

Patrones de Diseño

Linda María Fernández Olvera

UPIICSA-IPN

Ingeniería de Diseño

lfernandezol1500@alumno.ipn.mx

Abstract – Este ensayo proveerá información sobre los patrones de diseño, su clasificación, características, el lenguaje y dónde se pueden encontrar. El contexto de los patrones de diseño en este ensayo se centra orientado a objetos.

Índice de Términos – patrones, estructuras, conductuales, relaciones, orientado a objetos, creacionales.

I. INTRODUCCION

Los patrones de diseño se caracterizan por ser una relación entre cierto contexto, un problema y una solución, dependiendo del primero se puede encontrar la solución óptima para el problema. Primeramente, se describirá que es un patrón de diseño, sus características para que sus soluciones sean óptimas, la clase de patrones y tipos para situaciones específicas. Posteriormente se hablará sobre la diferencia entre los patrones y las estructuras. Al final se hablará sobre la semántica de los patrones de diseño y donde se almacenan para que se empleen posteriormente.

II. DESARROLLO

El contexto permite entender el ambiente donde se desarrolla el problema y que solución puede ser apropiada para el ambiente.

El conjunto de requerimientos, que incluye limitaciones y restricciones, actúan como sistema de fuerzas que afectan la manera en la que se puede interpretar el problema del contexto y la forma en que se podría hacer más eficiente la solución, si el sistema de fuerzas cambia también lo puede hacer la solución.

Gran parte de los problemas tiene una solución, sin embargo, solo una es suficientemente eficaz la que resulta apropiada para el contexto del problema existente. También es importante mencionar que toda solución tiene consecuencias que afectan otros aspectos del software que se desarrolla y se vuelven parte del sistema de fuerzas de otros problemas del sistema mayor.

Las características de un patrón de diseño eficaz son:

- Resolución de un problema: Los patrones son

soluciones, no solo una estrategia abstracta.

- Concepto probado: Los patrones se han usado y están probados, sus soluciones tienen un historial y no son especulaciones.
- La solución no es obvia: los patrones generan indirectamente las soluciones de los problemas, no las obtienen de principios generales.
- Describen una relación: los patrones describen módulos, estructuras y mecanismos del sistema.
- Minimiza la intervención humana: el software busca mejorar el confort humano, los patrones recurren a la estética y la utilidad para desarrollar soluciones.

El conjunto de requerimientos, que incluye limitaciones y restricciones, actúan como sistema de fuerzas que afectan la manera en la que se puede interpretar el problema del contexto y la forma en que se podría hacer más eficiente la solución, si el sistema de fuerzas cambia también lo puede hacer la solución.

Un buen patrón de diseño incorpora el conocimiento del diseño pragmático, que permite que otros lo reutilicen la solución sin tener que elaborar otra de la misma forma. Los patrones hacen del desarrollador un buen diseñador de software.

A. Clases de patrones

El ser humano tiene la capacidad de reconocer patrones, de esta manera aprendemos de nuestras experiencias. En el mundo real, los patrones se reconocen con el tiempo, ya conocidos los identificamos al instante y comprendemos lo que significan y como podemos emplearlos, incluso nos permiten detectar fenómenos recurrentes.

Los patrones no generativos tienen un contexto y un problema, pero no ofrecen una solución clara.

Los patrones generativos describen un aspecto importante y repetitivo del sistema, construyen el contexto con el sistema de fuerzas únicas. La generatividad es la aplicación sucesiva de varios patrones, cada uno incluye su propio problema y fuerzas, juntos despliegan una solución completa que emerge indirectamente como resultado de soluciones pequeñas.

Los patrones arquitectónicos describen problemas de diseño de base amplia que se resuelven empleando un enfoque

estructural. Los patrones de datos describen problemas recurrentes orientados a datos y las soluciones de modelado de datos que pueden emplearse para resolverlos. Los patrones de componentes (también llamados patrones de diseño) se enfocan a problemas asociados con el desarrollo de subsistemas y componentes, así como a la manera en la que se comunican entre sí y su ubicación dentro de una arquitectura mayor. Los patrones de diseño de la interfaz describen problemas comunes de interfaz de usuario y su solución con un sistema de fuerzas que incluye las características específicas de los usuarios finales.

Para los patrones de diseño nos centraremos en patrones de enfocados en el diseño orientado a objetos, los patrones creacionales, estructurales y conductuales.

B. Patrones creacionales

Los patrones creacionales se centran en la creación, composición y representación de objetos. Encierran conocimiento acerca de las clases concretas del sistema, pero ocultan la forma en la que las instancias de las clases se crean y agrupan. Estos patrones ofrecen mecanismos que forman fácilmente las instancias de los objetos dentro de un sistema y establecen restricciones en la cantidad de objetos que se pueden crear dentro de un sistema.

- Patrón de fábrica abstracta: centraliza la decisión de qué fábrica deben hacerse las instancias.
- Patrón de método de fabricación: su objetivo es la creación de un objeto de tipo específico para elegir una entre varias implementaciones.
- Patrón constructor: separa la construcción de un objeto complejo a partir de su representación, de modo que el mismo proceso de construcción pueda crear representaciones distintas.
- Patrón prototipo: se usa cuando el costo inherente que implica crear un nuevo objeto en la forma estándar es prohibitivo para una aplicación dada.
- Patrón de instancia única: restringe la formación de instancias de una clase a un objeto.

