

Los Diagramas de Unified Modeling Language (UML) para el Mejoramiento de los Requerimientos Funcionales en el Desarrollo de Aplicaciones

Msc. Jairo Augusto Cortes Mendez¹, Mcs. Jaime Alberto Páez Páez², Dr. Fredys Alberto Simanca Herrera³

Resumen: El proyecto propone la identificación y relación entre los requerimientos y los diagramas de UML para determinar en forma temprana posibles errores en su concepción. El fin último es soportar estas métricas sobre una herramienta que apoye la labor del diseñador y desarrollador de software valorando las relaciones que deben tener los diagramas y su calidad en términos de estándares. Para esto, la investigación que hace parte de la convocatoria interna de investigación de la Universidad Cooperativa de Colombia se basa en los estándares de la ISO, la IEEE asociados con los procesos y productos de software. El proyecto busca generar métricas que apoyen la labor de los docentes investigadores y estudiantes en el desarrollo de software que utilicen adecuadamente para su proceso de abstracción de la realidad los diagramas de Unified Modeling Language (UML). Se determinó que UML es fundamental a la hora del desarrollo, por lo que se indagara en su importancia como herramienta, exponiendo cada una de las funcionalidades que brinda y como facilita el desarrollo de un software, para luego ver sus respectivas aplicaciones dentro de lo establecido por la International Organization for Standardization - ISO.

Palabras clave: Calidad del Software, Diagramas de UML, Métricas del Software, Desarrollo de Software, Buenas prácticas.

Introducción

Para considerar bien hecho un diagrama UML (Lenguaje Unificado de Modelado) se debe considerar que tengan implementadas estándares de calidad correctos y para garantizar estos estándares, se deben seguir ciertos modelos o métricas de calidad para considerar un software eficiente con respecto a sus resultados mediante el estudio de sus atributos. Igualmente existen ciertas herramientas para determinar la calidad las cuales se enfocan a generar matrices y diagramas de comparación para después realizar el análisis sobre estos y no determinar si hay o no fallos en elaboración. Teniendo esto en cuenta, se realizará una investigación que se centra en realizar reingeniería sobre las reglas, códigos y pautas sobre la base de los diagramas UML ya sean realizados a mano o generados por una aplicación además también se realizara un estudio adecuadamente sobre los estándares de la Object Management Group (OMG) sobre UML integrado a XML-XMI para hacer una descripción de estos modelos en OCL (Object Constraint Language) y brindar así una investigación que amplíe la base de medición de estos diagramas y utilice una herramienta que le permita predecir el error en su elaboración en forma temprana.

La investigación busca determinar la relación existente entre el requerimiento funcional y los diagramas de Unified Modeling Language (UML). Este un lenguaje grafico para visualizar, especificar, construir y documentar los artefactos de un sistema de software de cualquier tipo e intensidad. Igualmente es un estándar para describir las bases de un sistema, incluyendo aspectos conceptuales como procesos de negocio y funciones del propio sistema, al igual que permite describir aspectos concretos como declaraciones del tipo de lenguaje de programación que se utiliza, esquemas de bases de datos y componentes de software de valor reutilizable.

Igualmente, este lenguaje ayuda a especificar, visualizar, construir y documentar los distintos tipos de arquitectura de software, y también poder ser usado en el modelado de negocios.

Una de las metas principales del lenguaje UML es poder hacer abstracción del mundo real y convertirse en una representación gráfica interoperable para modelar objetos; igualmente permite intercambiar información entre los distintos modelos y las distintas herramientas que se proveen, es estrictamente necesario tener un acuerdo entre semánticas y notaciones de los mismos, por lo que UML cumple con:

- Representación formal de un meta modelo de análisis y diseño de objetos comunes para representar distintos tipos de modelos, como los estáticos, de comportamiento, de uso y arquitectónicos

¹ El Msc. Jairo Augusto Cortés Méndez es Profesor Investigador del programa de Ingeniería de Sistemas de la Universidad Cooperativa de Colombia, Bogotá, Colombia. jairo.cortes@campusucc.edu.co (autor corresponsal)

² El Msc. Jaime Alberto Páez Páez es Profesor Investigador del programa de Ingeniería de Sistemas de la Universidad Cooperativa de Colombia, Bogotá, Colombia. jaime.paez@campusucc.edu.co

³ El Dr. Fredys Alberto Simanca Herrera es Profesor Investigador del programa de Ingeniería de Sistemas de la Universidad Cooperativa de Colombia, Bogotá, Colombia. fredys.simanca@campusucc.edu.co

- Especificaciones IDL (Interactive Data Language) para mecanismos de intercambio de modelos entre herramientas de análisis y diseño orientado a objetos.
- Una notación legible por humanos para representar modelos de análisis y diseño orientado a objetos. Este documento define la notación UML, una elegante sintaxis gráfica para expresar constantemente la rica semántica de UML.

