

Unidad III: Modelo de Análisis

3.1. Arquitectura de clases

El objetivo del modelo de análisis es crear una arquitectura de objetos que sirva como base para el diseño del sistema.

Dependiendo del tipo de aplicación existen varias arquitecturas. Ellas se distinguen según la organización de los objetos de acuerdo a su funcionalidad. Esto es llamado dimensión de arquitectura.

Dimensión de la arquitectura

- Unidimensional: un solo grupo de objetos para manejar la funcionalidad y la interacción externa.
- Bidimensional: un grupo de objetos para manejar la funcionalidad y otros para las interacciones externas.
- Tridimensional: La más usada en los sistemas de información que siendo el Modelo-Vista-Control.

Arquitectura Modelo-Vista-Control

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz del usuario y la lógica de negocio en tres componentes distintos. El modelo es el sistema de gestión de base de datos y la lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Modelo --> información

Vista --> presentación con el usuario

Control à comportamiento

3.2. Identificación de clases según Estereotipos

Para llevar a cabo la transición del modelo de requisitos al modelo de análisis se deben identificar los objetos necesarios para implementar todos los casos de uso. La arquitectura de objetos debe considerar los tres tipos de estereotipos de objetos. Para ello se deben identificar primero las clases borde, las clases entidad y finalmente las de Control. En general, los cambios más comunes a un sistema son los de funcionalidad y bordes. Los cambios de las interfaces deben afectar solo a los objetos borde.

Los cambios a la funcionalidad son más difíciles de administrar, ya que ésta puede abarcar todos los tipos de objetos.

Bordes

Toda la funcionalidad especificada en las descripciones de los casos de uso que depende directamente de los aspectos externos del sistema, se ubica en los objetos borde, pues a través de ellos se comunican los actores con el sistema. La tarea de una clase borde es traducir los eventos generados por un actor en eventos comprendidos por el sistema, y traducir los eventos del sistema en una presentación comprensible para el actor.

Las clases borde en otras palabras, describen la comunicación bidireccional entre el sistema y los actores.

Las clases borde son bastante fáciles de identificar, donde se cuenta con al menos tres estrategias:

1. Se pueden identificar con base a los actores.
2. Se pueden identificar con base en las descripciones de las interfaces del sistema que acompañan al modelo de requisitos.
3. Se pueden identificar con base en las descripciones de los casos de uso y extraer la funcionalidad específica a los objetos bordes.

Entidad

Se utilizan objetos entidad para modelar la información que el sistema debe manejar a corto y largo plazo. La información a corto plazo existe durante la ejecución de un caso de uso, mientras que la información a largo plazo trasciende los caso de uso, por lo que es necesario guardarla en alguna base de datos o archivos.

Durante la identificación de objeto entidad, se encontrara que objetos que objetos similares aparecen en varios casos de uso.

Control

En la mayoría de los casos de uso, existe un comportamiento que no se puede asignar de forma natural a ninguno de los otros dos tipos de objetos ya vistos. Una posibilidad es repartir el comportamiento entre los dos tipos de objetos, pero la solución no es buena si se considera el aspecto de extensibilidad. Un cambio en el comportamiento podría afectar varios objetos, dificultando su modificación. Para evitar estos problemas, tal comportamiento se asigna a objetos control. Es difícil lograr un buen balance en la distribución del comportamiento del caso de uso entre los objetos entidad, borde y control. Los objetos de control normalmente proveen a administración de los demás tipos de objetos.

3.3. Clases

Las clases son declaraciones o abstracciones de objetos, lo que significa, que una clase es la definición de un objeto. Cuando se programa un objeto y se definen sus características y funcionalidades, realmente se programa una clase. Una clase es un contenedor de uno o más datos (variables o propiedades miembro) junto a las operaciones de manipulación de dichos datos (funciones/métodos). Las clases pueden definirse como estructuras (struct), uniones (union) o clases (class) pudiendo existir diferencias entre cada una de las definiciones según el lenguaje. Además las clases son agrupaciones de objetos que describen su comportamiento

Clases

Las clases son lo más simple de Java. Todo en Java forma parte de una clase, es una clase o describe cómo funciona una clase. El conocimiento de las clases es fundamental para poder entender los programas Java.

