|  |  |
| --- | --- |
| http://deontologiauba.wikispaces.com/file/view/ingenieria_software.jpg/243749521/ingenieria_software.jpgMODELO DE ANALISISUNIDAD 3 | Descripción breveEl modelo de análisis tiene como objetivo generar una arquitectura de objetos que sirva como base para el diseño posterior del sistema. Dependiendo del tipo de aplicación existen diversas arquitecturas que se pueden utilizar, siendo de nuestro interés aquellas arquitecturas especialmente diseñadas para el manejo de los sistemas de informaciónTAPIA HERNANDEZ ARELIFUNDAMENTOS DE INGENIERIA DE SOFTWARE |

**INDICE**

**UNIDAD 3 MODELO DE ANALISIS**

**3.1 Arquitectura de clases…………………………………………………….PAG.2**

**3.2 Identificación de clases según estereotipo………...………………....PAG.2**

**3.3 Clases………………………………………………………………..……….PAG.3**

**3.4 Diagramas de secuencias……………….……………………………….PAG.3**

**3.5 Diccionario según modelos……………………………………………...PAG.4**

**3.6 Herramientas CASE para análisis……………………………………..…PAG.5**

 **Bibliografía…………………………………………………………….…..…PAG.7**

**3.1 Arquitectura de Clases**

El modelo de análisis tiene como objetivo generar una arquitectura de objetos que sirva como base para el diseño posterior del sistema. Dependiendo del tipo de aplicación existen diversas arquitecturas que se pueden utilizar, siendo de nuestro interés aquellas arquitecturas especialmente diseñadas para el manejo de los sistemas de información, las cuales involucran ricas bordes de usuario y accesos a base de datos como aspectos fundamentales de la arquitectura. En término de las propias arquitecturas, éstas se distinguen según la organización de la funcionalidad que ofrecen los objetos dentro de ellas o la dimensión de los objetos. Esta dimensión corresponde a los diferentes tipos de funcionalidad que manejan los objetos dentro la arquitectura. Por ejemplo, en el caso de funcionalidad para el manejo de bordes y base de datos, si existen tipos distintos de objetos para el manejo de cada una de estas por separado, entonces se considera que la arquitectura es de dos dimensiones. Por el contrario, si todos los objetos manejan de manera indistinta los dos tipos de funcionalidades, entonces se considera que la arquitectura es de una sola dimensión.

La vista o presentación de la información corresponde a las bordes que se le presentan al usuario para el manejo dela información, donde por lo general pueden existir múltiples vistas sobre un mismo modelo. Típicamente la información representa el dominio del problema y es almacenada en una base de datos. Por otro lado el control corresponde a la manipulación de la información a través de sus diversas presentaciones. Y aunque existe cierta dependencia entre estas tres dimensiones se considera que la manera de presentar la información es independiente dela propia información y de cómo esta se controla. Sin embargo, cada una de ellas probablemente experimente cambios a lo largo de la vida del sistema, donde el control es el más propenso a ser modificado, seguido de la vista y finalmente el modelo. En el modelo de análisis descrito aquí utilizaremos como base la arquitectura.

**3.2 Identificación de clases según estereotipos**

Clases con Estereotipos

El tipo de funcionalidad o “la razón de ser” de un objeto dentro de una arquitectura se le conoce como su estereotipo

 Para los sistemas de información la arquitectura del sistema según nuestro modelo de análisis se basa en tres estereotipos básicos de objetos:

 El estereotipo entidad (“entity” en inglés) para objetos que guarden información sobre el estado interno del sistema, a corto y largo plazo, correspondiente al dominio del problema. Todo comportamiento naturalmente acoplado con esta información también se incluye en los objeto entidad. Un ejemplo de un objeto entidad es un registro de usuario con sus datos y comportamiento asociados.

 El estereotipo interface o borde (“boundary” en inglés) para objetos que implementen la presentación o vista correspondiente a las bordes del sistema hacia el mundo externo, para todo tipo de actores, no sólo usuarios humanos. Un ejemplo de un objeto borde es la funcionalidad de interface de usuario para insertar o modificar información sobre el registro de usuario.