El lenguaje de modelado unificado proporciona lo siguiente:

- Semántica y notación para estudiar los problemas de modelado contemporáneo de manera directa y económica.
- Semántica para abordar ciertos problemas de modelado futuros esperados, específicamente relacionados con la tecnología de componentes, la informática distribuida, los marcos y la capacidad de ejecución.
- Mecanismos de extensibilidad para que proyectos individuales puedan extender el meta modelo para su aplicación a bajo costo, sin cambiar directamente el meta modelo UML.
- Mecanismos de extensibilidad para que los futuros enfoques de modelado puedan crecer sobre el UML.
- Semántica para facilitar la realización de modelos y herramientas.
- Semántica que representa la interfaz de los repositorios para compartir y almacenar artefactos modelo.

La presente investigación se enfocará al estudio de las métricas y estándares de calidad de los diagramas de UML ya que al tener una guía de como calificar los criterios de un software es más sencillo tener un área de desarrollo más eficiente. Al tener en cuenta que los desarrollos son homogéneos y comparables, es más fácil tener una optimización de recursos y estimados monetarios si no se cumple errores y se tiene una guía adecuada de desarrollo.

La pregunta que orienta el proceso de investigación es: ¿Cómo mediante un sistema de almacenamiento de las métricas de calidad del software se puede determinar fallos en su elaboración?

Desarrollo

Para el desarrollo del proyecto, se utilizó una metodología general como es el método de ingeniería y en forma específica lo relacionado con las arquitecturas orientadas a modelos - MDA de la OMG.

Observar Soluciones existentes

Inicialmente se observaron las soluciones existentes mediante una revisión de las investigaciones que han propuesto metodologías, modelo y métricas para valorar los diagramas de UML y su formato XMI propuesto por la OMG. Con esta valoración, se determina las métricas asociadas a los diagramas de UML. Las métricas de calidad del software han evolucionado mediante los estándares de procesos, productos y recursos.

Existen las métricas tradicionales como la Complejidad Ciclomática, la métricas de McCabe y Mc Call, están las métricas orientadas a objetos como MOOSE, MOOD y las de Loren y Kidd definidas a nivel de diseño y las métricas para diagramas de UML que se definen más a nivel conceptual.

Las Métricas MOOSE o conocidas por CK Chidamber y Kemerer son métricas orientadas a clases: clases individuales, herencia y colaboraciones. Es uno de los conjuntos de métricas más referenciado.

Las Métricas de Lorenz y Kidd orientadas a clases miden el tamaño, herencia, valores internos y externos.

Las Métricas de diagramas UML o conceptuales divididas según el diagrama entre los más utilizados están el diagrama de clases, casos de uso y transición y estados. (Piattini et al, 2008, pp 68-119)

Los diagramas de UML son variados por nombrar algunos tenemos los diagramas de casos de uso, los diagramas de clase, de componentes, de objetos, de despliegue, de paquetes, de actividades y de estados.

Proponer la mejor solución

Luego del anterior análisis de las soluciones se propone la mejor, en esta fase se diseña la solución con respecto al formato XMI de los diagramas UML, se describen las características y variables asociadas a los diagramas para validarlas con las métricas propuestas en la etapa inicial. Para el diseño se utiliza la Arquitectura orientada por Modelos - MDA de la OMG.

Se utilizan varios modelos previos al desarrollo, los MDA es un marco arquitectónico abierto permitiendo generar modelo independientes partiendo de modelos específicos. Se utilizan en esta fase tres modelos:

CIM(Computationally-indepenent Model): se establece el dominio del problema y requisitos del sistema

PIM(Platform-Idendent Models), se describe la solución de software a partir del análisis y diseño

PSM(Platform-Specific Models), se detalla la plataforma que implementara la solución de los metadatos.

En esta fase se determinan diversas métricas determinadas de varios estudios y que son la base de esta investigación, para esto analizaremos cada diagrama con sus métricas en general.

