

CARRERA DEL INVESTIGADOR CIENTÍFICO Y TECNOLÓGICO

Informe Científico

PERIODO: 1/1/2014 al 31/12/2015

1. DATOS PERSONALES

APELLIDO: Favre

NOMBRES: Liliana María

Dirección Particular: Calle:

Localidad: Tandil CP: 7000 Tel:

Dirección electrónica: 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: Julio de 1998

ACTUAL: Categoría: Investigador Adjunto sin director desde Julio de 1998

4. INSTITUCIÓN DONDE DESARROLLA LA TAREA

Universidad: Universidad Nacional del Centro de la Provincia de Buenos Aires

Facultad: Ciencias Exactas

Departamento: Computación y Sistemas

Cátedra:

Otros:

Dirección: Calle: Campus Universitario. Paraje Arroyo Seco N°: -

Localidad: Tandil CP: 7000 Tel: 249-154466861

Cargo que ocupa: Profesora Asociada Exclusiva

5. DIRECTOR DE TRABAJOS. (En el caso que corresponda)

Apellido y Nombres:

Dirección Particular: Calle: N°:

Localidad: CP: Tel:

Dirección electrónica:

.....
Firma del Director (si corresponde)

.....
Firma del Investigador

6. RESUMEN DE LA LABOR QUE DESARROLLA

Desempeño tareas de investigación en la Universidad Nacional del Centro de la Provincia de Buenos Aires. Mi cargo actual es Profesora Asociada Exclusiva por concurso y mi categoría en el Programa de Incentivos es 1. Durante los años 2014 y 2015 desempeñé tareas de investigación en el proyecto “Tecnologías de software” acreditado en el Programa de Incentivos a docentes-investigadores e identificado con el código 03/C249. Actualmente dicho proyecto fue acreditado para el período 2016-2017 bajo el código 03/C266. Mis tareas de investigación las desarrollo en la línea “Modernización de Software”. El objetivo de la misma es analizar procesos de modernización de software basados en la integración de desarrollos dirigidos por modelos (MDD- *Model Driven Development*), técnicas de ingeniería inversa tradicionales de análisis estático y dinámico, y técnicas de verificación basadas en el formalismo algebraico. Se analizan diferentes escenarios de modernización para adaptar software a nuevas tecnologías. Las actividades desarrolladas como Investigadora Adjunta en CICPBA se enmarcaron en el proyecto “Tecnologías de software” de acuerdo al plan propuesto en el informe previo.

7. EXPOSICIÓN SINTÉTICA DE LA LABOR DESARROLLADA EN EL PERÍODO.

La investigación realizada durante el período 2014-2015 se centró en la definición de procesos rigurosos de modernización de software en desarrollos dirigidos por modelos (MDD- *Model Driven Development*). MDD define 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 no es en sí misma una especificación de una tecnología sino un plan para lograr especificaciones centradas en modelos y cohesivas de diferentes tecnologías. MDA propone separar la especificación de la funcionalidad del sistema de su implementación sobre una plataforma en una tecnología específica y controlar así la evolución del software tendiendo a aumentar el grado de automatización. Una de las características esenciales de este enfoque es que todos los artefactos involucrados en un proceso de desarrollo son representados a partir del lenguaje de metamodelado MOF (*Meta Object Facility*) que puede verse como la forma común de capturar todas las construcciones de los estándares de modelado e intercambio que son usadas en MDA permitiendo la interoperabilidad entre diferentes tipos de artefactos provenientes de diversas tecnologías.

Actualmente existe una creciente demanda de soporte para la modernización de sistemas y OMG está involucrado en el desarrollo de estándares a través de la iniciativa ADM (*Architecture Driven Modernization*) para la modernización de software. ADM sigue los principios esenciales de MDA: todos los artefactos involucrados en el mismo pueden verse como modelos que conforman a un metamodelo, el proceso en sí mismo puede verse como una secuencia de transformaciones entre modelos y la información recuperada es representada en forma estándar a partir de metamodelos.

Se propuso como objetivo de esta investigación definir procesos rigurosos de modernización de software basados en tres pilares: desarrollos dirigidos por modelos, técnicas tradicionales de análisis estático y dinámico para ingeniería inversa y técnicas de verificación basadas en el

formalismo algebraico. En esta dirección, se investigaron formalismos y teorías para analizar, administrar y controlar la modernización de software y se coordinó el desarrollo de prototipos de herramientas para validar los resultados alcanzados.

