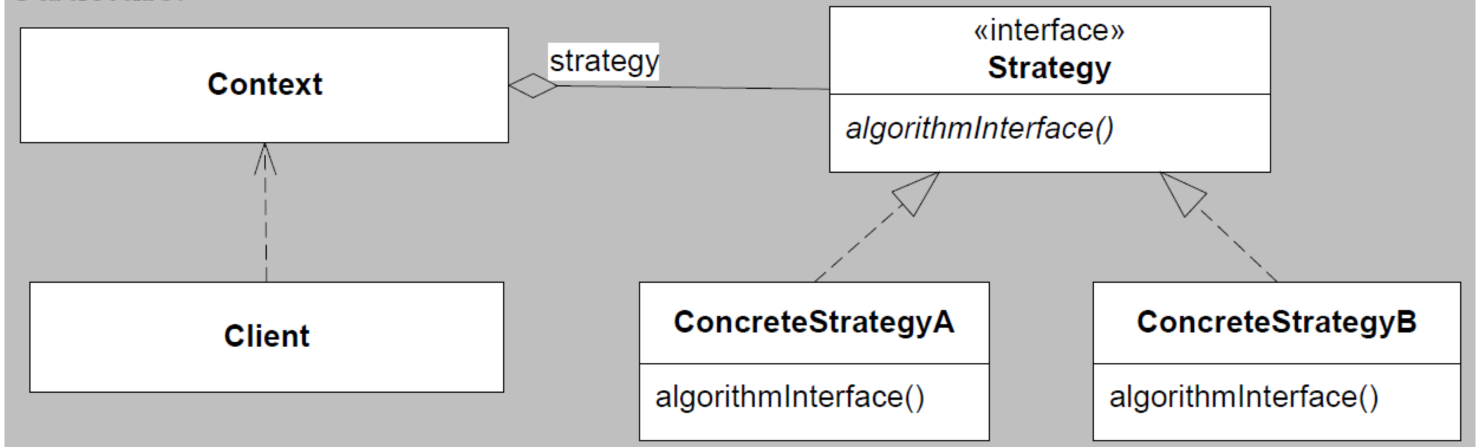
Gamma et al.'s definition

Definition: Design Pattern (Gamma et al.)

Patterns are descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context.

Exercise: How does it relate to…

Structure:



Beck et al.'s Definition

Definition: Design Pattern (Beck et al.)

A design pattern is a particular prose form of recording design information such that designs which have worked well in the past can be applied again in similar situations in the future.

Prose form = "writing template"

The template varies from author to author.

