

QUÉ ES 'UML' Y PARA QUÉ SIRVE?

UNIVERSIDAD IBEROAMERICANA

ASIGNATURA: METODOLOGÍA DE LA PROGRAMACIÓN

PROFESOR: MTRO. ALFONSO GREGORIO RIVERO DUARTE

TAREA 1

NOMBRE: LUIS MANUEL HERNÁNDEZ TAPIA

MATRÍCULA: 206749-8

31 DE AGOSTO DE 2017

Investigación

Abstract—Existen 2 formas de desarrollar un diseño de software: Una es hacerla tan simple que obviamente no hay deficiencias, y la otra es que sea tan complicada que no existan deficiencias obvias. El primer método es mucho más difícil -C.A.R. Hoare

I. DEFINICIÓN

EL UML (Unified Modeling Language) por sus siglas en inglés o Lenguaje Unificado de Modelado es un Lenguaje como Pascal, C, el español, francés y el latín; probablemente es uno de los lenguajes más recientes, nació en enero de 1997, aunque sus primeros pasos se registran desde 1994. Como los demás lenguajes su origen se debió a la necesidad de programar de manera diferente pero usando como todos símbolos para transmitir significado.

UML (Unified Modeling Language) fue definido por Booch, Rumbaugh y Jacobson (2005), como un lenguaje de propósito general usado para especificar, visualizar y construir un modelo o pieza de información producido en el proceso de desarrollo de software. Dado que UML es un lenguaje de

propósito general, la descripción de dominios de aplicación específicos requiere de la definición de un nuevo lenguaje que describe el metamodelo, utilizando el estándar MOF (OMG, 2006b), o la extensión propiamente de UML a través de un mecanismo denominado Pírfiles UML.

II. PÉRFILES

Los pírfiles UML proveen un mecanismo de extensión genérico para construir modelos UML en dominios particulares. Están basados en estereotipos, restricciones y valores etiquetados adicionales que son aplicados a los elementos o relaciones de un diagrama:

1. Un estereotipo es un nuevo elemento del lenguaje extendido que se define sobre algún elemento pre-existente de UML. Gráficamente, los estereotipos se denotan entre comillas francesas (stereotype) dentro de cajas.
2. Los estereotipos pueden tener restricciones que imponen condiciones sobre los elementos estereotipados. Para la especificación

de restricciones puede utilizarse el lenguaje natural o un lenguaje formal como OCL (Object Constraint Language) propuesto por OMG (2006a).

3. Los valores etiquetados son meta-atributos asociados a una metaclase extendida por el perfil. Un valor etiquetado se caracteriza por un nombre y un tipo, y se representa como un atributo de la clase que define el estereotipo.

III. CARACTERÍSTICAS

EL UML es un modelo para la construcción de software orientado a objetos como standard de ISO propuesto por la OMG.

Consta de un conjunto de tipos de diagramas interrelacionados, dentro de los cuales se utilizan elementos del modelo, que sirven para describir distintos aspectos de la estructura y dinámica del software

Los 9 diagramas que forman la base de UML, y dictan la manera en que es diseñado un sistema:

1. Uso-Caso
2. Clases
3. Objetos
4. Secuencia
5. Colaboración
6. De Estado (Statechart)
7. Actividad
8. Componentes
9. Ejecución (Deployment)

IV. RESPONSABILIDAD DEL UML

El Object Management Group (OMG), creado en 1989, es una organización no lucrativa en la cual participan más de ochocientas grandes empresas de software, de hardware, usuarias y consultoras, y tiene la finalidad de fomentar el uso de la tecnología de objetos e impulsar la introducción de software orientado a objetos que ofrezca reusabilidad, portabilidad e interoperabilidad en entornos distribuidos heterogéneos. El medio con que el OMG intenta conseguir sus objetivos es la elaboración de estándares, para los cuales acepta propuestas.

En cambio, no produce software ni elabora especificaciones de implementación o funcionalidad. El otro estándar del OMG, además del UML, que

ha elaborado el OMG es CORBA, sobre objetos distribuidos, cuyas implementaciones tienen una expansión rápida.

V. DESCRIBIENDO UML

Como sabemos, un modelo es una colección de imágenes y texto que representa algo. Ese algo es lo que se aterriza en software a través de un lenguaje. En palabras de algunos escritores como Paul Kimmel: Los modelos son valiosos porque es más fácil para dibujar algunas imágenes sencillas que escribir código o el mismo texto que describa lo mismo.

El UML comprende símbolos y una gramática que define la manera en la que se pueden usar estos símbolos.