Los resultados más relevantes de este período de investigación están vinculados a la definición de procesos de modernización de software C/C++ y Java a fin de adaptarlo a diversas plataformas móviles. Se desarrollaron prototipos de herramienta para la ingeniería inversa de código Java a fin de extraer modelos UML y para la migración de software C/C++ o Java a plataformas móviles. Asimismo, se especificaron metamodelos para los lenguajes C/C++ y Haxe alineados a los estándares de MDA y se desarrollaron herramientas de metamodelado formal que proveen soporte para la validación de metamodelos MOF.

Los principales resultados logrados en este período fueron publicados en revistas, capítulos de libros y actas de conferencias internacionales y nacionales.

8. TRABAJOS DE INVESTIGACIÓN REALIZADOS O PUBLICADOS EN ESTE PERÍODO.

Las publicaciones listadas a continuación muestran resultados originales de investigación logrados conjuntamente con integrantes del grupo “Tecnologías de software” que dirijo en la Facultad de Ciencias Exactas de la Universidad Nacional del Centro de la Provincia de Buenos Aires. Se incluye en este informe un resumen de cada publicación y se adjunta una copia de las mismas y documentación respaldatoria.

8.1 PUBLICACIONES.

[1] Améndola, Federico; Favre, Liliana

Adapting CRM Systems for Mobile Platforms: An MDA Perspective

International Journal of Computer & Information Science, Vol. 14, No. 1, June 2013

ISSN 15259293, pág. 31-40 (accesible en www.acisinternational.org desde 2014, no informado en el período previo)

Abstract

Mobile phones have become as powerful as any desktop computer in terms of applications they can run. However, the software development does not take advantage of the whole potential of this kind of devices. Enterprises are now adopting mobile technologies for numerous applications. A current problem in software engineering is the modernization of useful legacy systems to mobile platforms. In this context, we describe a reengineering process that integrates traditional reverse engineering techniques such as static and dynamic analysis with Model Driven Development (MDD), MDA (Model Driven Architecture) in particular. We describe a case study that shows how to move CRM (Customer Relationship Management) applications from desktop to mobile platforms. We validated our approach by using the open source application platform Eclipse, EMF (Eclipse Modeling Framework), EMP (Eclipse Modeling Project), the Atlas Transformation Language (ATL) and the Android platform.

Keywords: Mobile Computing, Reverse Engineering, Reengineering, Model Driven Architecture, Customer Relationship Management

[2] Pereira, Claudia; Favre, Liliana; Martinez, Liliana
 "A Rigorous Approach for Metamodel Evolution"
 Capítulo 9 del libro "Advances and Applications in Model-Driven Software Engineering"
 Editores: Garcia-Diaz, V.; Cueva Lovelle, J.M.; Pelayo Garcia-Bustelo,
 B.; Sanjuán Martínez, O. ISBN: 9781466644946 Serie: Advances in Systems
 Analysis, Software Engineering, and High Performance Computing (Asashepc) ISSN
 2327-3453. IGI Global, USA. Agosto de 2014.
<http://www.igiglobal.com/book/advances-applications-model-driven-engineering/75847>

Abstract

Model-Driven Development (MDD) is an initiative proposed by the Object Management Group (OMG) to model centric software development. It is based on the concepts of models, metamodels, and automatic transformations. Models and metamodels are in constant evolution. Metamodel evolution may cause conforming models to become invalid. Therefore, models must be migrated to maintain conformance to their metamodels. Metamodel evolution and model migration are typically performed manually, which is an error-prone task. In light of this, the authors propose a framework for metamodel evolution and model migration that combine an operator-based approach with refactoring and specification matching techniques. They combine classical metamodeling techniques with formal specifications to reason about transformations, metamodels, and their evolution. The authors describe foundations for MDA-based metamodel evolution that allow extending the functionality of the existing CASE tools in order to improve the MDA-based process quality.

