

¿Cómo crear representaciones coherentes del modelo de diseño?

Linda María Fernández Olvera
UPIICSA-IPN
Ingeniería de Diseño
lfernandezol1500@alumno.ipn.mx

Abstract – Este ensayo plantea una respuesta a las consideraciones necesarias para diseñar representaciones de los sistemas que incluyen su estructura y elementos con características del funcionamiento. Se plantean los pasos necesarios para definir la arquitectura hasta el punto de la evaluación de calidad.

Índice de Términos – diseño, arquitectura, arquetipos, patrones, estilos, mapeo.

I. INTRODUCCIÓN

El diseño elabora representaciones coherentes del software, está motivado por la información, para determinar el plano preliminar del cual se elabora el software es necesario crear un enfoque sistemático que obtenga el diseño arquitectónico.

La arquitectura va más allá de la estructura que tomará el proyecto, son las decisiones que se toman en la etapa temprana del diseño y tienen un efecto profundo en todas las demás acciones, eliminan las restricciones prematuras que producen una mala implementación del software.

La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos.

La arquitectura no es el software operativo. Es una representación que permite analizar la efectividad del diseño para cumplir los requerimientos establecidos, considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil y reducir los riesgos asociados con la construcción del software.

El diseño del software comienza una vez que se han analizado y modelado los requerimientos, es la última acción de la ingeniería de software dentro de la actividad de modelado y prepara la etapa de construcción.

Un componente del software puede ser algo tan simple como un módulo de programa o una clase orientada a objeto, pero también puede ampliarse para que incluya bases de datos y “middleware” que permitan la configuración de una red de

clientes y servidores.

Un diseño es una instancia de una arquitectura, similar a un objeto que es una instancia de una clase.

II. DESARROLLO

Una descripción arquitectónica en realidad es un conjunto de productos del trabajo que reflejan puntos de vista distintos del sistema.

El estándar IEEE define una descripción arquitectónica (DA) como “un conjunto de productos para documentar una arquitectura”. Los desarrolladores desean lineamientos claros y decisivos sobre la forma de proceder con el diseño. Los consumidores desean la comprensión clara de los cambios ambientales que deben ocurrir y las garantías de que la arquitectura satisfará las necesidades de negocios.

Los objetivos de la descripción arquitectónica buscan establecer un marco conceptual con un vocabulario que se use durante el diseño de la arquitectura del software, proporcionar lineamientos detallados para representar una descripción arquitectónica y estimular las mejores prácticas del diseño arquitectónico.

Cada perspectiva desarrollada como parte de una descripción arquitectónica se aboca a una preocupación de un participante específico. El arquitecto del sistema considera varias alternativas y decide en última instancia las características arquitectónicas específicas que satisfagan del mejor modo la preocupación.

A. Géneros Arquitectónicos

El género implica una categoría específica dentro del dominio general del software. Dentro de cada categoría hay varias subcategorías.

Grady Booch, en su libro *Handbook of Software Architecture*, sugiere los siguientes géneros arquitectónicos para sistemas basados en software:

Inteligencia artificial: Sistemas que simulan o incrementan la cognición humana, su locomoción u otros procesos orgánicos.