El proceso general de la medición están relacionados con la supervisión, las predicciones, el control y prueba de cada una de las etapas del desarrollo y mantenimiento de proyectos de software. El proceso de medición inicia desde el tamaño de las aplicaciones por sus líneas de código asociado con las horas de programación y costos por hora. En este proceso se determina la productividad de los programadores y las proyecciones en el tiempo. Con esto se determina varias formas de medir o de obtener métricas utilizando funciones o modelos de análisis, la forma de contar las líneas de código y calcular la complejidad de los programas. La investigación de este proyecto se fundamenta en determinar estas medidas como por ejemplo para las bases de datos relacionales es fundamental el análisis de su actividad y volatilidad para determinar adecuadamente las llaves foráneas y primarias. Algunas medidas pueden estar asociadas al porcentaje de llaves foráneas (KF) en una tabla calculado entre la relación entre el número de atributos (NA) de una tabla sobre el número de llaves foráneas (NKF), la expresión es $KF = NKF/NA$, para un estudio más profundo se puede remitir a los autores Rodríguez y otros en el libro sobre la medición en la Ingeniería del Software (Rodríguez, et al, 2000)

Otras medidas clásicas son el tamaño expresado como líneas de código fuente(LOC), las que establece McCabe mediante su complejidad ciclomática(V(G)) expresado como $V(G)= A-N+2$ donde A es el número de aristas del grado, N el número de nodos.

Las clases tienen una estructura derivada del paradigma orientado a objetos, donde cada objeto está conformado por atributos, métodos y asociaciones propuestas por Merchesi(Marchesi, 1998), existen otras como las de Genero(Genero, et al, 2000) divide las clases en métricas de asociación, agregación y dependencia como se aprecia en el cuadro 1.

Tipo de Diagrama	Métricas asociadas
Diagrama de Clases	Cantidad de atributos, cantidad de métodos, cantidad de asociaciones, longitud de profundidad, longitud de anchura, cantidad de nodos u objetos, número de hijos, Métodos ponderados por clase(MPC), acoplamiento entre objetos, respuestas de clase.

Cuadro 1. Métricas para diagramas de Clase

Para diagramas UML como los casos de uso se utiliza el método de puntos de caso de uso de Gustav Karner en el año 1993 fundamentados en el método de punto de función. Este consta de cuatro etapas y se derivan algunos factores de medida como se aprecia en el cuadro 2.

Tipo de Diagrama	Métricas asociadas
Diagrama de Casos de uso	Cantidad de actores y casos de uso, número de transacciones, numero de interacciones entre el usuario y sistema, número de includes y extends.

Cuadro 2. Métricas de Casos de Uso.

Para los diagramas de Secuencia tienen una correspondencia con los casos de uso, estos consisten en el envío de mensajes entre objetos y actividades con el fin de cumplir el objetivo de cada caso de uso. Están compuestos de varios objetos ubicados en forma horizontal y otra vertical que representa el tiempo de las acciones. Los objetos se pueden enviar mensajes así mismo y entre objetos y estos a su vez avanzan en el tiempo, algunas métricas son como las mostradas en el cuadro 3.

Tipo de Diagrama	Métricas asociadas
Diagrama de Secuencia	Cantidad de objetos, cantidad de mensajes, Cantidad de relaciones de objetos entre mensajes, cantidad de unidades de tiempo entre objetos.

Cuadro 3. Métricas de diagramas de Secuencia.

Los diagramas de interacción están conformados por objetos, relaciones y mensajes, se caracteriza por el orden temporal de los mensajes (Booch, et al, 1999) y sus métricas están relacionadas en el cuadro 4.

Tipo de Diagrama	Métricas asociadas
Diagrama de Interacción	Cantidad de objetos, cantidad de mensajes, Cantidad de Enlaces o relaciones.

Cuadro 4. Métricas de diagramas de Interacción.

Los diagramas de colaboración son los diversos objetos que están relacionados en la interacción y sus métricas se pueden ver en el cuadro 5.

Tipo de Diagrama	Métricas asociadas
Diagrama de colaboración	Cantidad de objetos, cantidad de mensajes, Cantidad de arcos o relaciones. Cantidad de secuencias

Cuadro 5. Métricas de diagramas de colaboración.

Los diagramas de actividades representan el flujo de las acciones y se detalla su contenido, está conformado por las siguientes métricas como esta en el cuadro 6.

Tipo de Diagrama	Métricas asociadas
Diagrama de Actividades	Cantidad de actividades, cantidad de estados de acciones, número de transiciones y objetos.

Cuadro 6. Métricas de diagramas de actividad

Los diagramas de componentes representan elementos físicos como ejecutables, tablas, archivos y documentos, sus elementos de medida se pueden ver en el cuadro 7.

