HISTORIA DEL UML

¿Qué es UML?

El Lenguaje Unificado de Modelado preescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML ofrece nueve diagramas en los cuales modelar sistemas.

- Diagramas de Casos de Uso para modelar los procesos 'business'.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema.

UML es una consolidación de muchas de las notaciones y conceptos más usadas orientados a objetos. Empezó como una consolidación del trabajo de Grade Booch, James Rumbaugh, e Ivar Jacobson, creadores de tres de las metodologías orientadas a objetos más populares.

En 1996, el Object Management Group (OMG), un pilar estándar para la comunidad del diseño orientado a objetos, publicó una petición con propósito de un metamodelo orientado a objetos de semántica y notación estándares. UML, en su versión 1.0, fue propuesto como una respuesta a esta petición en enero de 1997. Hubo otras cinco propuestas rivales. Durante el transcurso de

1997, los seis promotores de las propuestas, unieron su trabajo y presentaron al OMG un documento revisado de UML, llamado UML versión 1.1. Este documento fue aprobado por el OMG en Noviembre de 1997. El OMG llama a este documento OMG UML versión 1.1. El OMG está actualmente en proceso de mejorar una edición técnica de esta especificación, prevista su finalización para el 1 de abril de 1999.

2.1. UML ofrece notación y semántica estándar

UML preescribe una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos. Previamente, un diseño orientado a objetos podría haber sido modelado con cualquiera de la docena de metodologías populares, causando a los revisores tener que aprender las semáticas y notaciones de la metodología empleada antes que intentar entender el diseño en sí. Ahora con UML,diseñadores diferentes modelando sistemas diferentes pueden sobradamente entender cada uno los diseños de los otros.

2.2. UML no es un Método

Aun así, UML no preescribe un proceso o método estándar para desarrollar un sistema. Hay varias metodologías existentes; entre las más populares se incluyen las siguientes:

- *Catalysis*: Un método orientado a objetos que fusiona mucho del trabajo reciente en métodos orientados a objetos, y además ofrece técnicas específicas para modelar componentes distribuidos.
- *Objetory*: Un método de Caso de Uso guiado para el desarrollo, creado por Ivar Jacobson.
- *Shlaer/Mellor*: El método para diseñar sistemas de tiempo real, puesto en marcha por Sally Shlaer y Steven Mellor en dos libros de 1991, Ciclos de vida de Objetos, modelando el Mundo en Estados y Ciclos de vida de Objetos, Modelando el mundo en Datos (Prentice Hall). Shlaer/Mellor countinúan actualizando su método continuamente (la actualización más reciente es el OOA96 report), y recientemente publicaron una guía sobre cómo usar la notación UML con Shlaer/Mellor.
- *Fusion*: Desarrollado en Hewlett Packard a mediados de los noventa como primer intento de un método de diseño orientado a objetos estándar.

Combina OMT y Booch con tarjetas CRC y métodos formales. (www.hpl.hp.com/fusion/file/teameps.pdf)

- *OMT* : La Técnica de Modelado de Objetos fue desarrollada por James Rumbaugh y otros, y publicada en el libro de gran influencia "Diseño y Modelado Orientado a Objetos" (Prentice Hall, 1991). Un método que propone análisis y diseño 'iterative', más centrado en el lado del análisis.
- *Booch*: Parecido al OMT, y también muy popular, la primera y segunda edición de "Diseño Orientado a Objetos, con Aplicaciones" (Benjamin Cummings, 1991 y 1994), (Object-Oriented Design, With Applications), detallan un método ofreciendo también diseño y análisis 'iterative', centrándoso en el lado del diseño.

Además, muchas organizaciones han desarrollado sus propias metodologías internas, usando diferentes diagramas y técnicas con orígenes varios. Ejemplos son el método Catalyst por Computer Sciences Corporation (CSC) o el Worlwide Solution Design and Delivery Method (WSDDM) por IBM. Estas metodologías difieren, pero generalmente combinan análisis de flujo de trabajo, captura de los requisitos, y modelado de negocio con modelado de datos, con modelado de objetos usando varias notaciones
(OMT, Booch, etc), y algunas veces incluyendo técnicas adicionales de modelado de objetos como Casos de Uso y tarjetas CRC. La mayoría de estas organizaciones están adoptando e incorporando el UML como la notación orientada a objetos de sus metodologías.