- Comerciales y no lucrativos: Sistemas que son fundamentales para la operación de una empresa de negocios.
- Comunicaciones: Sistemas que proveen la infraestructura para transferir y manejar datos, para conectar usuarios de éstos o para presentar datos en la frontera de una infraestructura.
- Contenido de autor: Sistemas que se emplean para crear o manipular artefactos de texto o multimedia.
- Dispositivos: Sistemas que interactúan con el mundo físico a fin de brindar algún servicio puntual a un individuo.
- Entretenimiento y deportes: Sistemas que administran eventos públicos o que proveen una experiencia grupal de entretenimiento.
- Financieros: Sistemas que proporcionan la infraestructura para transferir y manejar dinero y otros títulos.
- Juegos: Sistemas que dan una experiencia de entretenimiento a individuos o grupos.
- Gobierno: Sistemas que dan apoyo a la conducción y operaciones de una institución política local, estatal, federal, global o de otro tipo.
- Industrial: Sistemas que simulan o controlan procesos físicos.
- Legal: Sistemas que dan apoyo a la industria jurídica.
- Médicos: Sistemas que diagnostican, curan o contribuyen a la investigación médica.
- Militares: Sistemas de consulta, comunicaciones, comando, control e inteligencia, así como de armas ofensivas y defensivas.
- Sistemas operativos: Sistemas que están inmediatamente instalados en el hardware para dar servicios de software básico.
- Plataformas: Sistemas que se encuentran en los sistemas operativos para brindar servicios avanzados.
- Científicos: Sistemas que se emplean para hacer investigación científica y aplicada.
- Herramientas: Sistemas que se utilizan para desarrollar otros sistemas.
- Transporte: Sistemas que controlan vehículos acuáticos, terrestres, aéreos o espaciales.
- Utilidades: Sistemas que interactúan con otro software para brindar algún servicio específico.

Cada género representa un desafío único.

B. Estilos Arquitectónicos

Un estilo arquitectónico es una transformación que se impone al diseño de todo el sistema. El objetivo es establecer una estructura para todos los componentes del sistema.

El software construido para sistemas basados en computadora también tiene uno de muchos estilos arquitectónicos. Cada estilo describe una categoría de

sistemas que incluye un conjunto de componentes (como una base de datos o módulos de cómputo) que realizan una función requerida por el sistema, un conjunto de conectores que permiten la “comunicación, coordinación y cooperación” entre los componentes, restricciones que definen cómo se integran los componentes para formar el sistema y modelos semánticos que permiten que un diseñador entienda las propiedades generales del sistema al analizar las propiedades conocidas de sus partes constituyentes.

Un patrón arquitectónico, como un estilo de arquitectura, impone la transformación del diseño de una arquitectura. Sin embargo, un patrón difiere de un estilo en varias formas fundamentales: el alcance del patrón es menos amplio y se centra en un aspecto de la arquitectura más que en el total de ésta, un patrón impone una regla a la arquitectura, describe la manera en la que el software manejará ciertos aspectos de su funcionalidad en el nivel de la infraestructura.

Aunque se han creado millones de sistemas basados en computadoras, la gran mayoría se clasifica en un número relativamente pequeño de estilos de arquitectura:

Arquitecturas centradas en los datos. En el centro de esta arquitectura se halla un almacenamiento de datos (como un archivo o base de datos) al que acceden con frecuencia otros componentes que actualizan, agregan, eliminan o modifican los datos de cierto modo dentro del almacenamiento.

Arquitecturas de flujo de datos. Esta arquitectura se aplica cuando datos de entrada van a transformarse en datos de salida a través de una serie de componentes computacionales o manipuladores.

Arquitecturas de llamar y regresar. Este estilo arquitectónico permite obtener una estructura de programa que es relativamente fácil de modificar y escalar. Estos estilos incluyen:

- Arquitecturas de programa principal/subprograma. Esta estructura clásica de programa descompone una función en una jerarquía de control en la que un programa “principal” invoca cierto número de componentes de programa que a su vez invocan a otros.

- Arquitecturas de llamada de procedimiento remoto. Los componentes de una arquitectura de programa principal/subprograma están distribuidos a través de computadoras múltiples en una red.

Arquitecturas orientadas a objetos. Los componentes de un sistema incluyen datos y las operaciones que deben aplicarse para manipularlos. La comunicación y coordinación entre los componentes se consigue mediante la transmisión de mensajes.

Arquitecturas en capas. Se define un número de capas

diferentes; cada una ejecuta operaciones que se aproximan progresivamente al conjunto de instrucciones de máquina. En la capa externa, los componentes atienden las operaciones de la interfaz de usuario. En la interna, los componentes realizan la interfaz con el sistema operativo. Las capas intermedias proveen servicios de utilerías y funciones de software de aplicación.