El UML comprende un cierto número de diagramas interrelacionados mediante conceptos comunes. Sólo para describirlos, los consideraremos agrupados en tres modelos: 1. Estático Dinámico describe la estructura de clases y objetos. 2. Dinámico. (o modelo de comportamiento), describe las interacciones entre los objetos dentro del software. 3. Implementación, describe la estructura del software en cuanto a los componentes de que consta y su ubicación.

VI. EJEMPLOS

Ejemplo 1: Un sistema que gestione un conjunto de escaleras mecánicas de una gran superficie comercial. Cada escalera se identifica por un código. En un momento determinado, una escalera puede estar en funcionamiento o en reparación. Independientemente de esto, una escalera puede estar subiendo, bajando o parada, pero si está en reparación está, necesariamente, parada. Si una escalera está en reparación, el sistema guarda cuál era el estado (subiendo, bajando, parada) que tenía la escalera en el momento de pasar a reparación.

En este ejemplo se pueden encontrar las necesidades de describir la estructura de clases de las que se están hablando así como de los objetos que participan (escaleras, funcionamiento, dirección y mantenimiento). De igual manera, las interacciones de los objetos.

Ejemplo 2: Los datos de un sistema que contiene sólo una relación ternaria R entre las entidades A, B y C. Sean a, b y c ocurrencias cualesquiera de las entidades A, B y C, respectivamente. Indica

cómo se deberán expresar en este modelo las restricciones siguientes: 1. Ningún c puede participar en ms de 10 ocurrencias de R. 2. Una pareja a,b cualquiera puede estar relacionada, vía R, con un máximo de 3 ocurrencias de C. 3. Un trío a,b,c cualquiera puede estar relacionado, vía R, como máximo una vez.

Ejemplo 3: Los datos de un sistema que contiene sólo una relación ternaria R entre las entidades A, B y C. Sean a, b y c ocurrencias cualesquiera de las entidades A, B y C, respectivamente. Indica cmo se deberían expresar en este modelo las restricciones siguientes: 1. Todos los c han de participar como mínimo en dos ocurrencias de R. 2. Una pareja a,b cualquiera ha de estar relacionada, vía R, con un mínimo de 3 ocurrencias de C. 3. Una ocurrencia c cualquiera ha de estar relacionada como máximo con tres ocurrencias distintas de B. 4. Un trío a,b,c cualquiera puede estar relacionado, vía R, como máximo una vez, y como mínimo ninguna.

VII. IMPORTANCIA

El UML (Ünified Modeling Language”) está consolidado como el lenguaje base para el análisis y diseño de sistemas de cómputo. Con este lenguaje es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software previo al proceso intensivo de escribir código. En lenguaje cotidiano se puede decir que en el caso de la construcción de una casa deben de realizarse los planos previos a la edificación, en Software se deben realizar diseños en UML previa codificación de un sistema.

VIII. CONCLUSIONES

Así como todo algoritmo requiere de un análisis y elaboración previa; el UML bien puede considerarse la antesala de la creación o el montaje de algún software. El UML puede otorgar eso: la visión previa a 'cómo' puede funcionar un sistema.

Por otro lado, la planificación de todo 'problemaó ñecesidad' permitirá que durante el desarrollo y la creación final del mismo tenga un mejor resultado siempre enfocado a un resultado efectivo, confiable y rápido. Características que pueden permitir la prolongación de ciclo de vida de cualquier software.

REFERENCIAS HEMEROGRÁFICAS

- Campderrich Falgueras, Benet. Ingeniería del software. Barcelona, ES: Editorial UOC, 2003. ProQuest ebrary. Web. 31 August 2017. Copyright 2003. Editorial UOC. All rights reserved.
- Kimmel, Paul. Manual de UML. México, D.F., MX: McGraw-Hill Interamericana, 2008. ProQuest ebrary. Web. 30 August 2017.
- Rodríguez, Rosseline, and Goncalves, Marlene. Perfil UML para el modelado visual de requisitos difusos. Caracas, VE: Red Enlace, 2009. ProQuest ebrary. Web. 31 August 2017. Copyright 2009. Red Enlace. All rights reserved.
- Teniente López, Ernest, Costal Costa, Dolors, and Sancho Sams, M. Ribera. Especificación de sistemas software en UML. Barcelona, ES: Universitat Politecnica de Catalunya, 2003. ProQuest ebrary. Web. 31 August 2017. Copyright 2003. Universitat Politecnica de Catalunya. All rights reserved.
- Enl@ce: Revista Venezolana de Información, Tecnología y Conocimiento Año 6: No. 3, Septiembre-Diciembre 2009, pp. 29-46 Perfiles UML