Keywords: Model Driven Development; Model Driven Architecture; Metamodeling; Software Evolution; Formal Specification

[3] Martinez, Liliana; Pereira, Claudia; Favre, Liliana
 "Recovering Sequence Diagrams from Object-Oriented Code: An ADM approach".
 Proceedings 9th International Conference on Evaluation of Novel Approaches to Software
 Engineering (ENASE 2014). Pág.188-195. DOI: 10.5220/0004894201880195. SCITEPRESS.
<http://www.scitepress.org> <http://www.enase.org>
<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7077134&queryText=enase%202014&pageNumber=2&newsearch=true>

Abstract

Software modernization is a current research area in the software industry intended to transform an existing software system to a new one satisfying new demands. The initiative Architecture-Driven Modernization (ADM) helps software developers in tackling reverse engineering, software evolution and, software modernization in general. To support modernization problems, the ADM Task Force has defined a set of metamodels such as KDM (Knowledge Discovery Metamodel), being the Eclipse-MDT MoDisco project the official support for software modernization. We propose the application of ADM principles to provide relevant model-based views on legacy systems. We describe a framework to reverse engineering models from object-oriented code. In this context, we show how to recover UML sequence diagrams from Java code. We validate our approach by using ADM standards and MoDisco platform. Our research can be considered a contribution to the MoDisco community; MoDisco does not support reverse engineering of sequence diagrams and, on the

other hand, the MoDisco KDM Discover was used and enriched to obtain the required information for recovering interaction diagrams.

Keywords: Architecture Driven Modernization; Reverse Engineering; UML; Sequence Diagram; Metamodeling; MoDisco

[4] Pereira, Claudia; Martinez, Liliana, Favre, Liliana
“Ingeniería inversa en el contexto de ADM”

2 Congreso Nacional de Ingeniería Informática/Sistemas de Información CONAIISI 2014, ISSN 2346-9927, pág. 900-911.13 y 14 de noviembre, San Luis, Argentina.

<http://conaiisi.unsl.edu.ar/>

Resumen

El proceso de analizar los artefactos de sistemas de software existentes para extraer información y proveer vistas de alto nivel del sistema (ingeniería inversa) es una etapa crucial de la modernización de software. La iniciativa Architecture–Driven Modernization (ADM) fue propuesta por OMG (Object Management Group) con el objetivo de definir estándares para soportar el proceso de modernización en el contexto de desarrollos dirigidos por modelos. Dentro de los estándares se definieron metamodelos relevantes para la ingeniería inversa tales como Knowledge Discovery Metamodel (KDM) y Abstract Syntax Tree Metamodel (ASTM). Se propone en este trabajo aplicar los principios de ADM para proveer vistas de alto nivel de sistemas heredados. Se describe un *framework* para la ingeniería inversa de código orientado a objetos a fin de extraer modelos UML. Se ejemplifica la propuesta con un caso de estudio que muestra cómo recuperar diagramas de casos de uso y de interacción a partir de código Java. La propuesta fue validada utilizando el *framework* de modelado Eclipse y la plataforma MoDisco que puede considerarse como la implementación oficial de los estándares ADM.

Palabras claves: Architecture Driven Modernization; Model Driven Development-MDD; Ingeniería Inversa; Knowledge Discovery Metamodel –KDM; MoDisco; UML

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

Ingeniería inversa en el contexto de ADM. Revista Tecnología y Ciencia. Universidad Tecnológica Nacional. Edición especial CONAIISI 2014. Artículo 11. ISSN 1666 6917. eissn 1666-6933. Vol. 27, noviembre de 2015.

<http://www.utn.edu.ar/secretarias/scyt/introrevista.utn>

Resumen

La ingeniería inversa es una etapa crucial en el proceso de modernización de software. La iniciativa Architecture–Driven Modernization (ADM) surgió con el objetivo de definir estándares para soportar el proceso de modernización en el contexto de desarrollos dirigidos por modelos, siendo relevantes para la ingeniería inversa los metamodelos Knowledge Discovery Metamodel y Abstract Syntax Tree Metamodel. Se propone en este artículo aplicar los principios de ADM para proveer vistas de alto nivel de sistemas heredados. Se describe un *framework* para la ingeniería inversa de código orientado a objetos a fin de extraer modelos UML. Se ejemplifica la propuesta con un caso de estudio que muestra cómo recuperar diagramas de casos de uso y de interacción a partir de código Java. La propuesta fue validada utilizando el *framework* de modelado Eclipse y la plataforma MoDisco que puede considerarse como la implementación oficial de los estándares ADM.

