CONSTRUCCION DE UN SISTEMA DE APOYO A LA INGENIERIA DE SOFTWARE

Javier Darío Fernández Ledesma[1]

Abstract — *La finalidad de este artículo es mostrar el proceso de construcción de un sistema de apoyo a la ingeniería de software, en el se desarrollan los antecedentes conceptuales e investigativos inherentes a los sistemas desarrollados para acompañar las labores de ingeniería de software en el contexto académico e industrial, se presenta el Metamodelo de construcción del sistema propuesto y el prototipo construido que sirve como marco de actuación e implementación de los proyectos de ingeniería de software.*

Index Terms — Ingeniería de Software, Métricas, Metamodelos, Proyectos

INTRODUCCIÓN

Este artículo es un acercamiento al desarrollo metodológico de una propuesta de construcción de un sistema de apoyo a la ingeniería de software. En la primera parte se muestran los antecedentes conceptuales e investigativos que dan cuenta del alcance logrado en materia de modelos, técnicas y herramientas usadas en los procesos de la ingeniería de software, en la segunda parte se hace una acercamiento a la propuesta metodológica y a la construcción del Metamodelo que soporta la construcción del sistema propuesto y en la tercera parte se muestra el desarrollo de un primer prototipo que sirve como marco de actuación e implementación de los proyectos de ingeniería de software.

ANTECEDENTES

SOBRE LAS HERRAMENTAS DE APOYO

En las últimas décadas se ha trabajado en el área de desarrollo de sistemas para encontrar técnicas que permitan incrementar la productividad y el control de calidad en cualquier proceso de elaboración de software, y hoy en día la tecnología CASE (Ingeniería de Software Asistida por Computador) reemplaza al papel y al lápiz por el ordenador, para transformar la actividad de desarrollar software en un proceso automatizado.

Las herramientas CASE evolucionan hacia tres tipos de integración:

- La integración de datos permite disponer de herramientas CASE con diferentes estructuras de diccionarios locales para el intercambio de datos.
- La integración de presentación confiere a todas las herramientas CASE el mismo aspecto.
- La integración de herramientas permite disponer de herramientas CASE capaces de invocar a otras CASE de forma automática.

Existen en la actualidad un sinnúmero de herramientas de soporte al modelado, análisis, planificación, y en sí, a todas las etapas del proceso de desarrollo de software, algunas de ellas son:

En términos de Gestión del Proceso, encontramos herramientas como:

GDPA: Son herramientas para la definición del proceso de investigación, desarrollada en la Universidad de Bremen en Alemania, proporciona un amplio aspecto de funciones para el modelado y la gestión del proceso.

SpeeDev: Desarrollada por Spee Dev Corporation, incluye una serie de herramientas para la definición del proceso, gestión de los requisitos, resolución de características, planeación del proyecto y seguimiento del mismo.

Step Gate Process: Desarrollado por Objexis, incluye muchas herramientas que ayudan en la automatización del flujo de trabajo.

En términos de herramientas para el Desarrollo Ágil de software encontramos:

Actif Extreme: Desarrollado por Microtool, proporciona soporte para la gestión de un proceso ágil en varias actividades técnicas dentro del proceso.

Ideogramic UML: Desarrollado por Ideogramic, es un conjunto de herramientas para el UML creado en forma específica para usarlo dentro de un proceso ágil.

Together Tool Set: Distribuido por Borland, proporciona un paquete de herramientas que apoya muchas actividades técnicas dentro de la programación extrema y otros procesos ágiles.

Así mismo, entre las herramientas de modelado de datos, encontramos:

Allfusion Edwin: Desarrollado por Computes Associates, ayuda en el diseño de objetos de datos, estructuras propias y elementos clave para bases de datos.

ER/Studio: Desarrollado por Embarcadero Software, brinda soporte al modelado Entidad – relación.

