Tarea 3
Patrones de Diseño
Ayudantía de Ingeniería de Software

Axel Rodríguez Espinoa
Nicolas Oyarzún Hernandez
Pedro Salas Vergara

11 de diciembre de 2013

PATRONES DE DISEÑO DE SOFTWARE

Introducción

El diseño es un modelo del sistema, realizado con una serie de principios y técnicas, que permite describir el sistema con el suficiente detalle como para ser implementado. Pero los principios y reglas no son suficientes, en el contexto de diseño podemos observar que los buenos ingenieros tienen esquemas y estructuras de solución que usan numerosas veces en función del contexto del problema. Este es el sentido cabal de la expresión "tener un mente bien amueblada", y no el significado de tener una notable inteligencia. Estos esquemas y estructuras son conceptos reusables y nos permiten no reinventar la rueda. Un buen ingeniero reutiliza un esquema de solución ante problemas similares.

Antecedentes

El concepto de "patrón de diseño"que tenemos en Ingeniería del Software se ha tomado prestado de la arquitectura. En 1977 se publica el libro .ª Pattern Language: Towns/Building/Construction". Contiene numerosos patrones con una notación específica de Christopher Alexander.

Alexander comenta que "Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma". El patrón es un esquema de solución que se aplica a un tipo de problema, esta aplicación del patrón no es mecánica, sino que requiere de adaptación y matices. Por ello, dice Alexander que los numerosos usos de un patrón no se repiten dos veces de la misma forma.

La idea de patrones de diseño estaba .en el aire", la prueba es que numerosos diseñadores se dirigieron a aplicar las ideas de Alexander a su contexto. El catálogo más famoso de patrones se encuentra en "Design Patterns: Elements of Reusable Object-Oriented Software", de Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides, 1995, Addison-Wesley, también conocido como el libro GOF (Gang-Of-Four).

Siguiendo el libro de GOF los patrones se clasifican según el proposito para el que han sido definidos:

- **Creacionales**: solucionan problemas de creación de instancias. Nos ayudan a encapsular y abstraer dicha creación.

- **Estructurales**: solucionan problemas de composición (agregación) de clases y objetos.

- **De Comportamiento**: soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan.

Concepto de patron de diseño

-Una arquitectura orientada a objetos bien estructurada está llena de patrones. La calidad de un sistema orientado a objetos se mide por la atención que los diseñadores han prestado a las colaboraciones entre sus objetos. Los patrones conducen a arquitecturas más pequeñas, más simples y más comprensibles.- (Grady Booch).
Los patrones de diseño son descripciones de clases cuyas instancias colaboran entre sí. Cada patrón es adecuado para ser adaptado a un cierto tipo de problema. Para describir un caso debemos especificar:

- Nombre
- Propósito o finalidad
- Sinónimos (otros nombres por los que puede ser conocido)
- Problema al que es aplicable
- Estructura (diagrama de clases)
- Participantes (responsabilidad de cada clase)
- Colaboraciones (diagrama de interacciones)
- Colaboraciones (diagrama de interacciones)
- Otros patrones con los que está relacionado

Ventajas de los patrones de diseño

La clave para la reutilización es anticiparse a los nuevos requisitos y cambios, de modo que los sistemas evolucionen de forma adecuada. Cada patrón permite que algunos aspectos de la estructura del sistema puedan cambiar independientemente de otros aspectos. Facilitan la reusabilidad, extensibilidad y mantenimiento.

Patrón Mediador (Mediator)

El mediador o intermediario es un patrón de comportamiento que encapsula la forma en que interaccionan un conjunto de objetos (çolegas"). Es el especialista que define las interdependencias entre ellos. Favorece un bajo acoplamiento, ya que evita que los objetos se referencien unos a otros de forma explícita. Permite variar la interacción sin tocar los colegas, por tanto favorece la reutilización.

Cuando muchos objetos interactúan con otros objetos, se puede formar una estructura muy compleja, con objetos con muchas conexiones con otros objetos.

En un caso extremo cada objeto puede conocer a todos los demás objetos.

Participantes:

Mediator: Define una interface para comunicarse con los objetos colegas.

ConcreteMediator: Implementa el comportamiento cooperativo entre los colegas (como se comunican entre ellos). Además los conoce y mantiene.

Colleagues: Cada colega conoce su mediador, y usa a este para comunicarse con otros colegas.

Ventajas:

- Evita crear subclases
- Desacopla a los colegas
- Simplifica los protocolos entre las clases
- Abstrae el cómo cooperan los objetos
- Centraliza el control en el mediador: clase difícil de mantener

Ejemplo de Mediador (Mediator)

En el caso de la comunicación que tienen los militantes en un partido político. Todas estas organizaciones tienen distintos delegados de territorio o de sectores determinados (p.ej. en la comuna de Ñuñoa, en un espacio educacional como CONFECH, sindicatos, entre otros) que deben informar a todo el partido la actualidad con respecto al contexto en su área y a su vez deben estar informados en torno a las labores que se realizan dentro de la organización.

Existe el comando del partido, que se encarga de recepcionar la información que reportan los militantes de los diversos territorios, sectores, etc. Luego de tenerla en su poder la transmite a las demás, con el objetivo de mantener informados a todos los militantes del partido.

En este caso los participantes son:

Mediator: El comando del partido.
ConcreteMediator: Mecanismo de recepción por parte del comando de información de los militantes de un área y de difusión de la información a las demás áreas.
Colleagues: Militantes del partido de un territorio o sector determinado.

Esta forma permite al partido tener un orden en cuanto a la información que se transmite dentro de este y como finalidad del patrón evita que todos los militantes interactúen con todos.

Anexo

Repositorio del documento en LaTeX:
`https://github.com/AxelRodriguezE/Tarea3_Ayudantia_IngSoftware`