Palabras claves: Architecture Driven Modernization; Model Driven Development-MDD; Ingeniería Inversa; Knowledge Discovery Metamodel–KDM; MoDisco; UML

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

Capítulo 16: "Reverse Engineering of Object-Oriented Code: An ADM Approach" Handbook of Research on Innovations in Systems and Software Engineering," (Editores Garcia-Diaz, V.; Cueva Lovelle, J.M.; Pelayo Garcia-Bustelo) Serie: Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC). IGI GLOBAL, USA. 2015. DOI: 10.4018/978-1-4666-5888-2, ISBN13: 9781466658882, ISBN10: 1466658886, EISBN13: 9781466658899. Vol II, pág. 366-389
<http://www.igi-global.com/book/handbook-research-innovations-systems-software/105945>

Abstract

Software modernization is a new research area in the software industry that is intended to provide support for transforming an existing software system to a new one that satisfies new demands. Software modernization requires technical frameworks for information integration and tool interoperability that allow managing new platform technologies, design techniques and processes. To meet these demands, Architecture-Driven Modernization (ADM) has emerged as the new OMG (Object Management Group) initiative for modernization. Reverse engineering techniques play a crucial role in system modernization. This chapter describes the state of the art in the model-driven modernization area, reverse engineering in particular. A framework to reverse engineering models from object-oriented code that distinguishes three different abstraction levels linked to models, metamodels and formal specification is described. The chapter includes an analysis of technologies that support ADM standards and provides a summary of the principles that can be used to govern current modernization efforts.

Keywords: Architecture-Driven Modernization; Reverse Engineering; Model Driven Architecture; Knowledge Discovery Metamodel; Metamodeling; MoDisco; Model Transformation; Atlas Transformation Language

[7] Améndola, Federico; Fernández, Martín; Favre, Liliana

GLIESE, A framework for experimental game development.

Proc. 12th International Conference on Information Technology : New Generations ITNG 2015. April 13-15, 2015, Las Vegas, Nevada, USA. pp. 528-533. IEEE Press.

<http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7106053>

Abstract

Many computer applications require solving difficult problems (NP or PSPACE), which can be expensive to analyze the space of feasible solutions. In the last 20 years, the theoretical computer science area showed enormous progress in the algorithm design that is not being put in practice. Experimental algorithmic can help to bridge the gap between theory and practice. A domain of applications that involves hard problems is the development of computer games, in particular, games in which the user's opponent is the computer. Heuristic search technique in games has been highly successful; however, there are no tools to assist in testing and improving implementations based on it. In this paper, we describe a framework called GLIESE that focuses on the reuse of software components for experimental game development. Its main goal is to provide software support to simplify the most complex issues

in the implementation of video games enabling to experiment with games in which parts of its logic can be expressed by using different heuristic searches. The simulation and assess of the quality of approximate solution based on different heuristic search techniques is the original aspect that impacts on the game implementation. GLIESE was used in undergraduate courses of “Analysis and Design of Algorithms” in the development of variety of popular games.

Keywords: Game development; education; experimental algorithmic; heuristic search; NP-hard

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

Capítulo 723. “Software Modernization: The State of the Art and the Challenges”

Encyclopedia of Information Science and Technology, 3rd Edition (Editor Medhi Khosrow-Pour). IGI Global, USA. Pág. 7347-7358. DOI: 10.4018/978-1-4666-5888-2.ch723 2014
DOI: 10.4018/978-1-4666-5888-2, ISBN13: 9781466658882, ISBN10: 1466658886, EISBN13: 9781466658899, 2015

Abstract

This chapter analyzes ADM-based software modernization. It provides an overview of the state of the art in reverse engineering techniques and software modernization techniques. Taxonomy of different techniques of reverse engineering is described. We describe how traditional techniques such as static and dynamic analysis can be integrated with ADM standards playing a central role in the evolution of software. Foundations of ADM CASE tools to develop industrial size software are analyzed. Finally, challenges and strategic directions in software modernization are included.

