

Herramientas y Métodos de soporte a la trazabilidad y evolución de modelos aplicado a la Producción Ágil de Software en ambientes de Desarrollo Dirigido por Modelos (MDD)

Giandini, Roxana ^(1,2), Nahuel, Leopoldo ^(1,2), López, Carlos ⁽²⁾,
Losada, Marina ⁽²⁾, Robles, Nicolás ⁽²⁾

⁽¹⁾ LIFIA - Facultad de Informática - UNLP
{giandini; lnahuel}@lifia.info.unlp.edu.ar

⁽²⁾ LINSI - Departamento de Sistemas de Información - Facultad Regional La Plata - UTN
{mlosada; nrobles; clopez}@linsi.edu.ar

RESUMEN

El objetivo de este proyecto es realizar actividades de investigación y desarrollo en temas relacionados a Ingeniería de Software Basado en Modelos (ISBM) y el paradigma de Desarrollo de Software Dirigido por Modelos (MDD) en conjunción con las metodologías ágiles para la producción de software, con el interés particular de evaluar e implementar mecanismos de trazabilidad entre los distintos artefactos involucrados en el proceso de construcción de proyectos software que utilizan lenguajes notacionales estándar UML-OCL-SysML y herramientas CASE de soporte a la edición de modelos. En este contexto es necesario establecer mecanismos que permitan automatizar la transformación de modelos en un ambiente de desarrollo ágil de aplicaciones conducido por modelos, considerando aspectos que incluyen trazabilidad de artefactos de modelado y evolución en un marco de trabajo cambiante.

Palabras clave: Ingeniería de Software, Metodologías ágiles, Lenguajes de modelado, Herramientas CASE, Trazabilidad de modelos, Desarrollo Dirigido por Modelos.

CONTEXTO

Las bases de conocimiento y línea de investigación se apoyan en la experiencia de estudios y desarrollos obtenidos a través del proyecto "PAMPA" (Precise Assistant for Modelling Process Activities) [1, 2] y el proyecto "Lenguajes para transformaciones de Modelos en MDD" [4], homologados por resolución académica UTN y apoyados por

Microsoft que ha desarrollado el grupo de investigación en la UTN-Facultad Regional La Plata. Como resultado del primer proyecto se construyó una herramienta CASE (Computer Aided Software Engineering) orientada al modelado formal incluyendo especificación de refinamientos de modelos, aspectos que puede verse como un caso particular de transformación de modelos en MDD. Sobre estos tópicos han sido publicados - entre otros - los artículos: [3, 6, 15] que aportan una base formal a la herramienta.

Como resultando del segundo proyecto, relacionado con la definición de lenguajes y transformaciones de modelos en MDD se han publicado los trabajos [7, 8, 9, 10, 11, 12, 14, 16] y dictado tutoriales como [13].

Ambos proyectos, publicaciones citadas y sus resultados teóricos constituyen un aporte a la construcción de prototipos de herramientas CASE de código abierto. Este grupo de investigación también ha recibido apoyo de la industria a través de *grants* como el siguiente: Grant Microsoft a la Innovación Curricular: "Request For Proposals 2005" en el tema "Application of formal methods to software development", consistente en US\$25,000 (Marzo 2005, por concurso internacional).

Actualmente, esta propuesta se enmarca dentro de un proyecto PID-UTN 2010 con Incentivos Docentes aprobado recientemente, con número de Resolución en trámite.

1. INTRODUCCION

Actualmente la ingeniería de sistemas está siendo orientada a un enfoque basado en modelos soportado por potentes herramientas CASE para generar código fuente de software,

frente al tradicional enfoque documental, a la hora de escribir los sistemas.