Estructuras arquitectónicas canónicas

Estructura funcional. Los componentes representan entidades de función o procesamiento. Los conectores representan interfaces que proveen la capacidad de “usar” o “pasar datos a” un componente. Las propiedades describen la naturaleza de los componentes y la organización de las interfaces.

Estructura de implementación. Los componentes son paquetes, clases, objetos, procedimientos, funciones, métodos, etc., que son vehículos para empaquetar funciones en varios niveles de abstracción. Los conectores incluyen la capacidad de pasar datos y control, compartir datos, “usar” y “ser una instancia de”. Las propiedades se centran en las características de la calidad

Estructura de concurrencia. Los componentes representan unidades de concurrencia que están organizadas como tareas o trayectorias paralelas. Las relaciones incluyen sincronizarse con, tiene-mayor-prioridad-que, envía-datos-a, no-corre-sin y no-corre-con. Las propiedades relevantes para esta estructura incluyen prioridad, anticipación y tiempo de ejecución.

Estructura física. Esta estructura es similar al modelo de despliegue desarrollado como parte del diseño. Los componentes son el hardware físico en el que reside el software. Los conectores son las interfaces entre los componentes del hardware y las propiedades incluyen la capacidad, ancho de banda y rendimiento, entre otros atributos.

Estructura de desarrollo. Esta estructura define los componentes, productos del trabajo y otras fuentes de información que se requieren a medida que avanza la ingeniería de software. Los conectores representan las relaciones entre los productos del trabajo; las propiedades identifican las características de cada aspecto.

Patrones arquitectónicos

Los patrones arquitectónicos abordan un problema de aplicación específica dentro de un contexto dado y sujeto a limitaciones y restricciones. El patrón propone una solución arquitectónica que sirve como base para el diseño de la arquitectura.

Organización y refinamiento

Es importante establecer un conjunto de criterios de diseño que se usan para evaluar el diseño arquitectónico obtenido debido a que es frecuente que el proceso de diseño permita varias alternativas de arquitectura. Por lo mismo es necesario plantear preguntas para darnos la visión del estilo de arquitectura.

Control. ¿Cómo se administra el control dentro de la arquitectura? ¿Existe una jerarquía de control distinta? Si es así, ¿cuál es el papel de los componentes dentro de esta jerarquía de control? ¿Cómo transfieren el control los componentes dentro del sistema?

Datos. ¿Cómo se comunican los datos entre los componentes? ¿El flujo de datos es continuo o los objetos de datos pasan al sistema en forma esporádica? ¿Cuál es el modo de transferencia de datos? ¿Cómo interactúan los componentes funcionales con los componentes de datos?

C. Diseño Arquitectónico

Cuando se inicia con el diseño arquitectónico, el software que se va a desarrollar debe situarse en contexto, es decir, el diseño debe definir las entidades con las que interactúa el software y la naturaleza de dicha interacción. El arquitecto del software usa un diagrama de contexto arquitectónico (DCA) para modelar la manera en la que el software interactúa con entidades más allá de sus fronteras.

Los sistemas que interactúan con el sistema están representados de la siguiente manera:

- **Sistemas superiores:** aquellos que utilizan al sistema objetivo como parte de algún esquema de procesamiento de alto nivel.
- **Sistemas subordinados:** los que son usados por el sistema objetivo y proveen datos o procesamiento que son necesarios para completar las funciones del sistema objetivo.
- **Sistemas entre iguales:** son los que interactúan sobre una base de igualdad.
- **Actores:** entidades (personas, dispositivos, etc.) que interactúan con el sistema objetivo mediante la producción o consumo de información que es necesaria para el procesamiento de los requerimientos.

Cada una de estas entidades externas se comunica con el sistema objetivo a través de una interfaz.

