

Tecnología de Objetos

PATRONES DE DISEÑO

Experiencia en Diseño

- **Los analistas/diseñadores con gran experiencia aplican, de forma mayormente intuitiva y automática, criterios precisos que, de forma global, solucionan de forma elegante y efectiva los problemas de modelado software de sistemas reales.**
- **Usualmente estos diseñadores utilizan métodos, estructuras y subsistemas que son, a la vez, herramientas del diseño y partes de la solución final, de una manera que difícilmente puede transmitirse, en un sentido formal, a especialistas menos expertos.**

Problemas Repetidos

- Los “ingenieros de software” se enfrentan cada día a multitud de problemas de distinto calibre.
- La “efectividad” de un “ingeniero” se mide por su rapidez y acierto en la diagnosis, identificación y resolución de tales problemas.
- El mejor “ingeniero” es el que más reutiliza la misma solución -matizada- para resolver problemas similares.

Reinventar la Rueda

- **La Orientación-a-Objetos propugna “no reinventar la rueda” en la pura codificación respecto de la resolución de problemas. ¿Por que, entonces, reinventarla para el ataque genérico a problemas comunes de análisis, diseño e implementación?**
- **Debe existir alguna forma de comunicar al resto de los “ingenieros” los resultados encontrados tras mucho esfuerzo por alguno(s) de ellos.**
- **Se necesita, al fin, algún esquema de documentación que permita tal comunicación.**

¿Documentación?

- **La mera documentación de líneas de código resulta insuficiente, pues únicamente fomenta el uso de la técnica de “cut & paste”.**
- **El tipo de los problemas y soluciones a documentar es muy variado:**
 - PROGRAMACIÓN
 - ANÁLISIS
 - ARQUITECTURA
 - GESTIÓN, ETC.
- **Se necesita un formato de documentación único que aúne conceptualmente estos distintos tipos.**

La Esencia del Diseño

- **Los diseñadores bien-dotados poseen un sentido especial que “detecta” la completitud, en un sentido eminentemente arquitectónico, de un determinado diseño, con independencia de las posibles métricas y paradigmas utilizados.**
- **Naturalmente lo ideal sería extraer la quintaesencia de estos afortunados diseños para formular una suerte de “bálsamo de fierabrás” que pudieran ingerir los diseñadores noveles.**

Christopher Alexander

- Los trabajos de Alexander intentan identificar y resolver, en un marco descriptivo formal aunque no-exacto, problemas esenciales en el dominio de la arquitectura.
- Ha parecido adecuado a los diseñadores software trasladar muchas de las ideas de Alexander al dominio software.
- Alexander ha servido, en realidad, de catalizador de ciertas tendencias “constructivas” utilizadas en el diseño de sistemas software.

El Libro AIS

- **A Pattern Language: Towns/Building/Construction**, de *Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King y Shlomo Angel*, 1977, Oxford University Press. 253 patrones, con el formato específico propuesto por Alexander, se dan cita en este texto, en el que además se propugna una integración del mejor-vivir con el medio físico circundante: gente-gente-patrones-gente. Cuando se habla del “libro de Alexander” o del “libro AIS” (las iniciales de los primeros autores) se refieren a esta obra.

La Calidad sin Nombre

- **¿Existe en verdad una parte común en los buenos diseños, a veces tan dispares entre sí? Christopher Alexander así lo afirma, y da a esta parte la elusiva calificación de “la calidad que no se puede nombrar”.**
- **Alexander sostiene que existe un “algo innombrable” que no puede ser modelado únicamente por medio de un conjunto arbitrario de requerimientos. Los sistemas poseerían, así, una esencia cualitativa que les otorgaría verdadera identidad y equilibraría sus fuerzas internas.**

Calidad Adjetivada

Si bien la calidad intrínseca de los sistemas reales no tiene nombre, éstos pueden adjetivarse, siempre según Alexander, en razón de las características que tales poseen: **vivos, completos, libres, exactos, despersonalizados y eternos.**

Soluciones Reutilizables

“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”.

Christopher Alexander

Un Ejemplo Rural

Si nos fijamos en las construcciones de una determinada zona rural observaremos que todas ellas poseen apariencias parejas (tejados de pizarra con gran pendiente, etc.), pese a que los requerimientos personales por fuerza han debido ser distintos. De alguna manera la esencia del diseño se ha copiado de una construcción a otra, y a esta esencia se plegan de forma natural los diversos requerimientos. Diríase aquí que existe un “patrón” que soluciona de forma simple y efectiva los problemas de construcción en tal zona.

El Símil Textil

- Un patrón de diseño es **“Una solución a un problema en un determinado contexto”**. Tal solución es, empero, a la vez parte del “qué” y del “cómo” del sistema completo a construir: esto es, la pieza que conforma el patrón software es como la pieza del patrón de sastre que se utiliza para confeccionar vestidos y trajes, pues tal pieza, aparte de contener las especificaciones de corte y confección del producto final, representa a la vez, en apariencia, una parte de tal producto textil.

