



INFORME CIENTÍFICO-TECNOLÓGICO

PERIODO: 2010- 2011

Legajo N°: 311214

1. DATOS PERSONALES

APELLIDO: Favre

NOMBRES: Liliana María

Dirección Particular: Calle:

Localidad: Tandil *CP:* 7000

Tel:

Dirección electrónica: lfavre@arnet.com.ar

liliana.favre@gmail.com

lfavre@exa.unicen.edu.ar

2. TEMA DE INVESTIGACIÓN

Tecnologías de software

3. DATOS RELATIVOS A INGRESO Y PROMOCIONES EN LA CARRERA

INGRESO: Categoría: Investigador Adjunto sin director *Fecha:* 7/1998

ACTUAL: Categoría: Investigador Adjunto sin director *desde fecha:* 7/1998

4. INSTITUCIÓN DONDE DESARROLLA LA TAREA

Universidad y/o Centro:

Universidad Nacional del Centro de la Provincia de Buenos Aires- INTIA

Facultad: Facultad de Ciencias Exactas

Departamento: Computación y Sistemas

Dirección: Campus Universitario- Paraje Arroyo Seco s/n

Localidad: Tandil *CP:* 7000 *Tel:* 0249-4439681/82 int 43

Cargo que ocupa: Profesora Asociada Exclusiva

5. DIRECTOR DE TRABAJOS.

Apellido y Nombres:

Dirección Particular: Calle: *N°:*

Localidad: *CP:*

Dirección electrónica:

Firma del Director (si corresponde)

Firma del Investigador

Fecha 15/05/2012

6. EXPOSICIÓN SINTÉTICA DE LA LABOR DESARROLLADA EN EL PERÍODO

La investigación en este período se centró, de acuerdo al plan de actividades propuesto, en la definición de procesos rigurosos de evolución de software en desarrollos dirigidos por modelos.

Esta investigación se basa en la integración del enfoque MDD (Model Driven Development), técnicas clásicas del área de compiladores y técnicas de verificación basadas en especificaciones algebraicas.

Bajo la denominación MDD se define a un amplio rango de desarrollos basados en el uso de modelos como entidades de primera clase. Una realización específica de MDD propuesta por OMG (*Object Management Group*) es la arquitectura MDA (*Model Driven Architecture*). MDA propone separar la especificación de la funcionalidad del sistema de su implementación sobre una plataforma en una tecnología específica y controlar la evolución del software desde modelos abstractos a implementaciones tendiendo a aumentar el grado de automatización. Los procesos de desarrollo en MDA distinguen los siguientes modelos: modelo independiente de la computación (*Computation Independent Model* o CIM), modelo independiente de la plataforma (*Platform Independent Model* o PIM), modelo específico a la plataforma (*Platform Specific Model* o PSM) y modelos específicos a la implementación o código.

Una de las características esenciales de MDA es que todos los artefactos involucrados en un proceso de desarrollo son representados a partir del lenguaje de metamodelado MOF (*Meta Object Facility*). MOF es un meta-metamodelo que define una forma común de capturar todas las construcciones de los estándares de modelado e intercambio que son usadas en MDA y es la esencia de MDA al permitir que diferentes tipos de artefactos provenientes de diferentes vendedores sean usados juntos en un mismo proyecto. Otro concepto fundamental en MDA es el de transformaciones entre modelos. El estándar propuesto por OMG para especificar transformaciones es el metamodelo QVT (*Query, View, Transformation*).

Los resultados más relevantes en este período de investigación están vinculados a la definición de procesos de ingeniería inversa para la recuperación, a partir de código orientado a objetos, de modelos PIM y PSM expresados en UML. Específicamente, se definió la recuperación de modelos de clase, modelos de estado, modelos de secuencia, modelos de actividades y modelos de casos de uso. La validación de resultados se realizó bajo el proyecto de código abierto Eclipse dado que algunos de sus subproyectos proporcionan herramientas y entornos de ejecución alineados con estándares de MDA.

A fin de controlar la evolución del software en procesos basados en MDA, se definieron técnicas de metamodelado formal, particularmente dentro del marco de la evolución de metamodelos MOF se definió un DSL (*Domain Specific Language*) para metamodelado formal.

Los resultados fueron publicados en un libro, revistas, capítulos de libros y conferencias internacionales. Los resultados logrados podrían incorporar funcionalidad a herramientas CASE MDA existentes, representando una innovación a nivel internacional y nacional.

7. TRABAJOS DE INVESTIGACIÓN REALIZADOS O PUBLICADOS EN ESTE PERÍODO

Las publicaciones listadas a continuación muestran resultados originales de investigación. Las publicaciones [1] y [3] son de autoría única por invitación. Las restantes publicaciones muestran resultados de investigación logrados en forma conjunta con integrantes del grupo “Tecnologías de software”. La producción declarada incluye un libro, 2 revistas internacionales, 2 capítulos y 5 publicaciones en *proceedings* de conferencias internacionales. Se incluye en este informe un resumen de cada publicación y se adjunta documentación respaldatoria.

7.1. PUBLICACIONES

LIBROS

[1]

Favre, Liliana (Autora)

“Model Driven Architecture for Reverse Engineering Technologies: Strategic Directions and System Evolution”

Libro. 443 pág. IGI Global, Engineer Science Reference, USA. ISBN 978-1-61520-649-0. EISBN 978-1-61520-650-6. Enero 2010

Abstract

The demand for modernization of legacy systems has created the need for new architectural frameworks that improve interoperability, productivity, process quality, and maintenance costs involved with model driven architecture. *Model Driven Architecture for Reverse Engineering Technologies: Strategic Directions and System Evolution* emphasizes the importance of validation and verification in the modernization of systems that are critical to safety, security, and economic profits. Containing integration analyses to control the evolution of legacy systems towards object oriented technologies, this innovative publication provides critical research for those that have adopted reverse production practices or are considering an investment in system modernization.

