

Modelado Visual con UML

Resumen

El documento ofrece una perspectiva de UML, resaltando la importancia del modelado visual en el desarrollo de software, como una forma de mejorar la problemática de elaboración de software de calidad en las empresas. El modelado visual es independiente del lenguaje de programación y ofrece diferentes vistas dependiendo del el aspecto del sistema que se quiera resaltar.

Introducción

En la década de los ochenta, la programación orientada a objetos comenzó a entrar a las empresas, pero es a hasta finales de la década de los 80 cuando surgen infinidad de metodologías de análisis y diseño orientado a objetos. Es decir, el análisis que existía anteriormente - análisis estructurado- no se adecuaba del todo a la POO.

Aparecieron algunas propuestas apoyadas por su correspondiente bibliografía. Citaremos las principales:

- Sally Shlaer y Steve Mello. Publican en 1989 y 1991 sobre análisis y diseño. Su enfoque se conoce como Diseño Recursivo.
- Peter Coad y Ed Yourdon en 1991 escribieron con un enfoque hacia los métodos ligeros orientados a prototipos de Coad.
- La comunidad de Smalltalk de Portland, Oregon, aportó el Diseño Guiado por la Responsabilidad (*Responsibility-Driven Design*) en 1990 y las tarjetas de Clase-Responsabilidad-Colaboración (*Class-Responsibility-Collaboration*) conocidas como tarjetas CRC, en 1989.
- Grady Booch trabajando para Rational Software, desarrollando sistemas en Ada. 1994 y 1995.
- Jim Rumbaugh trabajando en laboratorios de investigación de General Electric. Publica un libro de la Técnica de Modelado de Objetos (*Object Modeling Technique*) en 1991. Su metodología, conocida como

OMT avanza hasta una segunda versión (OMT-2) en 1996.

- Ivar Jacobson trabajando para Ericsson en conmutadores telefónicos, introdujo el concepto de Casos de Uso (*Use Cases*) en 1994.

El caso es que existían múltiples metodologías, cada una con su grupo de seguidores. Muchas se parecen entre sí, con distintas notaciones o términos distintos.

Es en la OOPSLA '94 donde se conoce que Rumbaugh deja a General Electric para trabajar junto con Booch en Rational Software, con la intención de unificar sus métodos. En el mismo evento en 1995, se anuncia que Rational Software había comprado Objectory y que Jacobson se uniría a su equipo de trabajo.

En 1996 Grady Booch, Jim Rumbaugh e Ivar Jacobson (“los tres amigos”) proponen su método al que le llaman *Unified Modeling Language (UML, Lenguaje Unificado de Modelado)*.

En conclusión, **UML** unifica esencialmente los métodos de Booch, Rumbaugh (OMT) y Jacobson.

Modelado Visual

El Modelado Visual es el modelado de una aplicación usando notaciones gráficas. Pero, ¿qué tan importante es construir el modelo de una aplicación? En este sentido comúnmente se hace la comparación hacia la arquitectura tradicional, en la construcción de casas. Aún cuando la construcción que se planea hacer sea una casa sencilla, el resultado será más satisfactorio si cuenta con todo un respaldo en un correcto diseño.

Booch compara la construcción de software con la construcción de una casa para un perro, de una casa para tu familia y de un gran edificio [Booch, 1999]. En el primer caso no será tan evidente la falta de un buen diseño, o al menos nuestro perro no se quejará demasiado. En el segundo caso, es humanamente posible hacer la construcción de una casa sin los planos adecuados, pero la casa resultante seguramente tendrá varias carencias, o en un peor caso, posiblemente no resistirá ciertas condiciones extremas como un temblor. En el caso del edificio, definitivamente sería un grave error comenzar la construcción sin los estudios y planos adecuados.

En el caso del software, es curioso como muchos proyectos del tamaño de un rascacielos, son construidos como si se tratara de la casa de un perro. El mode-

lado visual del software ayuda a capturar las partes esenciales de un sistema:

- Se utiliza para capturar los procesos de negocios desde la perspectiva del usuario.
- El Modelado Visual se utiliza para analizar y diseñar una aplicación, distinguiendo entre los dominios del negocio y los dominios de la computadora.
- Ayuda a reducir la complejidad.
- El Modelado Visual se realiza de manera independiente al lenguaje de implementación.
- Promueve el reuso de componentes.

UML, Lenguaje de Modelado Unificado

Es un **lenguaje de modelado**¹ y es independiente del proceso, por lo que no se considera una metodología. Un lenguaje de modelado define la notación que es utilizada por los métodos para representar los diseños.

El método o proceso sería los pasos a seguir para llevar a cabo el análisis y diseño. UML debe en parte su gran aceptación a que no incluye un proceso como parte de su propuesta.

En la figura 1, es posible apreciar el conjunto de diagramas que componen la notación de modelado visual de UML, cada una de ellas con su propia sintaxis; pero al mismo tiempo, compartiendo aquellos elementos que llegan a ser comunes a diversos diagramas. Por ejemplo, la notación general de una clase en cualquier diagrama de UML es un rectángulo, aunque puedan variar algunas características o el nivel de detalle dependiendo del contexto en que se este presentando.

FIGURA 1. DIAGRAMAS DE UML [JACOBSON, 1999]

¹ Un modelo es una descripción completa de un sistema desde una perspectiva particular.