

**BASES CONCEPTUALES ACERCA DEL LENGUAJE UNIFICADO DE
MODELADO (UML) Y PATRONES DE DISEÑO.**

Sor Junny Londoño Rivera

Donaldo Andrés Beltrán Prieto

Instructor

Servicio Nacional de Aprendizaje-SENA

ANALISIS Y DESARROLLO DE SOFTWARE (2627038)

Regional Quindío.

2023

INTRODUCCIÓN

UML (Lenguaje Unificado de Modelado) comenzó a gestarse en 1994, cuando Jim Rumbaugh se unió a la compañía Rational Software Corporation fundada por Grady Booch, dos respetados investigadores en el área de desarrollo de notaciones para el análisis y diseño de sistemas orientados a objetos, cuya herramienta era Rational Rose.

El objetivo de los dos investigadores era unificar dos métodos que habían desarrollado: el método Booch, que describe conjunto de objetos y sus relaciones y el OMT (Object Modelling Tool) de Rumbaugh, orientada a objetos que combinaban notaciones provenientes de modelado orientado a objetos, a datos, a componentes y a flujos de trabajos. En esa misma época otro reputado investigador, Ivar Jacobson, cuya metodología es más centrada al usuario y a casos de uso (use case), se unió a Rational Software y se incluyeron ideas suyas. Estas tres personas son conocidas como los “tres amigos”. Construyeron lo que hoy conocemos como UML

Este lenguaje manejaba diversos métodos, técnicas y aspectos en común, pero utilizando diversas notaciones; inconvenientes para el aprendizaje, aplicaciones y uso de herramientas y además competía entre distintos enfoques, por lo tanto, debía establecer una notación estándar. Que se fue fomentando hasta lograr la aceptación de OMG (Grupo de administración de objetos) como notación estándar, el estándar líder en la industria para la programación de objetos distribuidos, quienes lo adoptaron para el análisis y el diseño orientado a objetos.

Es el primer método en publicar su propia notación, incluyendo la notación para la mayoría de la información de requisitos, análisis y diseño. El lenguaje de modelado unificado se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir. Y pretende unificar las experiencias pasadas sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

RESUMEN SOBRE UML

UML, se constituye como las técnicas que se emplean para la especificación de un sistema en cada una de sus fases y cuya sigla significa Lenguaje unificado de modelado (UML), es el lenguaje modelado más común y utilizado en la actualidad, teniendo como relevancia las acciones de visualizar y documentar los artefactos a tener en cuenta en un determinado sistema.

Se va conformando por diferentes gráficos, los cuales permiten la realización de algunos diagramas los cuales ayudan a comunicar una idea, entre ellos tenemos los siguientes:

Diagramas de casos de uso: representan a los actores y casos de uso (procesos principales) que intervienen en un desarrollo de software.

Diagramas de clases: Para UML una clase es una entidad, no una clase software. Un diagrama de clases UML puede ser un diagrama del dominio o representación de conceptos que intervienen en un problema, o también un diagrama de clases software.

Diagramas de secuencia: Suelen usarse para representar objetos software y el intercambio de mensajes entre ellos, representando la aparición de nuevos objetos de izquierda a derecha.

Diagramas de colaboración: Suelen usarse para representar objetos o clases y la forma en que se transmiten mensajes y colaboran entre ellos para cumplir un objetivo.

Diagramas de estados: Suelen usarse para representar cómo evoluciona un sistema (cómo va cambiando de estado) a medida que se producen determinados eventos.

Otros diagramas: Diagramas de actividad, diagramas de paquetes, diagramas de arquitectura software, etc.

En el desarrollo de UML, se deben tener en cuenta unas fases:

Análisis: Abarca las abstracciones primarias (clases y objetos) y mecanismos que están presentes en el dominio del problema.

Diseño: El resultado del análisis es expandido a una solución técnica. Se agregan nuevas clases que proveen de la infraestructura técnica: interfaces de usuario, manejo de bases de datos para almacenar objetos en una base de datos, comunicaciones con otros sistemas, etc.