REVISTAS

[2] Martinez, Liliana; Pereira, Claudia; Favre, Liliana

“Reverse Engineering Activity Diagram from Object-Oriented Code: An MDA-Based Approach”. *Computer Technology and Application*. ISSN 1934-7532 (versión impresa) ISSN 1934-7340 (versión online). Volume 2, Number 12, Diciembre de 2011.

Abstract

The success of system modernization depends on the existence of technical frameworks for information integration and tool interoperation like the Model Driven Architecture (MDA). Reverse engineering techniques play a crucial role in system modernization.

This paper describes how to reverse engineering activity diagrams from object oriented code in the MDA context focusing on transformations at model and metamodel levels. A framework to reverse engineering MDA models from object oriented code that distinguishes three different abstraction levels linked to models, metamodels and formal specifications, is described. At model level, transformations are based on static and

dynamic analysis. At metamodel level, transformations are specified as OCL (Object Constraint Language) contracts between MOF (Meta Object Facility) metamodels which control the consistency of these transformations. The level of formal specification includes algebraic specifications of MOF metamodels and metamodel-based transformations. This paper analyzes a recovery process of activity diagrams from Java code by applying static and dynamic analysis and shows a formalization of this process in terms of MOF metamodels. The authors validate their approach by using Eclipse Modeling Framework, Ecore metamodels and ATL (Atlas Transformation Language).

CAPÍTULOS EN LIBROS

[3] Favre, Liliana; Martinez, Liliana; Pereira, Claudia

“Software System Modernization: An MDA-Based Approach”

Capítulo 7 del libro “Emerging Technologies for the Evolution and Maintenance of Software Models” (Editores J. Rech, C. Bunse). Pág. 164-199. IGI GLOBAL, 2012. ISBN13: 9781613504383, ISBN10: 1613504381, EISBN13: 9781613504390. DOI 10.4018/978-1-61350-438-3.ch007, IGI Global, USA, 2012.

Abstract

System modernization requires the existence of technical frameworks for information integration and tool interoperability that allow managing new platforms technologies, design techniques, and processes. The Model Driven Architecture (MDA) is aligned with this requirement. It is an evolving conceptual architecture to achieve cohesive model-driven technology specifications. MDA distinguishes the following models: Computation Independent Model (CIM), Platform Independent Model (PIM), and Platform Specific Model (PSM). The integration of classical reverse engineering techniques with the MDA initiative will play a crucial role in software system modernization. In light of these issues, this chapter describes a framework for MDA-based reverse engineering that integrates static and dynamic analysis, meta-modeling, and formal specification. The essential idea is to combine static and dynamic analysis to generate software models (PSM and PIM) from code, and to analyze the consistency of these transformations by using meta-modeling techniques and formal algebraic specification. The chapter shows how to reverse engineer PSM and PIM (that are expressed in terms of UML models) from object oriented code. More specifically, the chapter emphasizes the bases of a reverse engineering approach and describes how to reverse engineer class diagram, state diagram, activity diagram, and use case diagram within the context of MDA initiative.

[4] Favre, Liliana

“MDA-Based Reverse Engineering”

Capítulo 3 del libro “Reverse Engineering: Recent Advances and Applications” ISBN 979-953-307-268-4. Editor Alexandrus C. Telea (Eindhoven University of Technology, Netherlands). ISBN 978-953-51-0158-1. pág. 55-82. Intech Open Access Publisher. 2012.

Abstract

This chapter describes a reverse engineering approach that uses Model-Driven principles, traditional techniques of reverse engineering such as static and dynamic analysis and formal specification. The chapter provides a description of the state-of-the art and practice in

Model Driven Reverse Engineering (MDRE). The core of the chapter describes how to reverse engineer MDA models that represent a view of existing legacy systems from its code. These models provide a uniform representation of the system which conforms to metamodels defined in the context of the standards of ADM (Architecture Driven Modernization). To reason about transformations, metamodels and their evolution we propose to use a formal Domain Specific Model for metamodeling. Finally, challenges and strategic directions in MDRE are included.

Invitada a publicar un capítulo como experta en el libro “Reverse Engineering: Recent Advances and Applications” ISBN 979-953-307-268-4. Editor Alexandrus C. Telea (Eindhoven University of Technology, The Netherlands). ISBN 978-953-51-01581.

CONFERENCIAS INTERNACIONALES

[5] Pereira, Claudia; Martinez, Liliana; Favre, Liliana

“Recovering Use Case Diagrams from Object-Oriented Code: an MDA-based Approach”

Proceedings ITNG 2011, 8th. International Conference on Information Technology: New Generations”, Abril de 2011, Las Vegas, USA. ISBN 978-0-7695-4367-3/11.

DOI 10.1109/ITNG.2011.130, pág. 737-742, IEEE Computer Press, USA.

Abstract

Modernization of legacy systems requires the existence of technical frameworks for information integration and tool interoperability that allow managing new platform technologies, design techniques and processes. The Model Driven Architecture (MDA), adopted by the Object Management Group (OMG), is aligned with this requirement. Reverse engineering techniques play a crucial role in system modernization. In light of these issues, this paper describes a framework to reverse engineering MDA models from object oriented code. This framework distinguishes three different abstraction levels linked to models, metamodels and formal specifications. This paper shows how to reverse engineering use case diagrams from Java code in the MDA context focusing on transformations at model and metamodel levels.