Patrón Documentario (I)

- **Intención:** sucinta descripción de lo que se pretende conseguir con el patrón.
- **También Conocido como:** otros nombres del mismo patrón.
- **Motivo:** explicación justificativa de la necesidad de que el patrón exista como entidad autónoma.
- **Aplicabilidad:** lista de usos para los que resulta especialmente adecuado el patrón que se describe.

Patrón Documentario (II)

- **Estructura:** descripción gráfica de los comportamientos, acciones y relaciones de los objetos que participan en el patrón.
- **Participantes:** diccionario de las partes que componen el patrón.
- **Colaboraciones:** diccionario de las relaciones e interacciones entre los participantes en un patrón.
- **Consecuencias:** detalle de los posibles beneficios y perjuicios que pueden derivarse del uso del patrón.

Patrón Documentario (III)

- **Implementación:** detalle de las posibles implementaciones y catálogo de las decisiones de diseño en la codificación de soluciones concretas basadas en el patrón.
- **Código de Ejemplo:** planteamiento de código práctico referido a un ejemplo (o ejemplos) suficientemente representativo del uso del patrón.
- **Usos Conocidos:** detalle de bibliotecas, productos y sistemas en que se ha utilizado el patrón.

Patrón Documentario (IV)

- **Patrones Relacionados:** referencias a otros patrones que bien son directamente utilizados por el descrito bien representan soluciones complementarias o suplementarias al mismo.

Catálogos de Patrones

Si aceptamos que los patrones pueden resultar útiles en el desarrollo de software, el siguiente paso es reunirlos en catálogos de forma que resulten accesibles mediante distintos criterios, pues lo que necesitamos no es tan sólo la completa descripción de cada uno de los patrones sino, esencialmente, la correspondencia entre un problema real y un patrón (o conjunto de patrones) determinado.

La Curva de Aprendizaje

- **Lo que se pretende con un catálogo de patrones no es favorecer al diseñador experto (que quizás no necesite en absoluto de los patrones), sino más bien ayudar al diseñador inexperto a adquirir con cierta rapidez las habilidades de aquél, como también comunicar al posible cliente, si es el caso, las decisiones de diseño de forma clara y autosuficiente.**
- **Un Catálogo de Patrones es un medio para comunicar la experiencia de forma efectiva, reduciendo lo que se conoce como “curva de aprendizaje” del diseño.**

El Libro GOF

- Existen varios catálogos de patrones pero la capacidad de elección todavía se basa en el conocimiento completo de los mismos.
- El catálogo de patrones más famoso es el contenido en el libro “**Design Patterns: Elements of Reusable Object-Oriented Software**”, de *Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides*, 1995, Addison-Wesley, 0-201-63361-2, también conocido como el **LIBRO GOF** (Gang-Of-Four Book).

Tipos de Patrones

SEGÚN EL LIBRO GOF:

- ***De Creación***: abstraen el proceso de creación de instancias.
- ***Estructurales***: se ocupan de cómo clases y objetos son utilizados para componer estructuras de mayor tamaño.
- ***De Comportamiento***: atañen a los algoritmos y a la asignación de responsabilidades entre objetos.

Patrones de Creación (I)

- **Abstract Factory:** provee un interfaz para crear familias de objetos dependientes o relacionados sin necesidad de especificar sus clases concretas (p.e.: widgets en distintas plataformas)
- **Builder:** separa la construcción de un objeto complejo de su representación, de tal manera que el mismo proceso de construcción puede crear diferentes representaciones (p.e.: conversores de formatos de texto)

Patrones de Creación (II)

- **Factory Method:** define un intefaz para crear un objeto, pero deja que las subclases decidan qué clase instanciar. Este patrón delega la instanciación de una clase a sus subclases (como un constructor virtual).
- **Prototype:** especifica las clases de objetos a crear usando un ejemplar prototípico, y crea nuevos objetos copiando este prototipo.
- **Singleton:** asegura que una clase sólo tendrá una instancia, y provee un punto global de acceso a la misma.

Patrones Estructurales

- **Adapter**
- **Bridge**
- **Composite**
- **Decorator**
- **Facade**
- **Flyweight**
- **Proxy**

Patrones de Comportamiento

- **Chain of Responsibility**
- **Command**
- **Interpreter**
- **Iterator**
- **Mediator**
- **Memento**
- **Observer**
- **State, Strategy**
- **Template Method**
- **Visitor**

Breves Notas

- **¿Patrones como moda?
Mejor como Intención.**
- **¿Patrones como solución?
Mejor como núcleo de
soluciones.**
- **¿Patrones como normas?
Mejor como sugerencias.**

Patrones de Diseño Software

PREGUNTAS