Todas las acciones de los programas Java se colocan dentro del bloque de una clase o un objeto. Todos los métodos se definen dentro del bloque de la clase, Java no soporta funciones o variables globales. Esto puede despistar a los programadores de C++, que pueden definir métodos fuera del bloque de la clase, pero esta posibilidad es más un intento de no separarse mucho y ser compatible con C, que un buen diseño orientado a objetos. Así pues, el esqueleto de cualquier aplicación Java se basa en la definición de una clase.

Todos los datos básicos, como los enteros, se deben declarar en las clases antes de hacer uso de ellos. En C la unidad fundamental son los ficheros con código fuente, en Java son las clases. De hecho son pocas las sentencias que se pueden colocar fuera del bloque de una clase. La palabra clave `import` (equivalente al `#include`) puede colocarse al principio de un fichero, fuera del bloque de la clase. Sin embargo, el compilador reemplazará esa sentencia con el contenido del fichero que se indique, que consistirá, como es de suponer, en más clases.

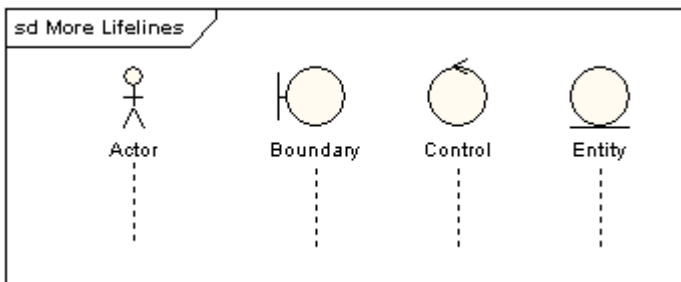
3.4. Diagramas de secuencias

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones. Los diagramas de secuencia no están pensados para mostrar lógicas de procedimientos complejos.

Línea de Vida

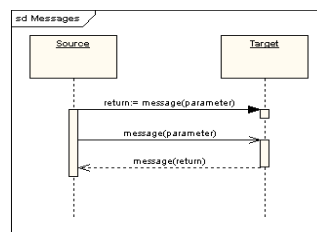
Una línea de vida representa un participante individual en un diagrama de secuencia. Una línea de vida usualmente tiene un rectángulo que contiene el nombre del objeto. Si el nombre es self entonces eso indica que la línea de vida representa el clasificador que posee el diagrama de secuencia.

Algunas veces un diagrama de secuencia tendrá una línea de vida con un símbolo del elemento actor en la parte superior. Este usualmente sería el caso si un diagrama de secuencia es contenido por un caso de uso. Los elementos entidad, control y límite de los diagramas de robustez también pueden contener líneas de vida.



Mensajes

Los mensajes se muestran como flechas. Los mensajes pueden ser completos, perdidos o encontrados; síncronos o asíncronos: llamadas o señales. En el siguiente diagrama, el primer mensaje es un mensaje síncrono (denotado por una punta de flecha oscura), completo con un mensaje de retorno implícito; el segundo mensaje es asíncrono (denotado por una punta de flecha en línea) y el tercero es un mensaje de retorno asíncrono (denotado por una línea punteada).

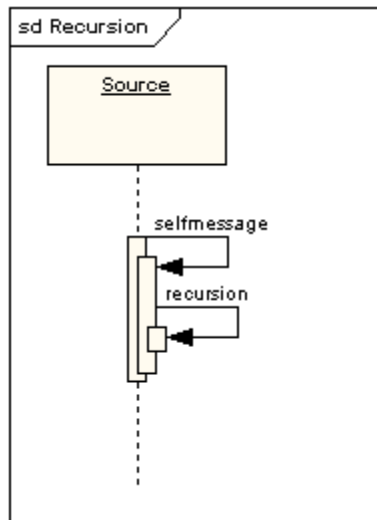


Ocurrencia de ejecución

Un rectángulo fino a lo largo de la línea de vida denota la ocurrencia de ejecución o activación de un foco de control. En el diagrama anterior hay tres ocurrencias de ejecución. El primero es el objeto origen que envía dos mensajes y recibe dos respuestas, el segundo es el objeto destino que recibe un mensaje asíncrono y retorna una respuesta, y el tercero es el objeto destino que recibe un mensaje asíncrono y retorna una respuesta.