[6] Favre, Liliana; Martinez, Liliana; Pereira, Claudia

“Modernizing Science & Engineering Software Systems”

Proceedings de “IV International Conference on Computational Methods for Coupled Problems in Science and Engineering- COUPLED PROBLEMS 2011”, junio de 2011, pág. 1033-1044, Kos, Grecia.

Abstract

Software industry is evolving and tackling new approaches aligned with Internet, object orientation, distributed components, new modeling languages and new platforms. However, the majority of the large engineering applications running today in many research organizations were developed many years ago with technology that is now obsolete. These old systems, known as legacy systems, are still business-critical. Their complete replacement is dangerous and their maintenance is increasingly expensive. That is the reason for the demand of modernization of legacy system to extend their useful lifetime.

As the demands for modernized legacy systems rise, so does the need for frameworks for information integration and tool interoperability. The Object Management Group (OMG) has adopted the Model Driven Architecture (MDA), which is an evolving

conceptual architecture that aligns with this demand. MDA could help for solving coupling problems of multidisciplinary character in science and engineering that consist of one or more applications, supported by one or more platforms.

The outstanding ideas behind MDA are separating the specification of the system functionality from its implementation on specific platforms, managing the software evolution from abstract models to implementations.

The objective of this paper is to describe rigorous techniques to control the evolution from science & engineering software legacy systems to MDA technologies. We propose a rigorous framework to reverse engineering legacy code in the context of MDA. Considering that validation, verification and consistency are crucial activities in the modernization of systems that are critical to safety, security and economic profits, our approach emphasizes the integration of MDA with formal methods.

[7] Martinez, Liliana; Pereira, Claudia; Favre, Liliana

“Recovering Activity Diagrams from Object-Oriented Code: an MDA-based Approach”
Proceedings de “2011 International Conference on Software Engineering Research and Practice”, SERP 2011, Julio de 2011, Las Vegas, ISBN 1-60132-199-6, pág. 67-73, CSREA Press, USA.

Abstract

The success of system modernization depends on the existence of technical frameworks for information integration and tool interoperability like the Model Driven Architecture (MDA). Reverse engineering techniques play a crucial role in system modernization. This paper describes how to reverse engineering activity diagrams from object oriented code in the MDA context focusing on transformations at model and metamodel levels. A framework to reverse engineering MDA models from object oriented code that distinguishes three different abstraction levels linked to models, metamodels and formal specifications, is described. At model level, transformations are based on static and dynamic analysis. At metamodel level, transformations are specified as OCL contracts between MOF (Meta Object Facility) metamodels which control the consistency of these transformations. The level of formal specification includes algebraic specifications of MOF metamodels and metamodelbased transformations. This paper analyzes a recovery process of activity diagrams from Java code by applying static and dynamic analysis and show a formalization of this process in terms of MOF metamodels.

[8] Favre, Liliana

“Foundations for QVT Transformation”

Proceedings 2010 International Conference on Software Engineering Research and Practice (SERP 2010), Las Vegas, USA, ISBN 1-60132-160-0, pág. 58-64, CSREA Press, USA.

Abstract

This paper includes foundations for transformations within the context of the Model Driven Architecture (MDA). The main contribution of this paper is the identification and formalization of an MDA kernel called “MDA Infrastructure” (MDA-I). It includes a set of packages of the following OMG standards: UML Infrastructure, OCL, and the metamodels MOF (Meta-Object Facility) and QVT (Query, View, Transformation). We show that one could reason about QVT transformations in terms of the formalization of MDA-I. It is precisely defined in a way independent of imperative constructs and can be

automatically translated to traditional algebraic languages. We propose to use an MDA-I formalization as a “formal kernel” to control QVT transformations.

[9] Favre, Liliana

“Model Driven Architecture for Round-Trip Engineering: A Conceptual Framework”
Proceedings 2010 International Conference on Software Engineering Research and Practice (SERP 2010), Las Vegas, USA, ISBN 1-60132-160-0, pág. 147-153, CSREA Press, USA.

Abstract

This paper describes a conceptual framework for round-trip engineering within the context of the Model Driven Architecture (MDA). We propose a roundtrip engineering process that is based on the integration of static and dynamic analysis with semiformal and formal metamodeling techniques. Semiformal metamodels are expressed in terms of an MDA kernel including a subset of MOF (Meta Object Facility) metamodel, QVT (Query, View, Transformation) metamodel, OCL and UML. We analyze a formalization of metamodels by using algebraic formal languages. The paper discusses foundations of innovative MDA round-trip engineering processes, challenges and strategic directions that can be adopted for extending the functionality of the existent MDA CASE tools.

7.2. TRABAJOS ACEPTADOS PARA SU PUBLICACIÓN

[10] Pereira, Claudia; Martinez, Liliana; Favre, Liliana

“Recovering Use Case Diagrams from Object-Oriented Code: an MDA-based Approach”. International Journal of Software Engineering (IJSE). ISSN 1687-6954 (versión impresa) ISSN 2090-1801 (versión online) Enviado Agosto 2011- Aceptado, febrero 2012

Abstract

Modernization of legacy systems requires the existence of technical frameworks for information integration and tool interoperability that allow managing new platform technologies, design techniques and processes. MDA (Model Driven Architecture), adopted by the OMG (Object Management Group), is aligned with this requirement. Reverse engineering techniques play a crucial role in system modernization. In light of these issues, this article describes a framework to reverse engineering MDA models from object oriented code. This framework distinguishes three different abstraction levels linked to models, metamodels and formal specifications. At model level, transformations are based on static and dynamic analysis. At metamodel level, transformations are specified as OCL (Object Constraint Language) contracts between MOF (Meta Object Facility) metamodels which control the consistency of these transformations. The level of formal specification includes algebraic specifications of MOF metamodels and metamodel-based transformations. This article shows how to reverse engineering use case diagrams from Java code in the MDA context focusing on transformations at model and metamodel levels. We validate our approach by using Eclipse Modeling Framework, Ecore metamodels and ATL (Atlas Transformation Language).