Los modelos son representaciones parciales de los sistemas. Utilizan una notación y una semántica claramente definida para describir de forma rigurosa ciertos aspectos, en contraposición al lenguaje natural. Son parte fundamental del paradigma de desarrollo de software conocido como "Model Driven Development" (MDD) [5], [24]. En las etapas de MDD se parte de los requisitos del sistema y se desarrollan modelos de arquitectura y diseño desde niveles de abstracción altos a niveles más detallados, que incluyen la definición de la plataforma de ejecución. En particular, UML (Unified Modeling Language), SysML [21] (Systems Modeling Language) y OCL (Object Constraint Language) representan tres lenguajes notacionales estándar. UML está ampliamente instalado en la industria de software, SysML fue específicamente concebido para la descripción de sistemas complejos (no necesariamente software) y actualmente ha logrado fuerte aceptación por la industria de sistemas, mientras que OCL [22] provee soporte formal para definir restricciones en fases de diseño de sistemas que siguen las bases del estilo de programación orientada a objetos. También existen en el mercado, herramientas estándares más genéricas basadas en la técnica de metamodelado, que permiten definir la sintaxis de nuevos lenguajes de modelado y no se reducen sólo a la edición de modelos en UML.

En el contexto del Desarrollo Dirigido por Modelos (MDD), la trazabilidad o rastreo de elementos es un importante desafío teórico y práctico [18], [19], [20]. Los mecanismos de rastreo son necesarios en procesos que involucran transformación automática entre modelos, en generación de código y en actividades de modelado manual. Actualmente se enfrenta el desafío de mantener modelos consistentemente sincronizados, modelos y código en estados consistentes, como así también sincronizar actualizaciones manuales mezcladas con procesos automáticos. Algunas de las áreas de aplicación son: Administración de cambios de modelos, Administración de cambios de código y Análisis de impacto de cambios.

Más allá de la metodología basada en modelos, en la última década ha surgido una nueva corriente metodológica de soporte al desarrollo ágil de aplicaciones software [17], centrada en

la productividad e integración de los equipos de trabajo para lograr una rápida interacción con el usuario, permitiendo así validar y consensuar los requerimientos de la aplicación requerida de forma temprana. Los procesos ágiles para el desarrollo de software, conocidos anteriormente como metodologías livianas, intentan evitar las rigidez de normativas, fuerte dependencia de planificaciones detalladas previas al desarrollo y burocráticos caminos de las metodologías formales de la Ingeniería de Software, enfocándose en los valores esenciales del Manifiesto Ágil [23]: "A los individuos y su interacción, por encima de los procesos y las herramientas"; "El software que funciona, por encima de la documentación exhaustiva."; "La colaboración con el cliente, por encima de la negociación contractual."; "La respuesta al cambio, por encima del seguimiento de un plan". Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos respecto a los procesos tradicionales, desarrollando software en cortos lapsos de tiempo. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Una iteración no debe agregar demasiada funcionalidad, pero la meta es lograr obtener una demo (versión de demostración que permite al usuario la prueba de un programa.) sin errores al cierre de cada iteración para validar con el usuario el avance, calidad y ajustes en los requerimientos de sistema; allí el equipo vuelve a evaluar las prioridades del proyecto.

Resulta entonces interesante poder combinar o relacionar la metodología MDD con el proceso ágil de desarrollo de software. Por lo expuesto, el objetivo general de este proyecto es el desarrollo, a través del estudio, análisis e implementación de prototipos, de una herramienta de producción de software con base formal (basada en metamodelos) en el contexto MDD, siguiendo un proceso ágil de modelado que favorezca la propagación de cambios (por el mecanismo de trazabilidad) a partir de los requisitos del sistema.

Por otro lado, la propuesta de Gestión de Procesos de Negocios (BPM) [26] ha adquirido una atención considerable en la última década, tanto por las comunidades de administración de negocios como por la comunidad informática. BPM provee un conjunto de metodologías para el análisis, comprensión y documentación de los procesos del negocio. En este dominio, el lenguaje de modelado estándar del OMG

(Object Management Group), Business Process Modeling Notation (BPMN) [25] se ha tornado popular, así como otras notaciones existentes en el mercado. Sin embargo estas notaciones no son simples y siguen existiendo problemas básicos; por ejemplo, la validación de estos modelos por parte del cliente. Esta validación es una etapa crucial en un entorno interorganizacional en el que varias empresas interactúan para lograr sus objetivos del negocio. En consecuencia, como continuación al objetivo ya expuesto, proponemos facilitar y agilizar el modelado de procesos del negocio desde los requisitos del sistema a través de los mecanismos de trazabilidad y transformaciones de modelos que integremos a la herramienta CASE a desarrollar.