Oracle/Designer: Desarrollado por Oracle Systems, modela procesos de negocios, entidades de datos y relaciones que se transforman en diseños a partir de los cuales se generan aplicaciones completas y bases de datos.

1 Javier Darío Fernández Ledesma. Phd (c) Ingeniería Electrónica, Msc en Ingeniería, Calle 50 A N.41-40, Medellín, Colombia, ingenierojdfl@gmail.com

MetaScope: Desarrollado por Madrone Systems, es una herramienta para el modelado de datos de bajo costo que da soporte a la representación grafica de datos.

ModelSphere: Desarrollado por Magna Solutions GMBH, da soporte a una variedad de herramientas de modelado relacional.

VisibleAnalyst: Desarrollado por Visible Systems, da soporte a una variedad de funciones de modelado del análisis, incluido el modelado de datos.

También encontramos herramientas para análisis estructurado como:

AxiomSys: Desarrollado por STG, Inc. Proporciona un paquete completo de herramientas para el análisis de la estructura que incluye extensiones de Hartley-Pirbhai para el modelado de sistemas en tiempo real.

MacA&D, winA&D: Desarrollado por Excel Software, proporciona un conjunto de herramientas simples y baratas para el análisis y el diseño de maquinas Mac y Windows.

MetaCASE WorkBench: Desarrollado por MetaCase Consulting, es una metaherramienta utilizada para definir un método de análisis o diseño, sus conceptos, reglas, notaciones y generadores.

System Architect: Desarrollado por Popkin Software, proporciona un amplio rango de herramientas de análisis y diseño, incluye herramientas para el modelado de datos y el análisis estructurado.

Se encuentran también herramientas para UML/OCL, entre las cuales están:

ArgoUml: Distribuido por Tigress.org, da soporte a Uml y Ocl completo, e incluye varias herramientas de ayuda para el diseño, que va mas allá de la generación de diagramas Uml y expresiones Ocl.

Dresden Ocl Toolkit: Desarrollado por Frank Finger en la Universidad Tecnológica de Dresden, es un juego de herramientas basada en un compilador Ocl que abarca varios módulos que analizan, revisan el tipo y normalizan las restricciones Ocl.

OCL parser: Desarrollado por IBM, este escrito en Java y esta disponible gratuitamente para la comunidad orientada a objetos con el fin de que se estimule el uso de OCL con modeladores Uml.

Entre las herramientas para la estimación de métricas tenemos:

Netmechanics Tool: Desarrollado por Netmechanic, es una colección de herramientas que ayudan a mejorar el desempeño de un sitio Web, se enfoca sobre los temas específicos de la implementación.

NIST Web Metrics Testbed: Desarrollado por The Nacional Institute of Standards and Technology, abarca la siguiente colección de herramientas útiles que están disponibles para ser descargadas:

- Web Static Analizar Tool: verifica el HTML de la página Web contra los lineamientos de facilidad de uso típicos.
- Web Category Analysis Tool: permite al ingeniero de facilidad de uso construir y dirigir un análisis de categoría Web.
- Web variable Instrumenter Program: instrumenta un sitio Web para capturar un registro de interacción de usuario.
- Framework for logging Usability Data: implementa un formateador y analizador gramatical de archivos para representar los registros de interacción de usuario.
- Vis VIP Tool: produce una visualización tridimensional de las rutas de navegación del usuario a través de un sitio Web.
- TreeDec: agrega auxiliares de navegación de un sitio Web.

También encontramos herramientas para gestores de software como:

Project Control Panel: Desarrollado por Software Program Manager's Network, ofrece a los gestores de proyectos un aviso directo del estado del proyecto, la herramienta tiene "calibradores" muy parecidos a un tablero y esta implementada con Microsoft Excell.

Ganthead.com ha desarrollado un conjunto de útiles listas de verificación para gestores de proyectos.

Ittoolkit.com ofrece una colección de guías de planificación, plantillas de proceso y hojas de cálculo inteligentes.