7. 3. TRABAJOS ENVIADOS Y AÚN NO ACEPTADOS PARA SU PUBLICACIÓN

7.4. TRABAJOS TERMINADOS Y AÚN NO ENVIADOS PARA SU PUBLICACIÓN

Se está redactando un libro de autoría que presentará resultados originales para la ingeniería inversa de código C++ y Java a modelos expresados en UML, enfatizando en aspectos de implementación.

7.5. COMUNICACIONES

7.6. INFORMES Y MEMORIAS TÉCNICAS

8. TRABAJOS DE DESARROLLO DE TECNOLOGÍAS

8.1. DESARROLLOS TECNOLÓGICOS

8. 2. PATENTES O EQUIVALENTES

8.3. PROYECTOS POTENCIALMENTE TRANSFERIBLES NO CONCLUÍDOS Y QUE ESTÁN EN DESARROLLO

Se están desarrollando dos proyectos potencialmente transferibles.

Por un lado, como validación de los resultados logrados en el proyecto se está desarrollando un prototipo de herramienta, integrado dentro del IDE Eclipse, para la ingeniería inversa desde código Java a modelos PIM y PSM expresados en UML.

Otro proyecto con posibilidad de transferencia está vinculado a un Framework, GLIESE, para desarrollo de juegos (ver 10.1). Se prevé adaptarlo a partir del enfoque y resultados de esta investigación para el desarrollo de video juegos en móviles. Se han establecido contactos con una empresa tandilense (Dokko Group) que desarrolla video juegos para dispositivos móviles a fin de establecer un marco de colaboración.

8.4. OTRAS ACTIVIDADES TECNOLÓGICAS CUYOS RESULTADOS NO SEAN PUBLICABLES

9. SERVICIOS TECNOLÓGICOS

10. PUBLICACIONES Y DESARROLLOS EN:

10.1. DOCENCIA

Desarrollos en docencia

Se desarrolló el *framework* GLIESE que provee la infraestructura básica para la generación de juegos a partir de una metodología de programación centrada en la reutilización de componentes. La idea es que un alumno de cursos de diseño de algoritmos pueda generar aplicaciones a partir de componentes concentrándose en aspectos esenciales vinculados a técnicas clásicas de diseño de algoritmos, y antes de tener un significativo conocimiento de arquitecturas de software, programación orientada a objetos, redes, sistemas distribuidos, etc. El *framework* facilita el desarrollo de juegos, la experimentación con diferentes técnicas de diseño (por ejemplo *backtracking* o *greedy*). Se experimentó con el Framework en la cursada de la materia Análisis y diseño de algoritmos II dictada en el segundo año de la carrera de Ingeniería de Sistemas (UNCPBA).

10.2. DIVULGACIÓN

11. DIRECCIÓN DE INVESTIGADORES EN PROYECTOS ACREDITADOS

Durante los años 2010 y 2011 dirigí el proyecto de incentivos 03/C217.

PROYECTO DE INCENTIVO 03/C217

Tecnologías de Software

Universidad Nacional del Centro de la Pcia. de Buenos Aires. 2010-2011-2012

El objetivo del proyecto es integrar técnicas orientadas al cliente, semiformales y formales en procesos de reingeniería basados en la arquitectura Model- Driven (Model-Driven Architecture - MDA). La investigación se concentra en dos líneas:

Línea 1: Procesos rigurosos de reingeniería de software basados en MDA

Línea 2: Técnicas de reingeniería para la integración de modelos de requisitos en desarrollos de software basados en MDA.

El objetivo de la Línea 1 es integrar técnicas semiformales y formales en desarrollos centrados en modelos (Model-Driven Development-MDD). Se definirán procesos de ingeniería directa (forward engineering), ingeniería inversa (reverse engineering) y en general de reingeniería de software a fin de extender la funcionalidad de las herramientas CASE (Computer Aided Software Engineering) basadas en MDA.

El objetivo de la Línea 2 es la integración de modelos de requisitos a través de procesos de reingeniería de software en el contexto de MDA. Se propondrán transformaciones entre modelos utilizando técnicas de reingeniería que abarquen tanto ingeniería directa como inversa y nuevos modelos de requisitos. Los resultados de las dos líneas serán integrados con el objetivo de definir estrategias que complementen notaciones y procesos a fin de permitir la reingeniería de sistemas heredados (*legacy*) para incorporar tecnología acorde a las necesidades actuales de las empresas.

Personal Participante

Apellido y Nombre	Categoría Investigador 31/12/2011	Cargo Docente Dedicación 31/12/2011	Proyecto 03/C217
Favre, Liliana María	I	Asociado Exclusiva	Directora
Felice, Laura Mabel	IV	Adjunto Exclusiva	Integrante
Leonardi, María Carmen	III	Adjunto Exclusiva	Integrante
Martinez, Liliana Inés	IV	JTP Exclusiva	Integrante
Mauco, María Virgina	III	Adjunto Exclusiva	Codirectora
Pereira, Claudia Teresa	IV	JTP Exclusiva	Integrante

Dirección de becarios graduados

Becario de estudio CIC.