Programación: En esta fase las clases del diseño son convertidas a código en un lenguaje de programación orientado a objetos.

Pruebas: Las pruebas de unidades se realizan a clases individuales o a un grupo de clases y son típicamente ejecutadas por el programador.

Los modelados permiten analizar y desarrollar software más complejos y eficaces, permitiendo obtener un modelo visual del sistema.

El lenguaje unificado de modelado es de gran importancia en la ingeniería, ayuda a modelar grandes sistemas y es muy utilizado en distintos tipos de software. El UML cada vez es más implementado, ya que ayuda a obtener un mayor entendimiento sobre las funcionalidades del software tanto para el desarrollador y para el cliente. Este modelado no solo permite diseñar, graficar, entre otros, sino que también facilita la construcción de los sistemas, a través de los diferentes modelos de estructurales y de comportamientos.

Estos diagramas son utilizados según la necesidad y el tipo de sistema que se vaya a desarrollar.

ELABORAR UN GLOSARIO DE TERMINOLOGÍA UML

Resumen: Un indicador aplicado a un clasificador (por ejemplo, actor, clase, caso de uso) o a algunas características de un clasificador (por ejemplo, las operaciones de una clase) que muestra que la característica está incompleta y no se pretende que se instancia, sino que se especializa por otras definiciones.

Clase abstracta: Una clase que no proporciona una declaración completa, quizás porque no tiene un método de implementación identificado para una operación.

Operación abstracta: a diferencia de los atributos, las operaciones de clase pueden ser abstractas, lo que significa que no se proporciona una implementación. Generalmente, una clase que contiene una operación abstracta debe marcarse como una clase abstracta. Una Operación debe tener un método suministrado en alguna Clase especializada antes de que pueda usarse.

Acción: Una acción es la unidad fundamental de especificación de comportamiento y representa alguna transformación o procesamiento en el sistema modelado, como invocar un método de una clase o una subactividad.

Activación: el tiempo durante el cual un objeto tiene un método en ejecución. A menudo se indica mediante una caja delgada o una barra superpuesta en la línea de vida del objeto en un diagrama de secuencia.

Diagrama de actividad: Diagrama que describe la lógica de procedimiento, el proceso empresarial o el flujo de trabajo. Un diagrama de actividades contiene una serie de actividades y están conectadas por flujos de control y flujos de objetos.

Clase activa: Una clase que define objetos activos.

Objeto activo: Un objeto que se ejecuta bajo su propio hilo

Actividad: Llevar a cabo el comportamiento en un diagrama de máquina de estados

Hacer: Un tipo de actividad que puede interrumpirse, a diferencia de las actividades normales que no pueden interrumpirse.

Interna: Una actividad que se ejecuta dentro de un estado

Entrada: Una actividad que se ejecuta cuando se ingresa a un estado

Salir: Una actividad que se ejecuta cuando se sale de un estado

Actor: un rol que asume un usuario cuando invoca un caso de uso.

Artefacto: Elementos que modelan piezas físicas de información en su sistema, como un manual del usuario, material de capacitación o archivo de contraseña.

Clase de asociación: Una clase que describe una asociación.

Asincrónico: El remitente de un mensaje asincrónico no espera una respuesta.

Atributo: Un dato significativo propiedad de una clase, que a menudo contiene valores que describen cada instancia de la clase. Además del nombre del atributo y una ranura para el valor del atributo, un atributo puede tener especificados Visibilidad, Tipo, Multiplicidad, Valor predeterminado y Cadena de propiedad.

Bloque: El bloque es un lugar donde todos los agregados se recolectan en un solo lugar

BPM - Business Process Model, se utiliza para definir el proceso que sigue el negocio. Es un diseño de alto nivel para el proceso empresarial.

BPMN - Notaciones de modelado de procesos de negocio, se utilizan para definir el proceso en papel con la ayuda de formas predefinidas de UML.