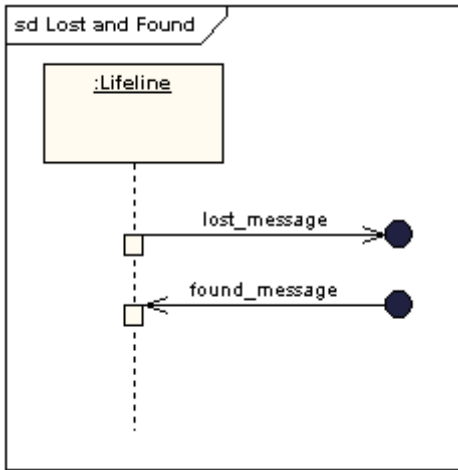
Mensaje Self

Un mensaje self puede representar una llamada recursiva de una operación, o un método llamando a otro método perteneciente al mismo objeto. Este se muestra como cuando crea un foco de control anidado en la ocurrencia de ejecución de la línea de vida.



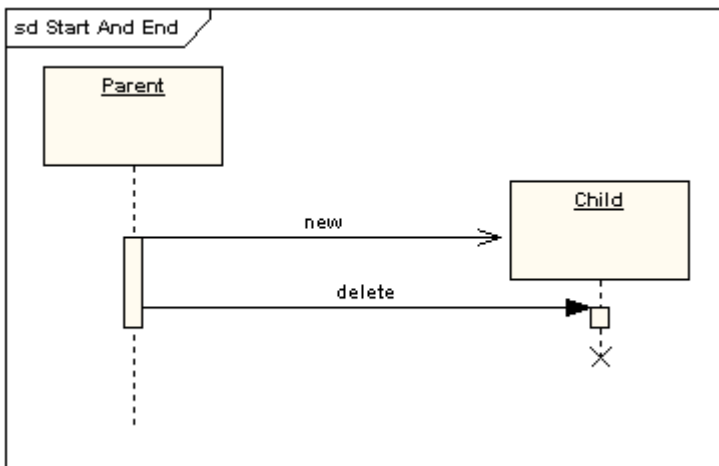
Mensajes perdidos y encontrados

Los mensajes perdidos son aquellos que han sido enviados pero que no han llegado al destino esperado, o que han llegado a un destino que no se muestra en el diagrama actual. Los mensajes encontrados son aquellos que llegan de un remitente no conocido, o de un remitente no conocido en el diagrama actual. Ellos se denotan yendo o llegando desde un elemento de punto final.



Inicio y final de línea de vida

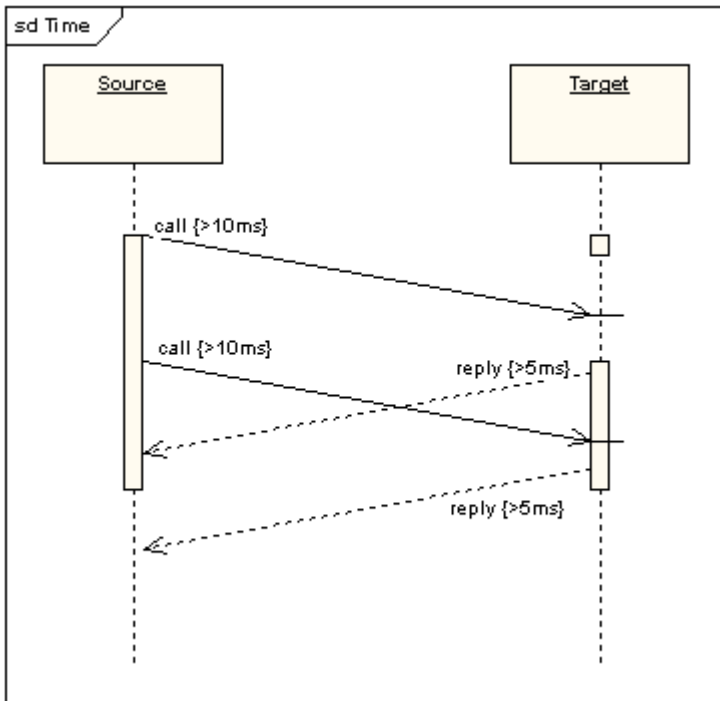
Una línea de vida se puede crear o destruir durante la escala de tiempo representada por un diagrama de secuencia. En el último caso, la línea de vida se termina por un símbolo de detención, representado como una cruz. En el primer caso, el símbolo al inicio de la línea de vida se muestra en un nivel más bajo de la página que el símbolo del objeto que causó la creación. El siguiente diagrama muestra un objeto que fue creado y destruido.