Ing. Federico Améndola

Tema: “Desarrollo de software dirigido por modelos y aplicaciones móviles”

Directora: Liliana Favre

Dirección de alumnos de grado

Fernández, Martín (tesista)

Galitiello, Emiliano (tesista)

Rolandi, Belén (tesista)

12. DIRECCIÓN DE TESIS

Tesis de grado

“GLIESE: Un framework para el desarrollo de juegos aplicado a la enseñanza de algoritmos”

Federico Améndola, Martín Fernández

Trabajo de tesis de la carrera de Ingeniería de Sistemas. Facultad de Ciencias Exactas. Universidad Nacional del Centro de la Pcia. de Buenos Aires. Presentado. Calificación 10 (diez).

Directora: Liliana Favre

“Ingeniería inversa en el contexto de MDA (Model Driven Architecture)”

Emiliano Galitiello, Belén Rolandi

Trabajo de tesis de la carrera de Ingeniería de Sistemas. Facultad de Ciencias Exactas. Universidad Nacional del Centro de la Pcia. de Buenos Aires. Plan aprobado, en desarrollo.

Directora: Liliana Favre

13. PARTICIPACIÓN EN REUNIONES CIENTÍFICAS

14. CURSOS DE PERFECCIONAMIENTO, VIAJES DE ESTUDIO, ETC.

15. SUBSIDIOS RECIBIDOS EN EL PERÍODO

Las investigaciones realizadas en el Proyecto "Tecnologías de Software" han sido subsidiadas por la Secretaría de Ciencia, Arte y Tecnología de la Universidad Nacional del Centro (SECAT) a través del Proyecto INTIA. El proyecto incluye a 6 integrantes y el monto asignado durante el período 2010-2011 fue de aproximadamente \$5000 anuales.

16. DISTINCIONES O PREMIOS OBTENIDOS EN EL PERÍODO.

17. ACTUACIÓN EN ORGANISMOS DE PLANEAMIENTO, PROMOCIÓN O EJECUCIÓN CIENTÍFICA Y TECNOLÓGICA

Integrante titular del Consejo de Control de Gestión de todos los núcleos de actividades científico-tecnológicas (NACT) vinculadas a la Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires. Res. 035/2010. Designación desde 26/2/2010 al 31/5/2012.

18. TAREAS DOCENTES DESARROLLADAS EN EL PERÍODO

Durante los años 2010 y 2011 desempeñé tareas docentes en el Departamento de Computación y Sistemas, (Facultad de Ciencias Exactas. UNCPBA) en las siguientes asignaturas:

Profesora a cargo de la materia *Análisis y diseño de algoritmos I* de la carrera de Ingeniería de Sistemas, Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires.

Profesora a cargo de la materia *Análisis y diseño de algoritmos II* de la carrera de Ingeniería de Sistemas, Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires.

Profesora a cargo de la materia optativa *Arquitecturas de software dirigidas por modelos* de la carrera de Ingeniería de Sistemas, Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires.

Profesora a cargo de la materia optativa *Técnicas de ingeniería inversa en el contexto de arquitecturas de software dirigidas por modelos* de la carrera de Ingeniería de Sistemas, Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires.

Dedicación docente aproximada: 10 hs semanales

19. OTROS ELEMENTOS DE JUICIO NO CONTEMPLADOS EN LOS TÍTULOS ANTERIORES

Integrante del Advisory Board de la revista internacional “Journal of Cases on Information Technology” (JCIT), Information Resources Management Resources Association, ISSN 1048-7717, USA.

Miembro integrante de la Comisión de Acreditación de la carrera de Ingeniería de Sistemas de la Facultad de Ciencias Exactas (UNCPBA). Res. CA 04/10, 2010

Evaluada de nuevos proyectos e informes de avance del período 2008-2009 del Programa de Incentivos. Universidad de San Luis. Res. 65/10-CS. 2010.

Evaluada de presentaciones de Becas de Doctorado y Maestría. 2da convocatoria 2010. Res 186/10 Consejo Superior. Universidad Nacional de La Pampa.

Evaluada de informes de avance de Becas de Doctorado y Maestría Convocatoria 2009. Res 185/10 Consejo Superior. Universidad Nacional de La Pampa.

Evaluada de informes pendientes de avance de Becas de Doctorado y Maestría Convocatoria 2009. Res 242/10 Consejo Superior. Universidad Nacional de La Pampa..

Integrante de la Comisión Evaluadora de Becas de Postgrados. Universidad Nacional de La Pampa, 20 de mayo de 2011

Evaluada de capítulos del libro “Emerging Technologies for the Evolution and Maintenance of Software Models” (Editores J. Rech, C. Bunse) . IGI Global, USA. 2011

Evaluada de trabajos del Workshop de Ingeniería de Software del CACIC 2011 (XVII Congreso Argentino de Ciencias de la Computación), La Plata.

20. TÍTULO Y PLAN DE TRABAJO A REALIZAR EN EL PRÓXIMO PERÍODO

TÍTULO: TECNOLOGÍAS DE SOFTWARE

RESUMEN

Gran parte de los sistemas de información vitales en organizaciones de nuestro medio fueron implementados hace varios años con tecnologías que hoy pueden considerarse obsoletas, y no alineadas con los actuales objetivos estratégicos de las organizaciones. Estos sistemas, conocidos como *legacy* (heredados), involucran software, hardware, procesos de negocio y, políticas y estrategias organizacionales. En general no están documentados o, si lo están, sus especificaciones no reflejan los cambios de requerimientos que se dieron a través de los años englobando sólo el código la historia de su evolución. La entrada en escena de nuevas tecnologías relacionadas por ejemplo a aplicaciones web y móviles motivan una creciente demanda de modernización de sistemas a fin de lograr ventajas competitivas para las organizaciones. Nuevos enfoques de desarrollo de software dirigidos por modelos como la arquitectura MDA (*Model*

Driven Architecture) podrían dar respuesta a esta demanda, sin embargo, no están lo suficientemente difundidos en nuestro medio que en general opta por nuevos desarrollos que no son rentables y conllevan un alto riesgo. El objetivo de esta investigación es definir rigurosas técnicas de evolución de software basadas en la arquitectura MDA, técnicas clásicas de análisis estático y dinámico y especificaciones formales. Se pretende controlar la evolución de sistemas *legacy* hacia nuevas o actuales tecnologías potenciando la productividad mediante la definición de las bases para el desarrollo de herramientas que automaticen actividades esenciales en estos procesos.