2. LINEAS DE INVESTIGACION y DESARROLLO

Será necesario para el desarrollo del trabajo, el estudio y análisis de los siguientes estándares y herramientas:

- Estándar MDA (Model Driven Architecture) [22] de OMG para la visión MDD (Model Driven Development).
- Lenguajes de modelado: UML (Unified Modeling Language), OCL (Object Constraint Language), SysML (Systems Modeling Language), BPMN (Business Process Modeling Notation).
- Lenguajes de metamodelado: MOF (Meta Object Facility), EMF (Eclipse Modeling Framework), EuGENia.
- Metodologías ágiles de desarrollo de software: Feature Driven Development (FDD), Adaptive Software Development (ASD), Agile Modeling (AM), Agile Unified Process (AUP), Essential Unified Process (EssUP), Test Driven Development (TDD), eXtreme Programming (XP), SCRUM, Kanban, Crystal Clear.
- Estándares para los procesos de ciclo de vida del software (Normas ISO/IEC 12207:2008).
- Patrones funcionales del campo de la Ingeniería de Requerimientos: Feature (característica/ventaja del producto), User Story, Casos de Uso, Requisitos SysML.
- Patrones arquitectónicos para el diseño e implementación de software: GoF (Gang Of Four), GRASP (General Responsibility Assignment Software Patterns).

- Patrones de Arquitectura (Capes, Pipeline, MVC, entre otros.)
- Diseño y ejecución de Test Cases (Casos de Prueba) para determinar que la herramienta CASE cumpla con los requisitos planteados.

3. RESULTADOS Y OBJETIVOS

Se planifica avanzar en la capacitación continua de los miembros de ésta línea de investigación.

Como objetivo general se espera desarrollar una herramienta CASE con base formal de soporte al manejo de evolución y trazabilidad de modelos en un ambiente ágil de producción de software conducido por modelos, aplicable a un dominio específico (tal como al modelado de procesos de negocio (BPM)).

Para tal fin, se han considerado desarrollar los siguientes objetivos específicos:

- Personalizar artefactos estándares de modelado que permitan construir ágilmente modelos PIM (Platform Independent Model – Modelo independiente de la plataforma tecnológica de desarrollo) y PSM (Platform Specific Models – Modelo descrito en una plataforma o tecnología de implementación específico) para producir software en un ambiente MDD.
- Establecer mecanismos de trazabilidad entre los distintos artefactos de modelado para facilitar la escalabilidad y evolución de los modelos productivos en ambientes MDD.
- Definir un metamodelo para especificar evolución y trazabilidad entre modelos de distinto nivel de abstracción: PIM → PSMs y PSM → Modelo texto (código fuente).
- En base al metamodelo definido previamente, definir un marco de proceso para el desarrollo ágil de sistemas que de soporte a la producción de software considerando evolución y trazabilidad de modelos.
- Construir una herramienta CASE con capacidades tales que permitan automatizar tareas de evolución y trazabilidad entre modelos de distinto nivel abstracción.

- Extender la herramienta CASE para agilizar y facilitar el modelado de procesos del negocio desde los requisitos del sistema, mediante mecanismos de trazabilidad y transformaciones de modelos.

4. FORMACION DE RECURSOS HUMANOS

El Proyecto I/D en el que se enmarcan estas líneas de trabajo, es parte de un proceso de incentivación para el desarrollo de actividades I/D, que se originan en el Laboratorio LINSI del Departamento de Sistemas de Información de la FRLP-UTN. Consecuentemente, se está poniendo acento y esfuerzo en las siguientes actividades:

- Desarrollo de seminarios abiertos de formación general en relación a temas, técnicas y tecnologías incluidos en esta línea de investigación, para alumnos avanzados en la carrera de Ingeniería en Sistemas y para becarios de este equipo de trabajo, realizados en UTN-Facultad Regional La Plata y UNLP-Facultad de Informática.
- A cargo de integrantes de este equipo de trabajo, se encuentran en curso 2 Tesis de Postgrado (Magíster en Ingeniería de Software, UNLP) relacionadas con el campo de investigación de éste proyecto.
- Finalmente, los docentes integrantes de esta línea de investigación participan en el dictado de asignaturas con contenidos relacionados a este campo I/D:

En grado: UTN-Facultad Regional La Plata, Ingeniería en Sistemas de Información y UNLP- Facultad de Informática.