Restricciones de tiempo y duración

En forma predeterminada, un mensaje se muestra como una línea horizontal. Ya que la línea de vida representa el pasaje de tiempo hacia abajo, cuando se modela un sistema en tiempo real, o incluso un proceso de negocios en tiempo límite,

puede ser importante considerar el tiempo que toma realizar las acciones. Al configurar una restricción de duración para un mensaje, el mensaje se mostrará como una línea inclinada.



3.5. Diccionario de clases según Módulos

Un diccionario de clases es un catálogo, un depósito, de los elementos en un sistema. Como su nombre lo sugiere, estos elementos se centran alrededor de los datos y la forma en que están estructurados para satisfacer los requerimientos de los usuarios y las necesidades de la organización. En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos en todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El diccionario guarda los detalles y descripciones de todos estos elementos.

El diccionario se desarrolla durante el análisis de flujo de datos y auxilia a los analistas que participan en la determinación de los requerimientos de sistemas.

Importancia del diccionario

Los analistas utilizan los diccionarios de datos por cinco razones importantes:

1. Para manejar los detalles en sistemas grandes.
2. Para comunicar un significado común para todos los elementos del sistema.
3. Para documentar las características del sistema.
4. Para facilitar el análisis de los detalles con la finalidad de evaluar las características y determinar dónde efectuar cambios en el sistema.
5. Localizar errores y omisiones en el sistema.

Manejo de detalles

Los sistemas grandes tienen enormes volúmenes de datos que fluyen por ellos en forma de documentos, reportes e incluso pláticas. De manera similar, se llevan a cabo muchas actividades que utilizan los datos existentes o que generan nuevos detalles. Recuérdese, como se mencionó en la historia al inicio de este capítulo, que todos los sistemas experimentan cambios continuos y manejar de manera completa todos los detalles es un desafío. Con franqueza, es imposible que los analistas recuerden todo. Los que tratan de hacerlo cometen de manera invariable equivocaciones u olvidan elementos importantes. Los mejores analistas no intentan recordarlo todo, en lugar de hacerlo registran toda la información. Algunos lo hacen sobre hojas de papel y otros quizá sobre tarjetas indexadas. Muchos emplean para tal fin un procesador de palabras y una computadora personal por supuesto. Los analistas mejor organizados y más eficaces utilizan diccionarios de datos automatizados diseñados de manera específica para el análisis y diseño de sistemas. Comunicación de significados

Los diccionarios de datos proporcionan asistencia para asegurar significados comunes para los elementos y actividades del sistema. Si se examina una muestra de diagramas de flujo de datos para el procesamiento de pedidos, es probable que se tengan pocas dificultades para comprender qué datos

representan a la factura y al cheque. Los dos son términos comunes en el mundo de los negocios y muchas personas conocen su significado. Los diccionarios de datos registran detalles adicionales relacionados con el flujo de datos en el sistema de tal forma que todas las personas participantes puedan localizar con rapidez la descripción de flujos de datos, almacenes de datos o procesos.

3.6. Herramientas CASE para el análisis

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.