PALABRAS CLAVES: MDA; UML; Ingeniería directa; Ingeniería inversa; Reingeniería de software; Evolución de software; Meta-modelado; Especificación formal

PLAN DE ACTIVIDADES

Período: 2012-2013

1. Introducción

MDD define a un amplio rango de desarrollos basados en el uso de modelos como entidades de primera clase. Una realización específica de MDD propuesta por OMG (*Object Management Group*), es la arquitectura MDA (*Model Driven Architecture*) [16].

La definición inicial de MDA tuvo que ver con el problema de interoperabilidad de *middleware* en internet. Más allá de resolver problemas de interoperabilidad, MDA ofrece otros beneficios vinculados a productividad, calidad de procesos y costos de mantenimiento. MDA no es en sí misma una especificación de una tecnología sino un plan para lograr especificaciones de tecnologías cohesivas y centradas en modelos. MDA propone separar la especificación de la funcionalidad del sistema de su implementación sobre una plataforma en una tecnología específica y controlar la evolución del software desde modelos abstractos a implementaciones tendiendo a aumentar el grado de automatización.

Los procesos de desarrollo en MDA distinguen los siguientes modelos: modelo independiente de la computación (*Computation Independent Model* o CIM), modelo independiente de la plataforma (*Platform Independent Model* o PIM), modelo específico a la plataforma (*Platform Specific Model* o PSM) y modelo específico a la implementación (*Implementation Specific Model* o ISM). La difusión inicial de MDA enfatizaba su relación con UML como lenguaje de modelado, sin embargo hay usuarios de UML que no usan MDA y usuarios de MDA que usan otros lenguajes de modelado como específicos DSL (*Domain Specific Language*) [22] [33].

Una de las características esenciales de MDA es que todos los artefactos involucrados en un proceso de desarrollo se representan a partir del lenguaje de metamodelado MOF (*Meta Object Facility*) [20]. MOF es un meta-metamodelo que define una forma común de capturar todas las construcciones de los estándares de modelado e intercambio que son usadas en MDA y es la esencia de MDA al permitir que diferentes tipos de artefactos provenientes de diferentes vendedores sean usados juntos en un mismo proyecto.

Otro concepto fundamental en MDA es el de transformaciones entre modelos. Una transformación es la especificación de mecanismos para convertir elementos de un modelo en elementos de otro modelo. El estándar propuesto por OMG para especificar transformaciones es el metamodelo QVT (*Query, View, Transformation*) [23].

Actualmente existe una creciente demanda de soporte para la modernización de sistemas y en particular, para la ingeniería inversa de sistemas que deben adaptarse a nuevas tecnologías o plataformas. OMG está involucrado en el desarrollo de estándares para la modernización de sistemas a través de la iniciativa ADM (Architecture Driven Modernization) [1]. Modisco (Modeling Discovery) [19] es un componente GMT (Eclipse Generative Modeling Technology) [8] para la ingeniería inversa dirigida por modelos que puede considerarse la implementación oficial de estándares de ADM para la modernización de sistemas, en particular para ingeniería inversa. Uno de los estándares implementados es el metamodelo KDM (Knowledge Discovery Model) [14] que garantiza interoperabilidad e intercambio de datos entre diferentes herramientas de modernización. Actualmente Modisco sólo soporta la ingeniería inversa de diagramas de clase desde código Java.

Las CASE MDA comerciales proveen un soporte incipiente para la modernización de sistemas y en particular para procesos involucrados como los de ingeniería inversa. Como ejemplos de herramientas comerciales podemos mencionar a Together Architect de Borland, Enterprise Architect de Spark Systems y Rational Software Architect de IBM [5]. Estas herramientas están integradas a Eclipse y sus principales limitaciones están vinculadas a la falta de soporte de transformaciones basadas en el estándar QVT (en particular QVT Declarative) [23] y la limitada especificación de plataformas y puentes entre plataformas.

Si bien las CASE MDA proveen estructuras y procesos para manipular artefactos de modelamiento tales como modelos, metamodelos y transformaciones, no proveen funcionalidad para administrar cambios durante su evolución. Por ejemplo, cuando un metamodelo es modificado, los modelos que lo conforman deberían migrarse para conformar a la versión modificada. Resolver este problema, referido como coevolución de metamodelos, es uno de los grandes desafíos en MDD.

En [28] se presentan relevantes resultados para la ingeniería inversa de código orientado a objetos a fin de extraer modelos para sistemas cuya única documentación confiable es el código. Esta propuesta se basa en técnicas de análisis estático y análisis dinámico [6]. Las técnicas propuestas son el resultado de un trabajo de colaboración con el CERN (Conseil Européen pour la Recherche Nucleaire) en el proyecto “Large Hadron Collider”. Uno de los resultados fue la implementación de la herramienta RevEng para la ingeniería inversa de código C a diagramas UML. A través de estos desarrollos fue mostrado que la propuesta puede ser aplicada a escala.

Teniendo en cuenta lo anteriormente expuesto se propone como objetivo general de este trabajo analizar procesos de evolución de software basados en la combinación del enfoque MDA y técnicas relacionadas (como metamodelado MOF), técnicas de ingeniería inversa tradicionales de análisis estático y dinámico [28] y técnicas de verificación basadas en el formalismo algebraico. En esta investigación se propone validar la propuesta en desarrollos de aplicaciones móviles y desarrollo de juegos de video.