However, must contain

- Name
- Problem
- Solution
- Consequences

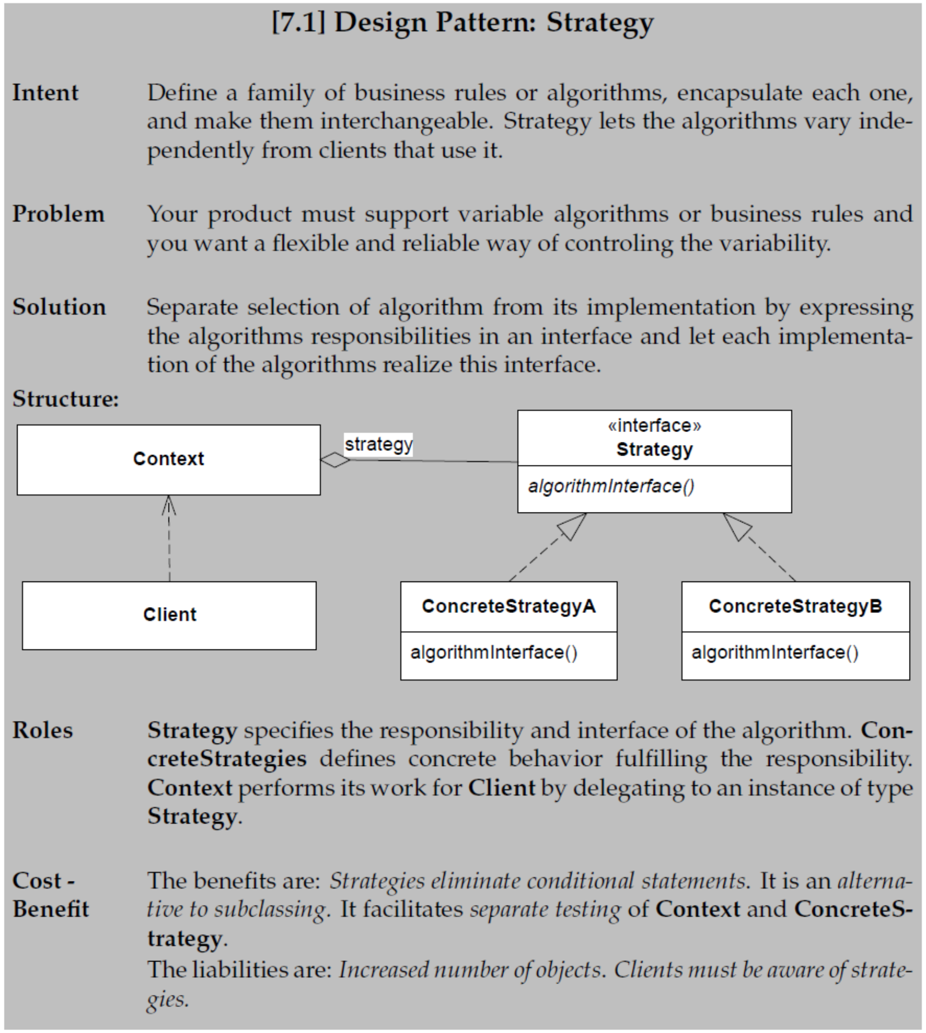
FRSE's template

Intent

- Short description

Roles

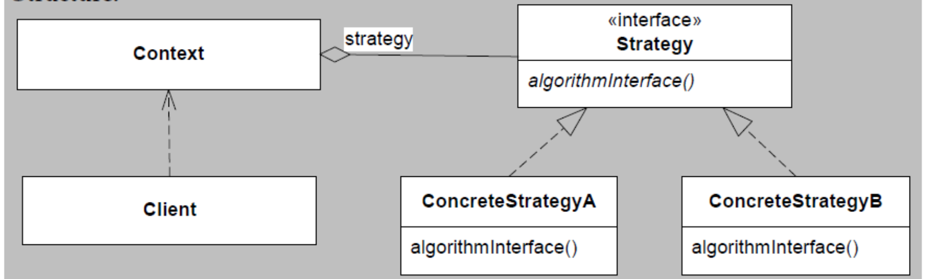
- Responsibilities of each participating object/abstraction in the pattern



[7.1] Design Pattern: Strategy

Intent	Define a family of business rules or algorithms, encapsulate each one, and make them interchangeable. Strategy lets the algorithms vary independently from clients that use it.
Problem	Your product must support variable algorithms or business rules and you want a flexible and reliable way of controling the variability.
Solution	Separate selection of algorithm from its implementation by expressing the algorithms responsibilities in an interface and let each implementation of the algorithms realize this interface.

Structure:



Roles	**Strategy** specifies the responsibility and interface of the algorithm. **ConcreteStrategies** defines concrete behavior fulfilling the responsibility. **Context** performs its work for **Client** by delegating to an instance of type **Strategy**.
Cost - Benefit	The benefits are: *Strategies eliminate conditional statements*. It is an *alternative to subclassing*. It facilitates *separate testing* of **Context** and **ConcreteStrategy**. The liabilities are: *Increased number of objects. Clients must be aware of strategies.*

Differentiating Patterns

Be aware that many patterns are *structurally equal* – their UML class diagrams are more or less identical!

Patterns are defined by the *problem they solve!*

Strategy is the problem of

Handling variability of algorithms / business rules, making them interchangeable.