Arquetipos

Un arquetipo es una clase o un patrón que representa una abstracción fundamental de importancia crítica para el diseño de una arquitectura para el sistema objetivo. Representan

elementos estables de la arquitectura, pero que son implementadas en muchos modos diferentes con base en el comportamiento del sistema. Los arquetipos se obtienen con el estudio de las clases de análisis definidas como parte del modelo de los requerimientos.

- **Nodo.** Representa una colección cohesiva de elementos de entrada y salida de la función.
- **Detector.** Abstracción que incluye todos los equipos de detección que alimentan con información al sistema objetivo.
- **Indicador.** Abstracción que representa todos los mecanismos que indican que está ocurriendo una condición de alarma.
- **Controlador.** Abstracción que ilustra el mecanismo que permite armar o desarmar un nodo. Si los controladores residen en una red, tienen la capacidad de comunicarse entre sí.

En la Figura 1 se muestran los arquetipos con el empleo de notación UML.

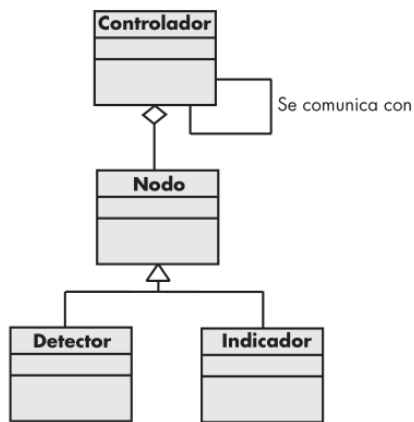


Figura 1. Notación UML de los arquetipos

Refinamiento de la arquitectura hacia los componentes

Conforme la arquitectura se refina hacia los componentes, comienza a emerger la estructura del sistema. Estas clases de análisis representan entidades dentro del dominio de aplicación que deben enfrentarse dentro de la arquitectura del software. Entonces, el dominio de aplicación es una fuente para la obtención y refinamiento de los componentes.

Descripción de las instancias del sistema

Para representar el contexto del sistema, se definieron los arquetipos que indican las abstracciones importantes dentro del dominio del problema, es visible la estructura general del sistema y están identificadas las componentes principales del software. Sin embargo, es necesario más refinamiento, por lo tanto, se desarrollan instancias de la arquitectura. Esto significa que la arquitectura se aplica a un problema específico con objeto de demostrar que tanto ella como sus componentes son apropiados.

Método de la negociación para analizar la arquitectura

El Instituto de Ingeniería de Software (SEI) desarrolló el método de la negociación para analizar la arquitectura (Architecture trade-off analysis method -ATAM), que establece un proceso de evaluación iterativo para arquitecturas de software. Las actividades de análisis del diseño que se mencionan a continuación se llevan a cabo en forma iterativa:

1. Escenarios de investigación. Se desarrolla un conjunto de casos de uso para representar al sistema desde el punto de vista del usuario.
2. Obtención de los requerimientos y restricciones, y descripción del ambiente.
3. Descripción de los estilos o patrones de arquitectura elegidos para abordar los escenarios y requerimientos. Debe describirse el estilo arquitectónico con el empleo de las siguientes perspectivas arquitectónicas:

- Perspectiva modular para el análisis de las asignaciones de trabajo con componentes y grado en el que se logra el ocultamiento de información.
- Perspectiva del proceso para el análisis del desempeño del sistema.
- Perspectiva del flujo de datos para analizar el grado en el que la arquitectura satisface los requerimientos funcionales.

4. Evaluación de los atributos de calidad, considerando cada atributo por separado.

5. Identificación de la sensibilidad de los atributos de calidad de varios atributos arquitectónicos para un estilo de arquitectura específico.

6. Crítica de las arquitecturas candidatas con el uso del análisis de sensibilidad.

Complejidad arquitectónica

Una técnica útil para evaluar la complejidad general de una arquitectura propuesta es considerar las dependencias entre los componentes dentro de la arquitectura. Estas dependencias están motivadas por el flujo de la información o por el control dentro del sistema.

