Lesson: Proxy Design Pattern in Software Engineering

Overview

In this lesson, we will explore the Proxy design pattern, its history, definition, and types. We will also discuss the usage of the Proxy pattern in software engineering.

Team Syntax Error

Before we delve into the Proxy pattern, let's acknowledge the team behind this lesson:

- Mikhaella Jing Carumbana
- Jomar Magdalaga
- Karl John Villardar

History of the Proxy Pattern

The Proxy pattern has its origins in the renowned book "Design Patterns: Elements of Reusable Object-Oriented Software," often referred to as the "Gang of Four" (GoF) book. It was written by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, collectively known as the Gang of Four. The book was first published in 1994 and played a significant role in introducing and popularizing various design patterns, including the Proxy pattern.

Definition of Proxy Pattern

The term "proxy" signifies acting on behalf of something or someone else, typically serving as a representative or stand-in. This concept directly applies to the Proxy Design Pattern. Proxies can also be referred to as surrogates, handles, or wrappers. While they share similarities in structure with Adapters and Decorators, their purpose sets them apart.

Types of Proxies

1. **Remote Proxy:** A remote proxy provides a local representation of an object that resides in a different address space or remote location. It acts as a communication medium between the client and the actual object, handling the underlying complexity.
2. **Virtual Proxy:** A virtual proxy creates a placeholder for an expensive or resource-intensive object. It delays the creation of the actual object until it is explicitly requested, thus providing a lazy loading mechanism.
3. **Protection Proxy:** A protection proxy controls access to an object by adding an extra layer of security. It validates the client's permissions and ensures that only authorized clients can access the real object.
4. **Smart Proxy:** A smart proxy performs additional tasks when accessing the real object. It can be used to add functionality such as caching, logging, or error handling.

Usage of the Proxy Pattern

The Proxy pattern is employed when there is a need to create a wrapper that conceals the complexity of the main object from the client. It acts as a middleman, providing a simplified interface and managing the interactions between the client and the actual object.By using the Proxy pattern, developers can achieve various benefits, such as:

- Improved performance through lazy loading and caching.
- Enhanced security by controlling access to sensitive objects.
- Simplified codebase by separating the client from the complex implementation details.
- Flexibility to add additional functionality without modifying the real object.

Conclusion

In this lesson, we have explored the Proxy design pattern, its history, definition, and various types. We have also discussed the usage of the Proxy pattern in software engineering. Understanding and utilizing this pattern can greatly contribute to the development of robust and efficient software systems.References:

- "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides (the Gang of Four book)