Keywords: Architecture Driven Modernization; Reverse engineering; Modernization; Static Analysis; Dynamic Analysis

[9] Favre, Liliana; Duarte, Daniel

“Formal MOF Metamodeling and Tool Support”

Proc. Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development. MODELSWARD 2016. Pág. 99-110. Roma, Italia, 19 de Febrero de 2016. ISBN: 978-989-758-168-7 SCITEPRESS (Science and Technology Publications). Aceptado en 2015. Publicado en 2016. <http://www.scitepress.org>
<http://www.modelsward.org/?y=2016>

Abstract

Model Driven Development has emerged as a new road to software development industrialization. The most well-known realization of MDD is the Model Driven Architecture (MDA). The essence of MDA is the metamodel MOF (Meta Object Facility) allowing interoperability of different kind of artifacts from multiple technologies. It is important to formalize and reason about MOF metamodels properly. In this paper we propose a rigorous framework for reasoning about “correctness” of metamodels. Our main contribution is related to the integration of MOF metalanguage with formal specification languages based on the algebraic formalism. We define NEREUS, a formal metamodeling language, and processes for reasoning about MOF-like metamodels such as Ecore metamodels. The paper describes a set of tools developed to make formal metamodeling feasible in practice.

Keywords: Model Driven Development; MDA; Metamodeling; MOF; Formal Specification; ANTLR; Test Driven Development (TDD)

[10] Díaz Bilotto, Pablo; Favre, Liliana

"Migrating JAVA to Mobile Platforms through HAXE: An MDD Approach," Capítulo 13 del libro "Modern Software Engineering Methodologies for Mobile and Cloud Environments". Editores: Antonio Miguel Cruz, Sara Paiva ISBN 9781466699168. Serie: Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) (ISSN: 23273453; eISSN: 2327-3461) Pág. 240-268. DOI: 10.4018/978-1-4666-9916-8.ch013. IGI GLOBAL, USA. Aceptado en 2015. Publicado en Enero de 2016
<http://www.igi-global.com/book/modern-software-engineering-methodologies-mobile/140982>

Abstract

The chapter includes a description of an MDA migration process from object-oriented code (Java in particular) to different mobile platforms. The main steps of this process are the specification of an ATL-based metamodel transformation between Java and Haxe and, the implementation of cross-platform, multi-device mobile applications. It includes a use case, the migration of a Java application, "The Set of Mandelbrot". It is simple but it allows us to exemplify in the chapter the different steps of the migration process from Java to Haxe.

Keywords: Model Driven Architecture; Mobile Development; Haxe Language; Atlas Transformation Language; Metamodeling

[11] Favre, Liliana; Martinez, Liliana; Pereira, Claudia. Modernizing software in science and engineering: From C/C++ applications to mobile platforms. ECCOMAS Congress 2016 European Congress on Computational Methods in Applied Sciences and Engineering 5 - 10 June, Crete Island, Greece.15 pág. Aceptado en 2015.

<http://www.eccomas2016.org/>

Abstract.

The evolution of software technology leads to continuous migration of software components and applications. Particularly, most software applications in applied science and engineering are for desktop computers and there is a need to migrate them to mobile technologies. This kind of migration faces many challenges due to the proliferation of different mobile platforms. New programming languages are thus emerging to integrate the native behaviors of the different platforms targeted in development projects. In this direction, the HAXE language allows writing mobile applications that target all major mobile platforms. Novel technical frameworks for information integration and tool interoperability such as the Model Driven Development (MDD) can help to manage a huge diversity of mobile technologies. A specific realization of MDD is the Model Driven Architecture (MDA) proposed by the Object Management Group (OMG). In this work, we propose a migration process from C/C++ software to different mobile platforms that integrates MDA standards with HAXE. C/C++ is one of the most commonly used programming language in science and engineering domains and numerous legacy software components written in C++ require to be modernized. On the one hand, the proposed process follows model-driven principles: all artifacts involved in the process can be viewed as models that conform a particular metamodel, the process itself can be viewed as a sequence of model-to-model transformations and all the extracted information is represented in a standard way through metamodels. On the other hand, HAXE easily adapts

the native behaviors of the different platforms targeted in development projects enabling extremely efficient cross-platform development, ultimately saving time and resources. The proposal was validated in Eclipse Modeling Framework considering that some of its tools and run-time environments are aligned with MDA standards. The paper includes a simple case study, the migration of a C++ application, “the Set of Mandelbrot”, that allow us to exemplify the different steps of the process.

Keywords: Software Modernization, Mobile Development, Model Driven Development, Model Driven Architecture, Migration, HAXE, C++.