Algunos modeladores usarán un subconjunto de UML para modelar 'what they're after', por ejemplo simplemente el diagrama de clases, o solo los diagramas de clases y de secuencia con Casos de Uso. Otros usarán una suite más completa, incluyendo los diagramas de estado y actividad para modelar sistemas de tiempo real, y el diagrama de implementación para modelar sistemas distribuidos. Aun así, otros no estarán satisfechos con los diagramas ofrecidos por UML, y necesitarán extender UML con otros diagramas como modelos relacionales de datos y 'CRC cards'.

2.3. Extensiones UML 1.1

Los mecanismos de de extensibilidad incorporados permiten a UML ser una especie de especificación abierta que puede cubrir aspectos de modelado no especificados en el documento 1.1. Estos mecanismos permiten extender la notación y semántica de UML.

2.3.1. Estereotipos

Los estereotipos son el mecanismo de extensibilidad incorporado más utilizado dentro de UML. Un estereotipo respresenta una distinción de uso. Puede ser aplicado a cualquier elemento de modelado, incluyendo clases, paquetes, relaciones de herencia, etc. Por ejemplo, una clase con estereotipo 'actor' es una clase usada como un agente externo en el modelado de negocio. Una clase patrón es modelada como una clase con estereotipo parametrizado, lo que significa que puede contener parámetros.

2.3.2. Extensiones de Modelado de Negocio

Un documento separado dentro de la especificación UML define clases y estereotipos de asociación específicos que extienden UML hasta cubrir conceptos de modelado de negocio. Esto incluye 'stereotyping' una clase como un actor, un trabajador ('both internal and case'), o una entidad, y 'stereotyping' una asociación como una comunicación simple, o una subcripción entre un origen y un objetivo.

2.3.3. Lenguaje restrictivo (constraint) de objetos (OCL)

Una imagen puede describir muchas palabras. De igual modo, un modelo gráfico puede describir una cierta parte del comportamiento, después de la cual es necesario rellenar detalles adicionales con palabras. Describiendo algo con palabras, sin embargo, casi siempre desemboca en ambiguedades; por ejemplo, "¿que quería decir cuando escribió eso?". El Lenguaje Restrictivo (constraint) de Objetos (OCL) está incorporado en UML como un estándar para especificar detalles adicionales, o precisar detalles en la estructura de los modelos.

Desarrollado dentro de la IBM Insurace Division como un lenguaje de modelado de negocio, el OCL es un lenguaje formal diseñado para ser fácil de leer y de escribir. OCL es más funcional que el lenguaje natural, pero no tan preciso como un lenguaje de programación - no puede ser usado para escribir lógicas de lógica de programación o control de flujo. Puesto que OCL es un lenguaje para la expresión pura, sus declaraciones están garantizadas de no tener efectos laterales - simplemente transportan un valor y nunca pueden cambiar el estado del sistema.

2.4. Más Extensiones

Dos áreas específicas que UML no cubre actualmente, ni con sus extensiones, son análisis guiados por la responsabilidad y modelado de bases de datos relacionales. Esta guía introduce estas técnicas como extensiones actuales del mundo real para UML que se deberían tener en cuenta.

2.4.1. Análisis guiados por la responsabilidad con tarjetas CRC

Una técnica muy usada para hacerse a la idea de cómo hay que pensar trantando con orientación a objetos son los análisis guiados por la responsabilidad con las tarjetas CRC (CRC - Colaborador y Responsabilidad de Clase). Con esta técnica, las clases descubiertas durante el análisis pueden ser filtradas para determinar qué clases son realmente necesarias para el sistema.

2.4.2. Modelo Relacional de datos

Aunque las bases de datos orientadas a objetos se están volviendo más populares, en el entorno de desarrollo actual, la base de datos relacional sigue siendo el método predominante para almacenar datos. Los diagramas de clases de UML se pueden usar para modelar la base de datos relacional en la que el sistema está basado, sin embargo, los diagramas tradicionales de modelado de datos capturan más información sobre la base de datos relacional y son más adecuados para modelarla. Esta guía trata el uso de Diagramas de Relaciones de Entidad (ER) como una extensión importante de UML para el modelado de bases de datos relacionales.