2. Metodología

Para lograr los objetivos del proyecto, se propone integrar técnicas semiformales y formales de metamodelado con técnicas clásicas de compiladores. Las técnicas semiformales de metamodelado se basarán en metamodelos MOF y las técnicas formales en lenguajes formales algebraicos. Con respecto a las técnicas del área de compiladores, se propone usar técnicas de análisis estático y dinámico para la definición tanto de ingeniería inversa y como de directa.

Se propone validar la propuesta con la ingeniería inversa de aplicaciones de escritorio en Java o C++ a modelos MDA. La validación se realizará bajo el proyecto de código abierto Eclipse dado que algunos de sus subproyectos proporcionan herramientas y entornos de ejecución alineados con estándares de MDA. Cabe destacar que, en particular, EMF (Eclipse Modeling Framework) se inicia como una implementación del estándar MOF, resultando el metamodelo Ecore. EMF provee infraestructura para el desarrollo de herramientas CASE MDA (editores, motores de transformación y soporte para metamodelado). Otro subproyecto Eclipse es M2M que soporta transformaciones entre modelos [8]. Dentro del ámbito de Eclipse también se han creado motores de transformación como ATL (Atlas Transformation Language) [2] y QVT Operational y se avanza en la implementación de QVT Declarative [23].

3. Actividades propuestas para el período

El plan de investigación propone dar continuidad a la investigación desarrollada durante los años 2010-2011 en el proyecto “Tecnologías de software”. Tal como en el período previo se investigarán formalismos y teorías para analizar, administrar y controlar la evolución del software en el contexto de MDA. Por otra parte, se dará continuidad a la coordinación del desarrollo de prototipos de herramientas para validar los resultados logrados. Se prevé la realización de estos desarrollos a través de la colaboración de tesis de grado, pasantes, becarios o auxiliares de docencia participantes en el proyecto.

Con respecto a la investigación básica, se trabajará específicamente en este período en la evolución de metamodelos MOF en el marco provisto por el formalismo algebraico. Se analizará la evolución de artefactos de desarrollos MDA descritos por metamodelos MOF y se definirán procesos para administrar la evolución de los mismos. Este análisis se basará en resultados previos de este proyecto, en particular en la definición de un DSL (Domain Specific Language) formal para metamodelado cuya semántica fue dada por traducción a CASL.

Asimismo, se continuará con la definición de procesos de ingeniería inversa en el marco de los estándares de OMG. Se definieron procesos de ingeniería inversa que integran técnicas clásicas basadas en análisis estático y dinámico con técnicas basadas en MDA, como por ejemplo metamodelado y lenguajes de transformación como ATL. Los resultados más relevantes están vinculados a la recuperación, a partir de código orientado a objetos, de modelos PIM y PSM expresados mediante diagramas UML. Específicamente, se definió la recuperación de modelos de clase, modelos de estado, modelos de secuencia, modelos de actividades y modelos de casos de uso. Los logros obtenidos en el período previo en cuanto a la definición de procesos de ingeniería inversa, permiten que en esta etapa el énfasis esté puesto en concretar el desarrollo de un prototipo de herramienta para la ingeniería inversa de código Java a modelos PSM y PIM expresados a partir de modelos UML y la evaluación de la propuesta particularmente respecto a productividad. Se adaptarán técnicas fundamentales en procesos de evolución de software como *refactoring*, patrones de diseño y *traceability*.

Se propone desarrollar en este contexto aplicaciones en el dominio específico de aplicaciones móviles. Se prevén otras actividades relacionadas al tema propuesto en una beca de estudio CIC que integrará las bases de esta investigación en el desarrollo de un CRM (*Customer Relationship Management*). Desde el punto de vista de sistemas de software podemos definir un CRM como un software para la administración de la relación con los clientes. Es decir, sistemas informáticos que apoyan y guían la relación

con los clientes , administrando por ejemplo las ventas, los eventos ocurridos como llamados telefónicos, mensajes de correo, presupuestos enviados y las tareas de los diferentes usuarios.

Otro dominio de aplicación a considerar será el dominio de video juegos. Cabe destacar que en el marco de este proyecto se ha desarrollado el Framework GLIESE (ver sección 10.1 de este informe) y se prevé adaptarlo para la generación de juegos en dispositivos móviles.

4. Aportes académicos y de transferencia

Se desarrollarán actividades de formación de investigadores así como de capacitación de recursos humanos en esta temática. Actualmente se dictan cursos optativos en la carrera de “Ingeniería de sistemas” de la UNCPBA vinculados al proyecto. Se prevé continuar con el dictado de estos cursos y otros nuevos que aporten a la capacitación de recursos humanos.

Los resultados de esta investigación podrían ser transferidos. Los resultados ya logrados en esta investigación posibilitarían el desarrollo de metodologías de desarrollo de software y herramientas CASE basadas en MDA cuyas características tecnológicas difieran de las existentes y que representen una innovación a nivel internacional y nacional. El grado de avance en el tema permitiría aportar a la modernización de sistemas de software existentes en nuestro medio.

Se prevé a través de la realización de una beca de estudio trabajar en el desarrollo de aplicaciones móviles con miras a una probable transferencia. Por otro parte, se han establecido contactos con una empresa tandilense que desarrolla video juegos para plataformas móviles a fin de definir un marco de colaboración.

Los resultados logrados en este proyecto permitirían automatizar actividades esenciales en el desarrollo de software. La automatización puede tener un alto impacto en países con las características demográficas de Argentina, con escasos recursos humanos, que no pueden reproducir modelos exitosos de desarrollo de software como la India o China. El uso de herramientas que, a través de la automatización, potencien una productividad de calidad y volumen puede generar grandes ventajas competitivas.