Finalmente, en términos de herramientas para métricas encontramos:

Function Point WORKBENCH: Desarrollada por Charismatek, ofrece una amplia variedad de métricas orientadas a PF.

Metric Center: Desarrollada por Distributive Software, soporta recopilación automatizada de datos, análisis, formateo de graficas, generación de reportes y otras tareas de medición.

PSM Insight: Desarrollada por Practical Software and Systems Measurement , auxilia en la creación y subsiguiente análisis de una base de datos de medición del proyecto.
SLIM Tool set: Desarrollado por QSM, proporciona un completo conjunto de métricas y herramientas de estimación.

SPR tool set: Desarrollado por Software Productivity Research, ofrece una colección detallada de herramientas orientadas a PF.

TychoMetrics: Desarrollado por Predicate Logia Inc, es una suite de herramientas para gestionar recopilación de métricas y reportes.

Como vemos existen un sinnúmero de herramientas enfocadas a la gestión del desarrollo de software, la automatización, todas a su vez impactando áreas especificas de la ingeniería de software que requiere de un integración y una plataforma de gestión única donde se tenga en cuenta el modelado, la verificación y la estimación como elementos claves para un desarrollo de software de alta calidad.

Uno de los aspectos más relevantes es la necesidad de mantener los niveles de consistencia entre diferentes diagramas de diseño y el código generado para asegurar la calidad del software, aspecto este que pretendemos atacar con el proyecto propuesto.

METODOLOGÍA

El prototipo del Sistema de Apoyo a la Ingeniería de Software se implementó utilizando la tecnología del lenguaje de programación Java, puesto que es una herramienta que nos permite manejar múltiples plataformas de implementación, portable y de bajo costo.

Java es un lenguaje de programación orientado a objetos el cual ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de los demás lenguajes de programación, al ser este un lenguaje más simple, usable y de fácil interacción.

Para el Proceso de Diseño y Almacenamiento de Datos referentes a los Proyectos y Modelos se utilizaron documentos del tipo XML, por su fácil interacción con la plataforma de desarrollo.

Por último, tanto la interfaz en Java como los Documentos XML generados permiten crear las estructuras necesarias para hacer una correcta gestión de los proyectos de software.

Para la construcción de un sistema de apoyo a la ingeniería de software que logre unificar los procesos de modelado, estimación y validación de reglas de consistencias se hace necesario dar respuesta al siguiente interrogante investigativo ¿Existe una metodología unificada que permita integrar los procesos de gestión y estimación de software con las herramientas de modelado y validación de inconsistencias entre diagramas?, al ser enfrentado este interrogante se ha propuesto un esquema de trabajo que permite brindar pautas para la construcción de una *MUSIS* (Metodología Unificada de Soporte a la Ingeniería de Software), este esquema viene representado a través de las siguientes fases:

- Fase de Definición de Proyectos de Software: Se definen los elementos necesarios que involucra el proceso de desarrollo de software para su estimación y modelado.

En ella se definen las estructuras que soportan el proyecto de software en términos de costos, tiempos, indicadores de gestión y métricas de estimación.

- Fase de Estimación: Implementando un algoritmo de estimación se pueden calcular las diferentes métricas que acompañan el proyecto de software.

En ella se definen las estructuras de las métricas de gestión y de calidad que soportaran el proyecto de software a estimar.

- Fase de Modelado y Validación: Implementando un Modelador Grafico se podrán construir los diagramas más representativos del proyecto de software estimado y su respectiva validación en correspondencia con el cumplimiento de los estándares UML.