En postgrado: UAI-Facultad de Tecnología Informática, Maestría en Tecnología Informática y UNLP-Facultad de Informática, Maestría en Ingeniería de Software.

5. REFERENCIAS

[1] Proyecto “Herramientas de soporte al proceso de desarrollo de software basado en modelos”, financiado por Microsoft RPF 2004. Grupo de Investigación: C. Pons, R. Giandini, M.Pinasco, D. Garcia, A.Gaig, L. Cuaderno, F. Salvatierra, L. Ocaranza, L. Nahuel, <http://firlp.utn.edu.ar/pampa>

[2] L. Cuaderno, E. Di Lorenzo, D. Garcia, R. Giandini, L. Nahuel, L. Ocaranza, M.Pinasco,

C. Pons, F. Salvatierra. “Una herramienta de soporte al proceso de desarrollo dirigido por modelos (MDD)” Actas XII° CACIC (Congreso Argentino de Ciencias de la Computación) 2006. Octubre de 2006, San Luis, Provincia de San Luis, Argentina.

[3] Pons, C., Pérez, G., Giandini, R., Kutsche, R. Understanding Refinement and Specialization in the UML. 2nd International Workshop on Managing Specialization/Generalization Hierarchies (MASPEGHI). In IEEE ASE 2003, Canada.

[4] Título del Proyecto: Lenguajes para transformaciones de Modelos en MDD. Acreditación: Proyectos de PID UTN 25-I025. Duración: 3 años. Fecha del Proyecto: 1/1/2006 hasta 31/12/2008. Disposición Resolutiva 119/08 para Programa de Incentivos. Grupo de trabajo integrado por 5 personas. Director: Roxana Giandini. Co-directores: C. Pons, C. López

[5] “Desarrollo de Software dirigido por Modelos - Conceptos teóricos y su aplicación práctica” Con referato, aprobado por el HCA de la Facultad de Informática, UNLP. Autores: Claudia Pons, Roxana Giandini y Gabriela Pérez. Editorial EDULP & McGraw-Hill Educación, 2010. Volumen 1, 300 páginas. ISBN: 978-950-34-0630-4

[6] “Herramientas de soporte al proceso de desarrollo dirigido por modelos y su implementación con DSL Tools”, L. Nahuel, M. Pinasco, R. Giandini, C. Pons - II° JIDIS (Jornadas de Investigación y Desarrollo en Ingeniería de Software). Organizadores: Microsoft de Argentina y UTN. Córdoba, Argentina. Mayo de 2007.

[7] “A Two-level Calculus for Composing Hybrid QVT Transformations”, C. Pons, R. Giandini, G. Perez, G. Baum. Editors: H. Astudillo & C. Bastarrica, CLEI Electronic Journal, Vol. 13(1) Special Issue of Best Papers presented at 2nd International Workshop on Advances Software Engineering (IWASE'2009), ISSN 0717- 5000, April 2010

[8] “A Formal Foundation for Transformations in Model Driven Engineering” Thesis Overview. R. S. Giandini. Directora: Claudia Pons. Published in JCS&T (Journal of Computer Science & Technology) Vol. 8 - No. 1 - April 2008 - ISSN 1666-6038, Fac. De Informática, UNLP.