5. Referencias y bibliografía

1. ADM (2010). ADM Task Force. Standards Roadmap. adm.omg.org
2. ATL (2012). ATL Documentation. www.eclipse.org/m2m/atl/documentation
3. Bidoit, M.; Mosses, P. (2004). CASL User Manual - Introduction to Using the Common Algebraic Specification Language, Lecture Notes in Computer Science 2900, Springer.
4. Canfora, G.; Di Penta, M. (2007). New Frontiers of Reverse Engineering. Future of Software Engineering. In Proceedings of Future of Software Engineering (FOSE 2007) págs. 326-341. Los Alamitos: IEEE Press.
5. CASE MDA (2011). www.modelnet.net/mda_tools.html
6. Cichetti, A.; Di Ruscio, D.; Eramo, R.; Pierantonio, A. (2008). Automating co-evolution in model-driven engineering. Proc. International IEEE Enterprise Distributed Object Computing Conference (EDOC), págs. 222-231, IEEE Press.
7. Dunkel, J. ; Brun, R. (2007). Model-Driven Architecture for Mobile Applications. Lecture Notes in Computer Science 4439, págs. 464-477, Springer.
8. Eclipse (2012). The eclipse modeling framework. <http://www.eclipse.org/emf/>

9. Ernst, M. (2003). Static and Dynamic Analysis: Synergy and duality. In Proceedings of ICSE Workshop on Dynamic Analysis (WODA 2003) pág. 24-27.
10. Favre, L.; Martinez, L.; Pereira, C. (2011). Software System Modernization: An MDA-Based Approach. Chapter 7. "Emerging Technologies for the Evolution and Maintenance of Software Models" (Editores J. Rech, C. Bunse) . IGI Global, USA.
11. Favre, L. Model Driven Architecture for Reverse Engineering Technologies: Strategic Directions and System Evolution.(2010). IGI Global, Engineer Science Reference, USA. ISBN 978-1-61520-649-0. EISBN 978-1-61520-650-6. 443 páginas.
12. Favre, L.; Martinez, L.; Pereira, C. (2012) Software System Modernization: An MDA Based Approach. Capítulo 7 del libro "Emerging Technologies for the Evolution and Maintenance of Software Models" (Editores J. Rech, C. Bunse). pág. 164-199. IGI GLOBAL.
13. Favre, L. (2012). MDA-Based Reverse Engineering. Capítulo 3 del libro "Reverse Engineering: Recent Advances and Applications" ISBN 979-953-307-268-4. Editor Alexandrus C. Telea (Eindhoven University of Technology, Netherlands). ISBN 978-953-51-01581. pág. 55-82. Intech Open Access Publisher.
14. KDM (2012). Knowledge Discovery Meta-Model, Version 1.3-beta 2, March 2011. OMG specification formal 2010-12-12. <http://www.omg.org/spec/kdm/1.3/beta2/pdf>
15. Maoz, S.; Harel, D. (2010). On Tracing Reactive Systems. Software & System Modeling. DOI 10.1007/510270-010-0151-2, Springer-Verlag.
16. MDA. (2005). The Model Driven Architecture. www.omg.org/mda.
17. Medini (2012). Medini QVT. <http://projects.ikv.de/qvt>
18. Mens, T.; Demeyer, S. (2007) Software Evolution.Springer-Verlag. Berlín.
19. Modisco (2012). <http://www.eclipse.org/Modisco>
20. MOF. (2006). MOF: Meta Object Facility (MOF TM) 2.0. OMG Specification formal/2006-01-01. www.omg.org/mof
21. OCL (2010). OCL: Object Constraint Language. Version 2.2. OMG: formal/2010-02-01. www.omg.org
22. Parr, T (2007). The Definitive ANTLR Reference: Building Domain-Specific Language. Pragmatic Programmers.
23. QVT (2008). QVT: MOF 2.0 Query, View, Transformation. Formal/2008-04-03. www.omg.org
24. Rech, J.; Bunse, C. (2012). Emerging Technologies for the Evolution and Maintenance of Software Models. IGI GLOBAL.
25. Rose, L. M. (2011). Structures and Processes for Managing Model-Metamodel Co-evolution. PhD. Thesis University of York
26. Systa, T. (2000). Static and Dynamic Reverse Engineering Techniques for Java Software Systems. Ph.D Thesis, University of Tampere, Report A-2000-4.
27. Thompson, C. Schmidt, D.C.; Turner H. A.; White, J. (2011). Analyzing Mobile Application Software Power Consumption via Model-driven Engineering. PECCS 2011 - Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems, SciTePress 2011, pág. 101-113
28. Tonella, P.; Potrich, A. (2005). Reverse Engineering of Object Oriented Code. Monographs in Computer Science. Heidelberg: Springer-Verlag.
29. UML. (2010a). Unified Modeling Language: Infrastructure. Version 2.3. OMG Specification formal/ 2010-05-03. www.omg.org.
30. UML. (2010b). UML: Unified Modeling Language: Superstructure. Version 2.3. OMG Specification: formal/2010-05-05. www.omg.org
31. Vermolen, S.; Wachsmuth, G.; Visser, E. (2011). Reconstructing Complex Metamodel Evolution. SLE 2011, Lecture Notes in Computer Science 6940, Springer, 201-22

32. Winkler, S.; von Pilgrim, J. (2010). A survey of traceability in requirements engineering and model-driven development. *Software and System Modeling*, 9:529-565.
33. Xtext (2012). XText Documentation. www.eclipse.org/Xtext/documentation.