Una vez se han establecido los parámetros de estimación y la definición de los requerimientos funcionales del proyecto, se procede al diseño del mismo haciendo uso de los diagramas más representativos de UML que contiene la herramienta propuesta; así mismo se definen las reglas de validación que entraran a soportar los niveles de consistencias de los diagramas generados con el estándar de UML, como se puede apreciar en la grafica siguiente:

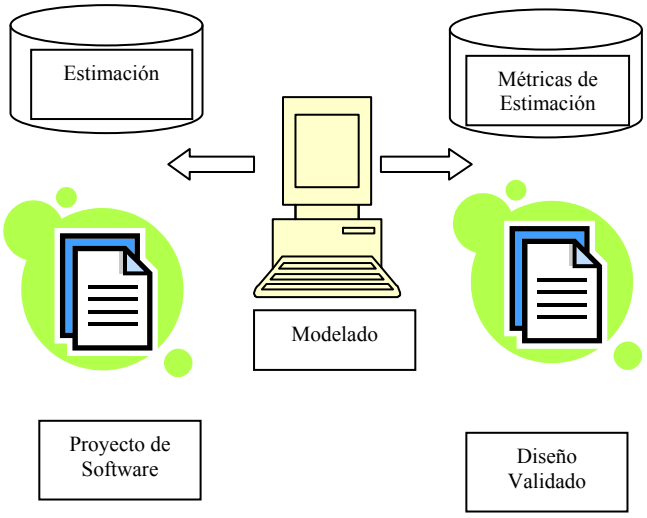

FIGURA 1.

Esquema General del Sistema Propuesto

Una vez presentado el esquema Macro del sistema propuesto, se procede, haciendo uso del UML –Lenguaje Unificado de Modelado-, a presentar el Metamodelo para la construcción del sistema de apoyo a la ingeniería de software.

Análisis y diseño

Diagrama de Casos de Uso

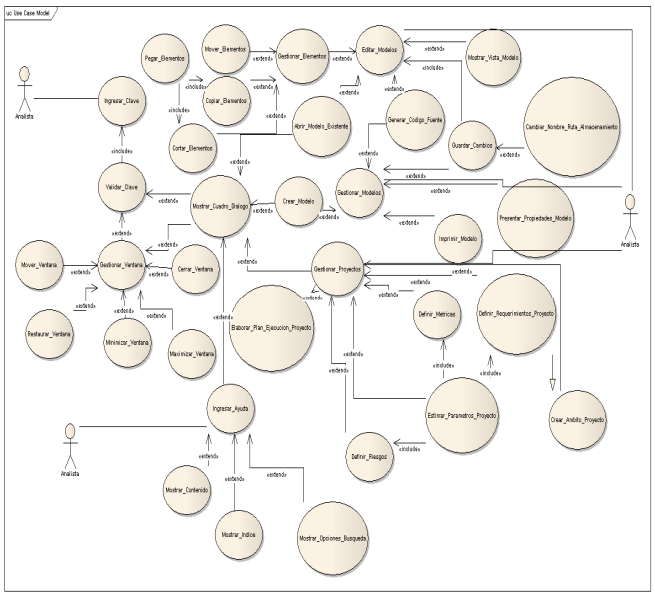

FIGURA 2.