C. Patrones estructurales

Los patrones estructurales se centran en problemas y soluciones asociados con la forma en la que se organizan e integran las clases y objetos para construir una estructura más grande, establecen relaciones entre entidades de un sistema, algunos patrones sugieran técnicas para combinar los objetos dentro de otros o integrarlos en estructuras más grandes.

- Patrón adaptador: adapta una interfaz para una clase en otra que espera un cliente.

- Patrón agregado: es una versión del patrón compuesto con métodos para agregar hijos.

- Patrón de puente: desacopla una abstracción de su implementación, de modo que las dos puedan variar en forma independiente.

- Patrón compuesto: estructura de árbol de objetos en los que cada uno tiene la misma interfaz.

- Patrón contenedor: crea objetos con el único propósito de que sostengan a otros y los administren.

- Patrón de proximidad: clase que funciona como interfaz respecto de otra cosa.

- Tubos y filtros: cadena de procesos en los que la salida de cada uno es la entrada del siguiente.

D. Patrones conductuales

Los patrones conductuales se enfocan a problemas asociados con la asignación de responsabilidad entre los objetos y a la manera en la que se efectúa la comunicación entre ellos.

- Cadena de patrones de responsabilidad: objetos de comando que son manejados o pasados a otros objetos por medio de otros que contienen procesamiento lógico.
- Patrón de comando: objetos de comando que encierran una acción y sus parámetros.
- Escucha de eventos: se distribuyen datos a objetos registrados para recibirlos.
- Patrón intérprete: implementa un lenguaje de computadora especializado para resolver con rapidez un conjunto específico de problemas.
- Patrón iterador: los iteradores se utilizan para acceder en forma secuencial a los elementos de un agregado sin exponer su representación subyacente.

- Patrón mediador: proporciona una interfaz unificada a un conjunto de interfaces en un subsistema.

- Patrón visitante: forma de separar un algoritmo de un objeto.

- Patrón visitante de un solo servicio: optimiza la implementación de un visitante que se haya asignado, utilizado sólo una vez y luego eliminado.

- Patrón visitante jerárquico: brinda una forma de visitar cada nodo en una estructura jerárquica de datos, como un árbol.

E. Estructuras

Algunos casos necesitan dar un esqueleto de infraestructura específica para la implementación, la cual se conoce como estructura, son arquitecturas reutilizables que dan estructura y comportamiento a las abstracciones, así como un contexto.

Las estructuras no son patrones arquitectónicos, sino solo puntos de conexión que permiten adaptar la infraestructura, en el contexto de objetos se entiende como un conjunto de clases que colaboran. Para ser más eficaces se aplican sin cambios.

Los patrones de diseño son más abstractos que las estructuras, ya que son elementos arquitectónicos más pequeños, por lo tanto, la estructura puede incluir varios patrones.

F. Descripción de un patrón

El diseño comienza con el reconocimiento de patrones en el proyecto, continua con la búsqueda de un historial del patrón y al final se concluye con la aplicación de un patrón apropiado.

Formato de diseño del patrón

Nombre del patrón: describe la esencia del patrón con un nombre corto pero expresivo

Problema: describe el problema al que se dirige el patrón

Motivación: proporciona un ejemplo del problema

Contexto: describe el ambiente en el que reside el problema, incluido el dominio de aplicación

Fuerzas: lista el sistema de fuerzas que afectan la manera en la que debe resolverse el problema; incluye el análisis de las limitaciones y restricciones que deben ser tomadas en cuenta

Solución: hace la descripción detallada de la solución propuesta para el problema

Objetivo: describe el patrón y lo que hace

Colaboraciones: describe la manera en la que otros patrones contribuyen a la solución

Consecuencias: describe los intercambios potenciales que deben de ser considerados cuando se implementa el patrón, y las consecuencias de usar éste

Implementación: identifica los aspectos especiales que deben considerarse cuando se implemente el patrón

Usos conocidos: da ejemplos de usos reales del patrón de diseño en aplicaciones reales

Patrones relacionados: menciona referencias de patrones de diseño relacionados

El formato del patrón es un medio estandarizado para describir un patrón de diseño.

G. Lenguajes y repositorios de patrones

El lenguaje tiene una sintaxis y semántica que se utiliza para comunicar instrucciones de un procedimiento.

El contexto de diseño de patrones, el lenguaje de patrón

agrupa un conjunto de patrones, cada uno describe su uso de un formato estandarizado, e interrelacionado para mostrar cómo colaboran los patrones para resolver los problemas.

En un lenguaje de patrón, los patrones de diseño están organizados en forma tal que proporcionan un “método estructurado para describir las buenas prácticas de diseño en un dominio particular.

El dominio del problema en cuestión primero se describe de manera jerárquica, comenzando con problemas de diseño amplio asociados con el dominio, y luego se refina cada uno de ellos en niveles de abstracción más bajos.

En la mayor parte de casos, los patrones de diseño que forman parte de un lenguaje de patrones se almacenan en un repositorio de patrones accesible a través de la web.

III. CONCLUSIÓN

Los patrones de diseño se componen de un contexto, un problema y una solución. Un diseño eficaz emplea el contexto y su sistema de fuerzas para generar las soluciones. En el contexto orientado a objetos se emplean patrones creacionales que se centran en la creación y representación de objetos y clases, los estructurales en su arquitectura y los conductuales en su comportamiento. Los requerimientos nos proporcionan toda la información sobre el contexto.

REFERENCES

- [1] Roger S. Pressman, “*Ingeniería del Software, un enfoque práctico*” 7a ed. Connecticut University, New York: McGraw-Hill, 2010, pp. 295 - 301.