Tipo de Diagrama	Métricas asociadas
Diagrama de componentes	Cantidad de paquetes, cantidad de relaciones entre componentes, Cantidad de interfaces

Cuadro 7. Métricas de diagramas de componentes

Los diagramas de despliegue es la forma de desplegar los componentes mediante nodos que representan por ejemplo un procesador, su métricas asociadas están relacionadas en el cuadro 8.

Tipo de Diagrama	Métricas asociadas
Diagrama de despliegue	Cantidad de nodos, cantidad de conexiones

Cuadro 8. Métricas de diagramas de despliegue

Con la identificación de las métricas de los diagramas de UML, viene un proceso de validación y de relación entre diagramas, proceso que representa bastante complejidad y en la que se han realizado múltiples investigaciones pero la consistencia de estos no ha sido muy trabajado. Para este proceso se realizan métodos relacionado con reglas definidas en un lenguaje de restricciones de objetos (OCL) para garantizar la consistencia entre diagramas (Zapata, et al, 2008). También existen otros métodos para validar los diagramas de UML mediante la trazabilidad y verificación de sus requerimientos funcionales controlando los cambios que se puedan presentar mediante la automatización de sus procesos a nivel conceptual (Medina, et al, 2016).

En la investigación se está desarrollando una herramienta que permita almacenar las reglas de construcción de los diagramas de UML y con base en este almacenamiento se pueda validar los requerimientos y los diagramas de UML para que en forma temprana se pueda determinar errores antes de desarrollar las aplicaciones o proyectos informativos.

Análisis

En este trabajo investigativo se analizaron los diferentes artefactos asociados a los diagramas de UML y sus métricas relacionadas.

Se desarrolló un repositorio para almacenar las métricas donde se permite asociar a cada diagrama de UML su conjunto de métricas, estas se podrán modificar, borrar o consultar para poder validar el proceso de los requerimientos funcionales de cualquier aplicación que se desarrolle. El sistema de almacenamiento se desarrolló en ambiente de Windows 10 pro con SQL server Management Studio, visual studio y C#.

La arquitectura que soporta el almacenamiento de métricas se puede visualizar como se muestra en la figura 1.

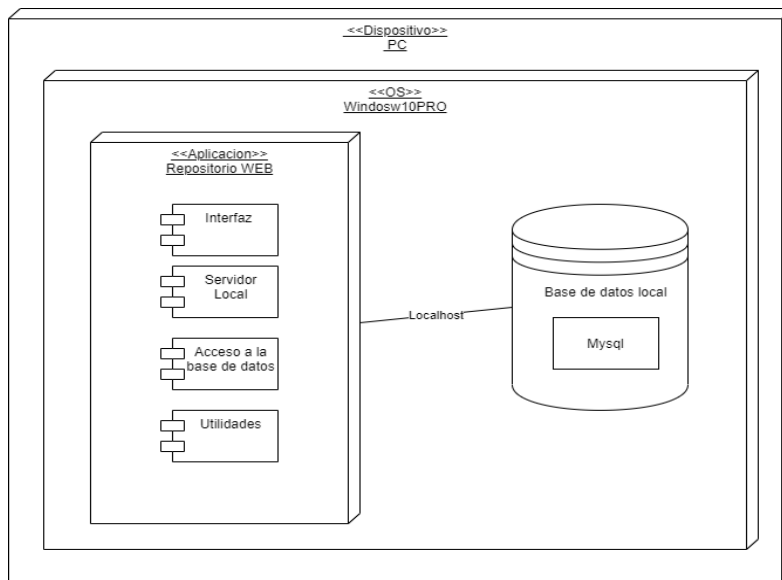


Figura 1. Arquitectura del sistema de almacenamiento de métricas de calidad

Conforme a los requerimientos el sistema permite almacenar métricas validadas y asociadas a cada uno de los diagramas de UML, para esta representación en la figura 2 se puede ver el caso de uso general.