Diagrama de Casos de Uso del Sistema

Diagrama de Objetos

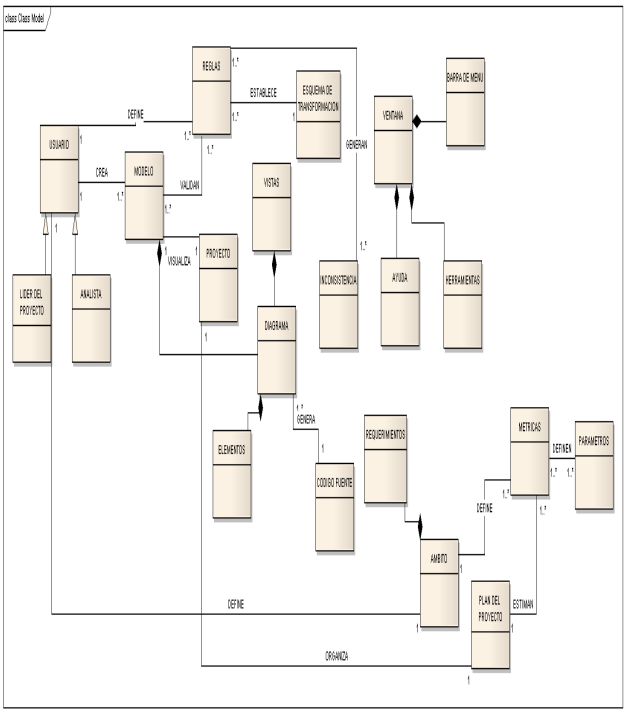

FIGURA 3.

Diagrama de Objetos del Sistema

RESULTADOS PARCIALES

Durante el proceso de desarrollo se han ido evaluando permanentemente los elementos que entran a conformar el sistema propuesto, entre los eventos importantes que se han presentado durante esta fase del proceso de desarrollo del sistema tenemos:

Se realizo un analisis en terminos de determinacion del estado del arte sobre los elementos que entraran a constituir la propuesta de construccion de un Sistema de para generar un esquema metodologico unificado como el propuesto.

Se han construido los diagramas mas representativos de UML para modelar el sistema propuesto, como con: el Diagrama de Casos de Uso con su respectiva documentacion, el diagrama de objetos, el diagrama de clases, el diagrama de actividades, el diagramade secuencias, el diagrama de estados, el de componentes y despliegue, haciendo una revision permanente sobre los cambios a realizar en el sistema.

Se ha construido un primer prototipo grafico con el lenguaje de programación Java y XML para el Modelador o Diseñador del sistema

REFERENCIAS

André P., Romanczuk A., Royer J. C. and Vasconcelos A. (2000). Checking the Consistency of UML Class Diagrams Using Larch Prover. Universidad de Nantes (Francia). Presentado en Conferencia Rigorous Object-Oriented Methods 2000, York, enero de 2000. Disponible en http://ewic.bcs.org/conferences/2000/objectmethods/papers/paper1.pdf

Bart Verheecke (2001). From declarative constraints in conceptual models to Explicit constraint classes in implementation models. Tesis para obtener el grado se licenciado en informática en Vrije Universiteit Brussel. Disponible en: http://ssel.vub.ac.be/Members/BartVerheecke/papers/Verheecke_thesis2001.pdf

Dorr, H. "Efficient Graph Rewriting and its Implementation". Lectures Notes in Computer Science. Springer, 922. 1995.

Favre Liliana. "Foundations for MDA-based Forward Engineering". Publicado en Journal of Object Technology, vol. 4, no. 1, enero-febrero de 2005, pp. 129-153. Disponible en http://www.jot.fm/issues/issue_2005_01/article4.

Halpin Terry. .Augmenting UML with Fact-orientation. Presentada en workshop UML: a critical evaluation and suggested future, HICCS-34 conference, Maui, enero de 2001. Disponible en http://www.orm.net/pdf/HICSS.pdf

Génova Gonzalo. Entrelazamiento de los aspectos estático y dinámico en las asociaciones UML. Tesis Doctoral en ingeniería informática Universidad Carlos III (España). Disponible en http://www.ie.inf.uc3m.es/ggenova/tesis-esp.html

Muñoz, Javier; Albert, Manoli; Ruiz, Marta; Pelechano, Vicente. Transformación de Relaciones de Asociación Mediante Gramáticas de Grafos. Aplicando MDA. I Taller sobre Desarrollo de Software Dirigido por Modelos, MDA y Aplicaciones (DSDM). Málaga, España. November 2004, pp. 43 - 56.

Pressman, Roger. "Ingenieria de Software: Un enfoque practico". 6 Ed. Publicado en Prentice Hall, 2005.

Varró, Daniel and Pataricza, András. "UML Action Semantics for Model Transformations". Publicado en International Journal of Periodica Politechnica, 2003.