8.2 TRABAJOS EN PRENSA Y/O ACEPTADOS PARA SU PUBLICACIÓN.

[12] Favre, Liliana; Martinez, Liliana; Pereira, Claudia
“Model Driven Software Modernization”. Artículo invitado en Encyclopedia of Information Science and Technology, 4 Edition (Editor Medhi Khosrow-Pour). IGI Global, USA. Propuesta aceptada y artículo enviado en 2015.

Abstract

This chapter analyzes ADM-based software modernization. It provides an overview of the state-of-the-art in software modernization techniques. Taxonomy of different techniques is described. A description of how traditional techniques such as static and dynamic analysis can be integrated with ADM standards is presented. We propose a framework for ADM software modernization that integrates different paradigms. Finally, challenges and strategic directions in software modernization are included.

Keywords: Modernization; Architecture Driven Modernization; Reverse Engineering; Refactoring; Static Analysis; Dynamic Analysis; Forward Engineering

8.3 TRABAJOS ENVIADOS Y AÚN NO ACEPTADOS PARA SU PUBLICACIÓN.

Martinez, Liliana; Pereira, Claudia; Favre, Liliana
Migrating C/C++ Software to Mobile Platforms in the ADM Context
International Journal of Interactive Multimedia and Artificial Intelligence. IJIMAI
ISSN 1989-1660. Special Issue on Advances and Applications in the Internet of Things.
Enviado 30/3/2016

8.4 TRABAJOS AÚN NO ENVIADOS PARA SU PUBLICACIÓN.

Se está redactando un libro de autoría sobre modernización de software bajo el enfoque propuesto en esta investigación. Presentará resultados originales para la ingeniería inversa que proponen una integración de técnicas tradicionales de ingeniería inversa, MDA, ADM y metamodelado formal. Preveía concretar su redacción en el año 2015, pero debido a la actualización de su contenido aún no lo he logrado. Estimo poder concluirlo en un plazo no mayor de un año.

También se prevé publicar otros resultados en los próximos meses relacionados a proyectos concluidos:

- la herramienta Papilio y procesos de ingeniería inversa [33]
- la migración de juegos C/C++ a plataformas móviles a través de Haxe [13]

8.5 COMUNICACIONES

8.6 INFORMES Y MEMORIAS TÉCNICAS

9. TRABAJOS DE DESARROLLO DE TECNOLOGÍAS.

9.1 DESARROLLOS TECNOLÓGICOS

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 recursos humanos de calidad pero escasa cantidad para responder a la gran demanda del mercado. El uso de herramientas que, a través de la automatización, potencien una productividad de calidad y volumen puede generar grandes ventajas competitivas.

Se presenta a continuación un detalle de aportes de este proyecto potencialmente transferibles a la industria del software.

- a) Se desarrolló un prototipo de herramienta integrado a Eclipse, para la ingeniería inversa desde código Java a modelos expresados en UML. Este prototipo, denominado *Papilio*, tiene la particularidad de integrar técnicas clásicas de ingeniería inversa con lenguajes para metamodelado y transformaciones de modelos alineados con los estándares de MDA. *Papilio* es un *plugin* para la plataforma Eclipse que toma como punto de partida un proyecto Java y produce como salida diagramas UML de clase, secuencia y casos de uso.
- b) Se definieron metamodelos Ecore para el lenguaje multiplataforma Haxe y para el lenguaje C/C++. Cabe destacar que no existían metamodelos Ecore alineados a MDA de los mencionados lenguajes y que la integración de estos lenguajes a procesos MDD requiere contar con los mismos. La definición de estos metamodelos se considera un aporte a la comunidad MDD que podrá utilizarlos en procesos de modernización, en particular en ingeniería inversa y directa.
- c) Actualmente se trabaja en la definición de un metamodelo C++ alineado a estándares ADM (Architecture Driven Modernization) con el fin de integrarlo a MoDisco.
- d) Se definieron procesos de migración desde Java y C/C++ a plataformas móviles a través de Haxe. Con respecto a Haxe puede remarcarse que surgió ante una necesidad de los desarrolladores de aplicaciones multiplataforma, que prefieren implementar una aplicación una vez, y desplegarla para diversas plataformas con un mínimo esfuerzo. Los procesos de migración definidos permiten aumentar altamente el grado de automatización en migraciones de software C/C++ y Java a diversas plataformas móviles. Se implementaron herramientas que llevan a cabo migraciones de código Java y C/C++ de manera semiautomática al lenguaje Haxe.
- e) A fin de controlar la evolución del software en procesos basados en MDA, se definieron en el marco de este proyecto técnicas de metamodelado formal, particularmente dentro del marco de la evolución de metamodelos MOF se definió un DSL (*Domain Specific*