El estereotipo control (“control” en inglés) para objetos que implementen el comportamiento o control especificando cuando y como el sistema cambia de estado, correspondiente a los casos de uso. Los objetos control modelan funcionalidad que no se liga naturalmente con ningún otro tipo de objeto, como el comportamiento que opera en varios objetos entidad a la vez, por ejemplo, hacer alguna computación y luego devolver el resultado a un objeto borde. Un ejemplo típico de objeto control es analizar el uso del sistema por parte de algún usuario registrado y presentar tal información posteriormente. Este comportamiento no le pertenece a ningún objeto entidad u objeto borde específico. Nótese que no hay ninguna restricción a los diferentes estereotipos que puedan utilizarse, no solamente las tres anteriores.

**3.3 clases**

Una clase es una construcción que se utiliza como un modelo (o plantilla) para crear objetos de ese tipo. El modelo describe el estado y el comportamiento que todos los objetos de la clase comparten. Un objeto de una determinada clase se denomina una instancia de la clase. La clase que contiene (y se utilizó para crear) esa instancia se puede considerar como del tipo de ese objeto. Por ejemplo, una instancia del objeto de la clase "Persona" sería del tipo "Persona".

Una clase por lo general representa un sustantivo, como una persona, lugar o (posiblemente bastante abstracta) cosa - es el modelo de un concepto dentro de un programa de computadora. Fundamentalmente, encapsula el estado y el comportamiento del concepto que representa. Encapsula el estado a través de marcadores de datos llamados atributos (o variable miembro o variables de instancia), y encapsula el comportamiento a través de secciones de código reutilizables llamados métodos.

**3.4 diagrama de secuencias**

Un diagrama de secuencia muestra:

* Interacción de un conjunto de objetos en una aplicación a través del tiempo.
* Un conjunto de mensajes, dispuestos en una secuencia temporal.
* Cada rol en la secuencia como una línea de vida, es decir: una línea vertical.

Un diagrama de secuencia representa una interacción como un gráfico bidimensional.

Ø La dimensión vertical: es el eje del tiempo

Ø La dimensión horizontal muestra los roles de clasificador que representan objetos individuales en la colaboración

Un rol de clasificador:

Ø Es la descripción de un objeto que desempeña un determinado papel dentro de una interacción, distinto de los otros objetos de la misma clase.

Ø La primera utilización de los diagramas de secuencia corresponde a la documentación de los casos de uso, se concentra en la descripción de la interacción,

 La segunda utilización corresponde a un uso más informático y permite la representación precisa de las interacciones entre objetos.

Por lo tanto puede mostrar:

Ø Escenario como la historia individual de la transacción que detalla los casos de uso, aclarándolos al nivel de mensajes de los objetos existentes, como también

Ø El uso de los mensajes de las clases diseñadas en el contexto de una operación.

Cuando está implementado el comportamiento, Cada mensaje en un diagrama de secuencia Corresponde a:

* Una operación en una clase,
* A un evento disparador, o
* A una transición en una máquina de estados.

**3.5 diccionario de clases según módulos**

Diccionario de datos

Los diccionarios de datos son el segundo componente del análisis del flujo de datos. En sí mismos los diagramas de flujo de datos no describen por completo el objeto de la investigación. El diccionario de datos proporciona información adicional sobre el sistema. Esta sección analiza que es un diccionario de datos, por qué se necesita en el análisis de flujo de datos y como desarrollarlo. Se utilizará el ejemplo del sistema de contabilidad para describir los diccionarios de datos.

Un diccionario de datos es una lista de todos los elementos incluido en el conjunto de los diagramas de flujo de datos que describen un sistema. Los elementos principales en un sistema, estudiados en las secciones anteriores, son el flujo de datos, el almacenamiento de datos y los procesos. El diccionario de datos almacena detalles y descripciones de estos elementos.

Si los analistas desean conocer cuántos caracteres hay en un dato, con qué otros nombres se le conocen en el sistema, o en donde se utilizan dentro del sistema deben ser capaces de encontrar la respuesta en un diccionario de datos desarrollado apropiadamente.

El diccionario de dato se desarrolla durante el análisis de flujo de datos y ayuda el analista involucrado en la determinación de los requerimientos de sistemas. Sin embargo, como se verá más adelante, también el contenido del diccionario de datos se utiliza durante el diseño del sistema.