[9] "A two-level formal semantics for the QVT language" Roxana Giandini, Claudia Pons, Gabriela Pérez. Memorias de la XII Conferencia Iberoamericana en "Software Engineering" (CIbSE). Medellín, Colombia del 13 al 17 de Abril de 2009. ISBN: 978-958-44-5028-9, Pag: 73-86

[10] "An Algebraic Approach for Composing Model Transformations in QVT" Claudia Pons, Roxana Giandini, Gabriela Pérez, Gabriel Baum. In Models in Software Engineering, Proceedings of ATEM 2007 - 4th International Workshop on (Software) Language Engineering, at MoDELS 2007 Conference. Nashville, USA. Editor: Springer Berlin/Heidelberg. ISBN 978-3-540-69069-6, ISSN (print) 0302-9743. Pages 28-33, October 2007

[11] "Composición de Transformaciones de Modelos en MDD basada en el Álgebra Relacional" Roxana Giandini, Gabriela Pérez, Claudia Pons. Memorias del X Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software (IDEAS'07) pág: 239-252. Editores: F. Losario, G. Horta Travassos, V. Pelechano, I. Díaz, A. Matteo. Caracas, Venezuela ISBN: 978-980-325-323-3, Isla de Margarita, del 7 al 11 de Mayo 2007

[12] "A Minimal OCL-based Profile for Model Transformation" Roxana Giandini, Gabriela Pérez, Claudia Pons. Publicado en las actas de las VI Jornadas Iberoamericanas de Ingeniería de software e Ingeniería del Conocimiento (JIISIC'07) ISBN: 978-9972-2885-2-4. Lima, Perú, Febrero de 2007.

[13] Curso Tutorial: "Transformaciones de Modelos en MDD: Conceptos teóricos y Lenguajes específicos", dictado en el XIII Congreso Iberoamericano en "Software Engineering" (CIbSE). Cuenca, Ecuador, del 12 al 16 de Abril de 2010. ISBN: 978-9978-325-10-0

[14] "Implementación de MS DSL Tools a herramientas CASE de modelado estándar de software con fundamentos formales (UML-OCL) y su aplicación sobre arquitecturas dirigidas por modelos (MDA)", CNEISI (Congreso Nacional de Estudiantes en Ingeniería de Sistemas de Información) 2008: - C. Pons, R. Giandini, M. Pinasco y L. Nahuel - La Plata, Bs.As., Arg.

[15] "Herramientas de soporte al proceso de desarrollo de software basado en modelos" -

XXXV° JAIIO (Jornadas Argentina de Informática e Investigación Operativa) 2006 - F. Salvatierra, L. Ocaranza y L. Nahuel - Mendoza, Arg. - 3° Puesto en competencia abierta de trabajos estudiantiles.

[16] "Software integrativo para procesos industriales aplicado al campo académico y construido sobre base de datos Open Source y herramientas de cálculo numérico" - CNEISI (Congreso Nacional de Estudiantes en Ingeniería de Sistemas de Información) 2008 - Lucas Garbacz y Leopoldo Nahuel - La Plata, Bs.As., Argentina.

[17] Agile Software Development. Cockburn, Alistair. Editorial Highsmith Series, 2002

[18] Building Model-Driven Engineering Traceability Classifications. Richard F. Paige, Gøran K. Olsen, Dimitrios S. Kolovos, Steffen Zschaler, Christopher Power. ECMDA Traceability Workshop 2008

[19] Survey of Traceability Approaches in Model-Driven Engineering. Ismenia Galvao and Arda Goknil 11th IEEE International Enterprise Distributed Object Computing Conference (2007)

[20] Grand Challenges in Traceability, Center of Excellence for Traceability Jane Huffman Hayes, University of Kentucky, 2009

[21] Systems engineering with SysML/UML: modeling, analysis, design. Tim Weilkiens. Editorial Morgan Kaufmann, 2007.

[22] The object constraint language: getting your models ready for MDA. Jos B. Warmer, Anneke G. Kleppe. Editorial Addison-Wesley Professional, 2° Edition, 2003.

[23] Manifesto for Agile Software Development. Jos B. Warmer, Anneke G. Kleppe. <http://www.agilemanifesto.org/>

[24] Model Driven Architecture. Object Management Group (OMG). <http://www.omg.org/mda/>

[25] OMG Business Process Modeling Notation (BPMN) - Version 1.2 www.omg.org/spec/BPMN/

[26] Weske Mathias, "Business Process Management: Concepts, Languages, Architectures". Springer, Pag 3-67. ISBN 978-3-540-73521-2, 2008.