Lenguajes de descripción arquitectónica

El lenguaje de descripción arquitectónica (LDA) provee la semántica y sintaxis para describir una arquitectura de software. Debe brindar al diseñador la capacidad de desintegrar los componentes arquitectónicos, integrar componentes individuales en bloques arquitectónicos más grandes y representar las interfaces.

D. Mapeo de la arquitectura con el uso de flujo de datos

Para ilustrar un enfoque al mapeo arquitectónico, considere la arquitectura denominada de llamada y regreso, estructura muy común para muchos tipos de sistemas.

Una técnica de mapeo llamada diseño estructurado se caracteriza con frecuencia como método de diseño orientado al flujo porque provee una transición conveniente de un diagrama de flujo de datos a la arquitectura del software.

El mapeo de transformación es un conjunto de pasos de diseño que permite mapear un DFD con características de flujo de transformación en un estilo arquitectónico específico. Para mapear estos diagramas de flujo de datos en una arquitectura de software deben darse los siguientes pasos de diseño:

Paso 1. Revisión del modelo del sistema fundamental. El modelo del sistema fundamental o diagrama de contexto ilustra la función de seguridad como una transformación única, y representa a los productores y consumidores externos de los datos que fluyen hacia dentro y fuera de la función.

Paso 2. Revisar y mejorar los diagramas de flujos de datos para el software. La información obtenida del modelo de requerimientos se refina para producir más detalles.

Paso 3. Determinar si el DFD tiene características de flujo de transformación o de transacción. Al evaluar el DFD se observa que los datos entran al software por una trayectoria de ingreso y lo abandonan por tres trayectorias de salida. Por tanto, se adoptará una característica general de transformación para el flujo de la información.

Paso 4. Aísle el centro de transformación, especificando las fronteras de entrada y salida del flujo. Los datos de entrada fluyen por una trayectoria en la que la información pasa de su forma externa a su forma interna; el flujo de salida convierte los datos internalizados a su forma externa. Las fronteras de los flujos de entrada y salida quedan abiertas a la interpretación.

Paso 5. Realizar el “rediseño de primer nivel”. La arquitectura del programa obtenida con este mapeo da como resultado una distribución del control de arriba abajo. El rediseño lleva a una estructura de programa en la que los componentes de alto nivel ejecutan la toma de decisiones y los de bajo nivel realizan la mayor parte del trabajo de entrada, computación y salida. Los componentes de nivel medio llevan a cabo cierto control y moderan las cantidades de trabajo.

Paso 6. Realizar “rediseño de segundo nivel”. El rediseño de segundo nivel se logra con el mapeo de transformaciones individuales (burbujas) de un diagrama de flujo de datos en módulos apropiados dentro de la arquitectura.

Paso 7. Refinar la arquitectura de primera iteración con el empleo de heurísticos de diseño para mejorar la calidad del software. Siempre es posible refinar la arquitectura de primera iteración, aplicando conceptos de independencia de funciones.

III. CONCLUSIÓN

La arquitectura de software modela una visión del sistema que se construirá. No solo define la estructura del sistema sino también la organización de los componentes de software, sus propiedades y conexiones, que pueden incluir módulos y distintas representaciones de datos.

En un género arquitectónico hay estilos y patrones diferentes que describen categorías de sistemas que agrupan un conjunto de componentes con funciones requeridas por el sistema.

El diseño arquitectónico se obtiene a partir de la definición del contexto, después se identifican las abstracciones de alto nivel (arquetipos) que representan a los elementos fundamentales del sistema. El diseño continúa con el dominio de la implementación donde se identifican componentes y se representan en el contexto de la arquitectura. Al final, se desarrollan instancias específicas de la arquitectura para hacer pruebas reales y evaluar el sistema.

REFERENCES

- [1] Roger S. Pressman, “*Ingeniería del Software, un enfoque práctico*” 7a ed. Connecticut University, New York: McGraw-Hill, 2010, pp. 206 - 233.