En informática, base de datos acerca de la terminología que se utilizará en un sistema de información. Para comprender mejor el significado de un diccionario de datos, puede considerarse su contenido como "datos acerca de los datos"; es decir, descripciones de todos los demás objetos (archivos, programas, informes, sinónimos...) existentes en el sistema. Un diccionario de datos almacena la totalidad de los diversos esquemas y especificaciones de archivos, así como sus ubicaciones. Si es completo incluye también información acerca de qué programas utilizan qué datos, y qué usuarios están interesados en unos u otros informes. Por lo general, el diccionario de datos está integrado en el sistema de información que describe.

**3.6 Herramientas CASE para el análisis**

De acuerdo con Kendall el desarrollo de sistema es asistida por ordenadores es la aplicación de informática, es acelerar el proceso para que han sido desarrolladas. En cambio la herramienta CASE (Computer-Aided Software Engineering) sirve para apoyar una fase del ciclo de vida del sistema.

Cuando se planifica la base de datos permite escoger una herramienta CASE para llevar de forma eficaz y posible las tareas, también suelen incluir.

• Un diccionario para los datos de la aplicación de base de datos.

• Herramientas de diseño para dar apoyo al análisis de datos.

• Herramientas para desarrollar el modelo de datos corporativo, los esquemas conceptual y lógico.

• Herramientas para desarrollar los prototipos de las aplicaciones.

• Con el uso de la herramienta CASE puede mejorar la productividad de aplicaciones de base de datos.

 HISTORIA

En la década de los setenta el proyecto ISDOS desarrollo un lenguaje llamado "Problem Statement Language" (PSL) para la solución de un problema informático en un diccionario automatizado. Era un producto de que analizaba los problemas y necesidades.

La primera herramienta era para PC llamada "Excelerator" en 1984, la oferta de herramientas es muy amplia como es el EASYCASE o WINPROJECT.

TECNOLOGÍA

La tecnología CASE es la automatización del desarrollo software para mejorar la calidad del sistema de información.

• Permitir aplicaciones prácticas de metodologías estructuradas, al ser realizadas con una herramienta consigue agilizar el trabajo.

• Facilitar la realización de prototipos y desarrollo conjunto de aplicaciones.

• Simplificar el mantenimiento de los programas.

• Mejorar y estandarizar la documentación

• Aumentar la PORTABILIDAD de las aplicaciones.

• Facilitar la reutilización de componentes software.

• Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de gráficos.

Herramientas de alto nivel, U-CASE (Upper CASE - CASE superior) o front-end, orientadas a la automatización y soporte de las actividades desarrolladas durante las primeras fases del desarrollo: análisis y diseño.

Herramientas de análisis y diseño. Permiten al desarrollador crear un modelo del sistema que se va a construir y también la evaluación de la validez y consistencia de este modelo. Proporcionan un grado de confianza en la representación del análisis y ayudan a eliminar errores con anticipación. Se tienen:

Herramientas de análisis y diseño (Modelamiento).

Herramientas de creación de prototipos y de simulación.

Herramientas para el diseño y desarrollo de interfaces. Máquinas de análisis y diseño. (Modelamiento)

 ERwin

PLATINUM ERwin es una herramienta de diseño de base de datos. Brinda productividad en diseño, generación, y mantenimiento de aplicaciones. Desde un modelo lógico de los requerimientos de información, hasta el modelo físico perfeccionado para las características específicas de la base de datos diseñada, ERwin permite visualizar la estructura, los elementos importantes, y optimizar el diseño de la base de datos.

Genera automáticamente las tablas y miles de líneas de stored procedure y triggers para los principales tipos de base de datos. ERwin hace fácil el diseño de una base de datos. Los diseñadores de bases de datos sólo apuntan y pulsan un botón para crear un gráfico del modelo E-R (Entidadrelación) de todos sus requerimientos de datos y capturar las reglas de negocio en un modelo lógico, mostrando todas las entidades, atributos, relaciones, y llaves importantes.

PowerDesigner

PowerDesigner es una suite de aplicaciones de Powersoft para la construcción, diseño y modelado de datos a través de diversas aplicaciones. Es la herramienta para el análisis, diseño inteligente y construcción sólida de una base de datos y un desarrollo orientado a modelos de datos a nivel físico y conceptual, que dan a los desarrolladores Cliente/Servidor la más firme base para aplicaciones de alto rendimiento

**Bibliografía:**

http://alfredohedzvega.blogspot.mx/2013/04/fundamentos-de-ingenieria-de-software.html