Language) para metamodelado formal denominado NEREUS basado en el formalismo algebraico. Actualmente se encuentra en ejecución un plan para el desarrollo de herramientas de metamodelado formal que proveerá soporte para la validación de metamodelos MOF a la comunidad MDD. Se concretó el desarrollo de un parser de NEREUS y su integración con herramientas de prueba formal para actividades de testing y verificación.

9.2 PATENTES O EQUIVALENTES.

9.3 PROYECTOS POTENCIALMENTE TRANSFERIBLES, NO CONCLUÍDOS Y QUE ESTÁN EN DESARROLLO.

9.4 OTRAS ACTIVIDADES TECNOLÓGICAS CUYOS RESULTADOS NO SEAN PUBLICABLES

9.5 Sugiera nombres (e informe las direcciones) de las personas de la actividad privada y/o pública que conocen su trabajo y que pueden opinar sobre la relevancia y el impacto económico y/o social de la/s tecnología/s desarrollada/s.

10. SERVICIOS TECNOLÓGICOS.

11. PUBLICACIONES Y DESARROLLOS EN:

11.1 DOCENCIA

En el marco de este proyecto 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 GLIESE 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).

El siguiente trabajo describe al *framework* GLIESE y la experiencia de su uso en el ámbito educativo como soporte para la enseñanza de análisis y diseño de algoritmos:

Améndola, Federico; Fernández, Martín; Favre, Liliana
GLIESE, A framework for experimental game development.

Proc. 12th International Conference on Information Technology : New Generations
ITNG 2015. April 13-15, 2015, Las Vegas, Nevada, USA. pp. 528-533. IEEE Press.

Otro desarrollo que aporta a la docencia es el analizador del lenguaje NEREUS. Si bien NEREUS es un lenguaje de metamodelado formal, un subconjunto del mismo se emplea para la enseñanza de tipos de datos abstractos a partir de especificaciones algebraicas en el curso “Análisis y diseño de algoritmos I”.

El siguiente trabajo resume las características de NEREUS:

Favre, Liliana; Duarte, Daniel

“Formal MOF Metamodeling and Tool Support”

Proc. Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development. MODELSWARD 2016. Pág. 99-110. Roma, Italia, 19 de Febrero de 2016. ISBN: 978-989-758-168-7 SCITEPRESS (Science and Technology Publications).

11.2 DIVULGACIÓN

12. DIRECCIÓN DE BECARIOS Y/O INVESTIGADORES.

12.1. Dirección de proyectos

12.1.1) Proyecto del Programa de Incentivos 03/C249

Título: Tecnologías de Software

Universidad Nacional del Centro de la Pcia. de Buenos Aires.

Directora: Liliana Favre

Co-directora: Mauco, Maria Virginia

Investigadores:

Felice, Laura

Leonardi, María Carmen

Martínez, Liliana

Pereira, Claudia

Colaborador

Améndola, Federico

Acreditado por el período 2013-2014- 2015

Se adjunta a continuación la carátula de la presentación del proyecto

Título del Proyecto: TECNOLOGÍAS DE SOFTWARE

Apellido y Nombre del Director: FAVRE, LILIANA MARÍA

Cargo Docente:
ASOCIADO EXCLUSIVO

Categoría de Investigador: 1

Apellido y Nombre del Codirector: MAUCO, MARÍA VIRGINIA

Cargo Docente:
ADJUNTO EXCLUSIVO

Categoría de Investigador: 3

Lugar de Radicación:
INTIA

Unidad Académica:
Facultad de Ciencias Exactas
Universidad Nacional del Centro de la Provincia de Buenos Aires

Código y Especificación de Disciplina:
1802

Fecha de Inicio del Proyecto : 1/1/2013

Fecha de Finalización : 31/12/2015

Personal Participante:

