Aguirre Olvera Miguel Angel

Tarea 5

Definición oficial de AS según IEEE:

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

Definición oficial de AS de Grady Booch

La arquitectura del software abarca el conjunto de decisiones significativas sobre la organización de un sistema de software

- selección de los elementos estructurales y sus interfaces por las cuales se compone un sistema

- comportamiento especificado en colaboraciones entre esos elementos

- composición de estas estructuras y comportamientos elementos en un subsistema más grande

- estilo arquitectónico que guía esta organización

Diferencia entre arquitectura e ingeniería

Obsérvese entonces que la noción clave de la arquitectura es la organización (un concepto cualitativo o estructural), mientras que la ingeniería tiene fundamentalmente que ver con una sistematicidad susceptible de cuantificarse.

Definición de ingeniería de software según IEEE

Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software

ADLS

Lenguajes de descripción de arquitectura un ADL como una entidad consistente en cuatro “Cs”: componentes, conectores, configuraciones y restricciones (constraints). Una de las definiciones más tempranas es la de Vestal [Ves93] quien sostiene que un ADL debe modelar o soportar los siguientes conceptos:

- Componentes

- Conexiones

- Composición jerárquica, en la que un componente puede contener una sub-arquitectura completa

- Paradigmas de computación, es decir, semánticas, restricciones y propiedades no funcionales

- Paradigmas de comunicación

– Modelos formales subyacentes

- Soporte de herramientas para modelado, análisis, evaluación y verificación - Composición automática de código aplicativo

Abstracción Arquitectónica según booch

Una abstracción denota las características esenciales de un objeto que lo distinguen de otras clases de objetos y provee de este modo delimitaciones conceptuales bien definidas relativas a la perspectiva del observador

Diferencia entre arquitectura y diseño

En algunas perspectivas la arquitectura se encuentra un nivel arriba de abstracción por encima del diseño, o es simplemente otro paso en el proceso de desarrollo de software

Otra establece que la arquitectura es algo nuevo y en alguna medida diferente del diseño

¿Que son los patrones de diseño?

Es una solución general reusable que puede ser aplicada a problemas que ocurren comúnmente en el desarrollo de software, es la descripción o plantilla de cómo resolver un problema que puede ser usada en diferentes situaciones.

Los patrones de diseño son soluciones probadas, expresivas y fáciles de mantener. Muchos developers esta familiarizados con los Patrones de diseño, así que podemos decir que es un tipo de estándar de desarrollo.

Pasos para describir patrones de diseño

Para describir un patrón se utilizan plantillas más o menos estandarizadas, de forma que sus características se expresen uniformemente y puedan constituir efectivamente un medio de comunicación uniforme entre diseñadores. Varios autores eminentes en esta área han propuesto plantillas ligeramente distintas, aunque la mayoría de estas plantillas definen los mismos conceptos básicos. La plantilla más común consta de los siguientes apartados:

* **Nombre del patrón:** nombre estándar del patrón por el cual será reconocido en la comunidad (normalmente se expresan en inglés). Otros nombres de uso común para el patrón.
* **Clasificación del patrón:** creacional, estructural o de comportamiento.
* **Motivación:** Escenario de ejemplo para la aplicación del patrón.
* **Aplicabilidad:** Usos comunes y criterios de aplicabilidad del patrón.
* **Estructura:** Diagramas de clases oportunos para describir las clases que intervienen en el patrón.
* **Participantes:** Enumeración y descripción de las entidades abstractas (y sus roles) que participan en el patrón.
* **Colaboraciones:** Explicación de las interrelaciones que se dan entre los participantes.
* **Consecuencias:** Consecuencias positivas y negativas en el diseño derivadas de la aplicación del patrón.
* **Implementación:** Técnicas o comentarios oportunos de cara a la implementación del patrón.
* **Código de ejemplo:** Código fuente ejemplo de implementación del patrón.
* **Usos conocidos:** Ejemplos de sistemas reales que usan el patrón.
* **Patrones relacionados:** Referencias cruzadas con otros patrones.

Enumere los distintos tipos de patrones de diseño

1. Patrones creacionales
2. Patrones estructurales
3. Patrones de comportamiento

Estilo de flujo de datos

Esta familia de estilos enfatiza la reutilización y la modificabilidad. Es apropiada para sistemas que implementan transformaciones de datos en pasos sucesivos. Ejemplares de estas serían las arquitecturas de tubería-filtros y las de proceso secuencial en lote.

La arquitectura de flujo de datos es una arquitectura de computadores que contrasta directamente con la tradicional Arquitectura de von Neumann o de estructuras de control. Las arquitecturas de flujo de datos no se basan en un contador de programa (al menos conceptualmente) en tanto en cuanto la posibilidad de ejecución de las instrucciones solamente viene determinada por la disponibilidad de los argumentos de entrada de las instrucciones.

Estilo tuberías y filtros

Por los tubos fluyen dato, y son salidas de un filtro y la entrada de otro.

Cada filtro admite uno o varios tubos

Cada filtro es independiente del resto y no conocen la identidad de los filtros antes y después de él.

No importa la secuencia (paralelismo).

Desventajas:

Si un filtro necesita contar con todas las entradas antes de producir la salida, requiere entonces de un buffer muy grande.

Cada filtro actúa como proceso independiente consumiendo recursos cada vez que se invoca.

Estilo centrado de datos

Sistemas que se fundan en acceso y actualización de costos estructurales de almacenamiento. Subestilos: repositorios, las bases de datos, arquitecturas basadas en hipertexto y las arquitecturas de pizarra.

Arquitectura de pizarra o repositorio

La arquitectura en pizarra consta de múltiples elementos funcionales, denominados **agentes**, y un instrumento de control denominado **pizarra**.

Los agentes suelen estar especializados en una tarea concreta o elemental. Todos ellos cooperan para alcanzar una meta común, si bien, sus objetivos individuales no están aparentemente coordinados.

El comportamiento básico de cualquier agente consiste en examinar la pizarra, realizar su tarea y escribir sus conclusiones en la misma pizarra. De esta manera, otro agente puede trabajar sobre los resultados generados por otro.

La computación termina cuando se alcanza alguna condición deseada entre los resultados escritos en la pizarra.

La pizarra tiene un doble papel. Por una parte, coordina a los distintos agentes y, por otra, facilita su intercomunicación. El estado inicial de la pizarra es una descripción del problema que resolver y el estado final será la solución del problema.

Los resultados generados por los agentes deben responder a un lenguaje y semántica común. En general, se suelen utilizar formalismos lógicos o matemáticos, tales como expresiones lógicas de primer orden.