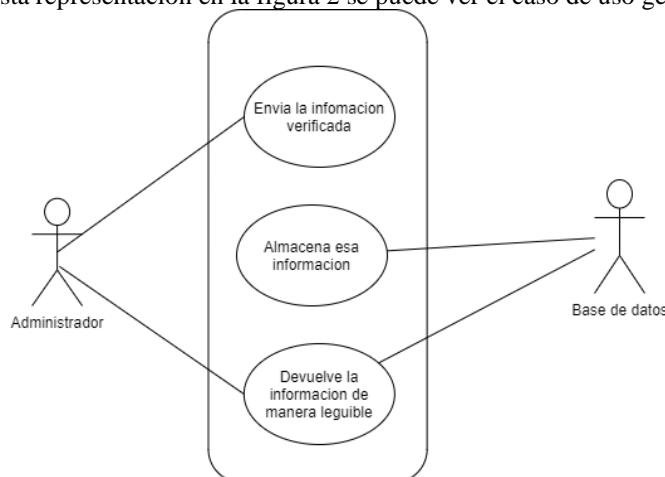


Figura 2. Casos de uso general del sistema de almacenamiento de métricas de calidad

Los resultados de la investigación determinan la importancia que tienen las métricas de calidad asociadas a los diagramas de UML para identificar errores de concepción y evitar que los proyectos acumulen defectos en el desarrollo de su ciclo de vida.

Conclusiones

El presente proyecto propone la identificación y relación entre los requerimientos y los diagramas de UML para determinar en forma temprana posibles errores en su concepción. El fin último fue soportar estas métricas sobre una

herramienta que apoye la labor del diseñador y desarrollador de software valorando las relaciones que deben tener los diagramas y su calidad en términos de estándares. Se determinó que UML es fundamental a la hora del desarrollo, por lo que se indagara en su importancia como herramienta, exponiendo cada una de las funcionalidades que brinda y como facilita el desarrollo de un software, para luego ver sus respectivas aplicaciones dentro de lo establecido por la International Organization for Standardization - ISO.

Los resultados demuestran la necesidad de utilizar y aplicar estándares de calidad del software, para esto es indispensable validar los diagramas con los requerimientos y los prototipos.

Uno de los impactos importantes de este proyecto fue investigar sobre el uso de los metadatos para poder desarrollar herramientas que permitan mejorar los conocimientos en arquitecturas y desarrollo de software, y otro relacionado con la forma en que se puede innovar en la identificación de métricas que midan en forma temprana la calidad del desarrollo de software sobre la base de los diagramas UML.

Recomendaciones

Los investigadores interesados en continuar nuestra investigación pueden automatizar los procesos derivados de las métricas de calidad del software por cada uno de los diagramas de UML que se utilicen como resultado de un proceso de abstracción del mundo real, aunque existen algunos estudios y herramientas relacionadas es importante seguir consolidando este proceso para que los desarrolladores utilicen la métricas como fundamento de buenas prácticas de desarrollo.

Referencias

- Booch, G., Rumbaugh, J., Jacobson, I.(2006) El Lenguaje Unificado de Modelado (UML).Person. 2 ed.. Pag. 432
- Género, M., Piattini, M. y Calero, C. Early Measures For UML class diagrams. L´Objet. 6(4), Hermes Science Publications, 489-515, 2000.
- Marchesi M., OOA Metrics for the Unified Modeling Language, Proceedings of the 2nd Euromicro Conference on Software maintenance and reengineering, p. 67-73, 1998
- Medina, O., Martín, M., Groppo, M. (2016). Trazabilidad y validación de requerimientos funcionales de sistemas informáticos mediante la transformación de modelos conceptuales. ReCIBE, 5(1).
- OBJECT MANAGEMENT GROUP (OMG).(2017). UML 2.5.1. Disponible en: <http://www.omg.org/spec/UML/2.5.1/PDF>.
- OBJECT MANAGEMENT GROUP (OMG).(2013). UML Testing Profile (UTP) versión 1.2. Disponible en: <http://www.omg.org/spec/UTP/1.2/>
- Piattini, M., García, F., Garzas J., Género, M. (2008). Medición y estimación del software. Técnicas y métodos para mejorar la calidad y la productividad. Alfaomega-Ra-Ma. pag. 332
- Piattini, M., García, F., Caballero, I. (2010). Calidad de Sistemas Informáticos. 2da edición. Alfaomega-RA-ma. pag 416.
- Standishgroup. (2015). Standish Group 2015 Chaos Report. Recuperado de: <https://www.standishgroup.com/store/>
- Rodríguez, D., Harrison, R., " Medición en la Orientación a Objeto ", en Medición para la Gestión en la Ingeniería del Software , Dolado, J. y Fernández, L., Eds., RA-MA, 2000
- Zapata, C., González, G. (2008). Especificación formal en OCL de reglas de consistencia entre los diagramas de clases y casos de uso de UML y el modelo de interfaces Formal OCL specification of consistency rules between the UML class and the use case models and the interfaces model. Revista Ingenierías Universidad de Medellín, 7(12), 169–191.