Cardinalidad: el número actual de ocurrencias de una propiedad. La cardinalidad debe ser un valor permitido por la multiplicidad

Clase: La construcción declarativa principal de la programación orientada a objetos; una unidad cohesiva de Atributos y Operaciones; una plantilla en tiempo de compilación para un objeto

Diagrama de clases: Tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos y las relaciones entre las clases.

Clasificador: Una categoría de elementos UML que tienen algunas características comunes, como atributos o métodos.

Colaboración: una colaboración es una sociedad de clases, interfaces y otros elementos que trabajan juntos para proporcionar un comportamiento cooperativo que es más grande que la suma de sus partes.

Componente: Un componente representa un módulo de software (código fuente, código binario, ejecutable, DLL, etc.) con una interfaz bien definida. La interfaz de un componente está representada por uno o varios elementos de interfaz que proporciona el componente. Los componentes se utilizan para mostrar las dependencias del compilador y del tiempo de ejecución, así como las dependencias de la interfaz y las llamadas entre los módulos de software. También muestran qué componentes implementan una clase específica.

Composición: Un tipo específico de relación que describe cómo un Objeto se compone de otro Objeto; una forma de agregación en la que el objeto secundario se destruye si el objeto principal se destruye.

Restricción: Lenguaje natural, lenguaje de programación o condición booleana del lenguaje de restricción de objetos que puede no ser falsa si una clase se considera válida

Contención: Contención por valor y contención por referencia. La contención por valor implica que un objeto contiene otro objeto; la contención por referencia implica que un objeto contiene un puntero a otro objeto.

Decisión: Un punto en un diagrama de actividad donde un flujo se divide en varios flujos protegidos mutuamente excluyentes. Una fusión marca el final del comportamiento opcional iniciado por la Decisión

Dependencia: Existe una dependencia entre dos elementos definidos si un cambio en la definición de uno da como resultado un cambio en el otro. En UML, esto se indica mediante una línea que apunta del elemento dependiente al independiente.

Propiedad derivada: Una propiedad que se puede calcular o inferir de otras propiedades.

Diagrama: Una representación visual de un subconjunto de características de un modelo UML

Dominio: Una agrupación lógica que declara explícitamente una regla que define la propiedad de los objetos en función de algún tipo o propiedad.

Encapsulación: Un medio para unir el código y los datos que utiliza. Es la base de la programación orientada a objetos.

Enumeración: Un conjunto de valores constantes para un nuevo tipo de datos

Evento: Cuando ocurre en un objeto, puede causar una transición en un diagrama de máquina de estado

Región de expansión: un conjunto de acciones en un diagrama de actividad que ocurren una vez para cada colección de tokens de entrada a la región de expansión.

Ampliar: un medio para utilizar una clase como punto de partida para la creación de una nueva clase. La nueva clase hereda todos los atributos, propiedades y métodos de la clase que extiende.

Estado final: El estado en el que un objeto deja de existir.

Flujo: Una conexión de navegación entre dos acciones

Flujo final: El punto en el que un flujo finaliza sin finalizar la actividad completa.

Bifurcación: Un punto en un diagrama de actividad donde un flujo de lógica se divide en varios flujos concurrentes (subprocesos)

Mensaje encontrado: Punto de partida para un diagrama de secuencia

CONCLUSIONES.

- UML, es un lenguaje modelado, no de programación.
- El objetivo de UML es ser capaz de describir el comportamiento de un sistema, subsistema u operación particular mediante un diagrama de secuencia el cual muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso, esto facilita como se distribuyen las tareas entre los componentes.
- Los diagramas a utilizar en las diferentes etapas del desarrollo de los sistemas de información, pueden variar dependiendo del tamaño y tipo de sistema, por lo que es necesario organizarlo según las fases del Proceso Unificado.

Referencias Bibliográficas

<file:///C:/Users/SOR%20LONDO%C3%91O/Downloads/Dialnet-LenguajeDeModelamientoUnificadoUMLParaModelamiento-5447353.pdf>