APELLIDO Y NOMBRE	CAT. INV.	CAT. DOC.	DEDIC.	HS	FUNCIÓN	UNIDAD ACADEMICA	OTRO PROY. (SI-NO)	FIRMA
FAVRE, LILIANA MARÍA	I	Asoc	1	30	Dir	Exa	No	
FELICE, LAURA CELIA	IV	Adj	1	30	Int	Exa	No	
LEONARDI, MARIA CARMEN	III	Adj	1	30	Int	Exa	No	
MARTINEZ, LILIANA INÉS	IV	JTP	1	30	Int	Exa	No	
MAUCO, MARIA VIRGINIA	III	ADJ	1	30	Cod	Exa	No	
PEREIRA,CLAUDIA TERESA	IV	JTP	1	30	Int	Exa	No	

12.1.2) Proyecto del Programa de Incentivos 03/C266

Título: Tecnologías de Software
Universidad Nacional del Centro de la Pcia. de Buenos Aires.
Directora: Favre, Liliana
Co-directora: Leonardi, María Carmen
Investigadores:
Felice, Laura

Mauco, María Virginia
Martínez, Liliana
Pereira, Claudia
Colaborador
Améndola, Federico
Acreditado por el período 2016-2017

Se adjunta a continuación la carátula del proyecto

**Presentación de Proyecto de Investigación para su evaluación y acreditación por la
Secretaría de Ciencia, Arte y Tecnología
Universidad Nacional del Centro de la Provincia de Buenos Aires.**

Título del Proyecto: TECNOLOGÍAS DE SOFTWARE

Apellido y Nombre del Director: FAVRE, LILIANA MARÍA

Cargo Docente:
ASOCIADO EXCLUSIVO

Categoría de Investigador: I

Apellido y Nombre del Codirector: LEONARDI, MARÍA CARMEN

Cargo Docente:
ADJUNTO EXCLUSIVO

Categoría de Investigador: III

Lugar de Radicación:
INTIA

Unidad Académica:
Facultad de Ciencias Exactas
Universidad Nacional del Centro de la Provincia de Buenos Aires

Código y Especificación de Disciplina:
1802

Fecha de Inicio del Proyecto : 1/2016

Fecha de Finalización : 12/2017

Evaluación externa: SI - NO
(adjuntar comprobante con período de acreditación)

Institución:

Personal Participante:

APELLIDO Y NOMBRE	CAT. INV.	CAT. DOC.	DEDIC.	HS	FUNCIÓN	UNIDAD ACADEMICA	OTRO PROY. (SI-NO)	FIRMA
FAVRE, LILIANA MARÍA	I	Asoc	1	30	Dir	Exa	No	
FELICE, LAURA MABEL	IV	Adj	1	30	Int	Exa	No	
LEONARDI, MARIA CARMEN	III	Adj	1	30	Cod	Exa	No	
MARTINEZ, LILIANA INÉS	IV	JTP	1	30	Int	Exa	No	
MAUCO, MARIA VIRGINIA	III	ADJ	1	30	Int	Exa	No	
PEREIRA,CLAUDIA TERESA	IV	JTP	1	30	Int	Exa	No	

12.2. Dirección de profesores graduados

He dirigido en el período 2014-2015 a los siguientes profesores auxiliares:

Liliana Martínez (Jefe de trabajos prácticos- Dedicación exclusiva- Facultad de Ciencias Exactas- UNCPBA)

Claudia Pereira (Jefe de trabajos prácticos- Dedicación exclusiva- Facultad de Ciencias Exactas- UNCPBA)

Federico Améndola (Ayudante de primera, dedicación simple- Facultad de Ciencias Exactas- UNCPBA).

12.3. Dirección de becarios

Carolina Spina. Becaria de entrenamiento CIC. Comisión de Investigaciones Científicas de la Pcia. de Buenos Aires. 2014-2015. RES. 1382 del 1/12/2014 Directora: Liliana Favre

Maximiliano Duthey Becario de entrenamiento CIC. Comisión de Investigaciones Científicas de la Pcia. de Buenos Aires. 2014-2015. RES. 1382 del 1/12/2014. Directora: Liliana Favre

Emiliano Galitiello. Becario del Programa DELTA G- Ministerio de Educación. Secretaría de Políticas Universitarias Marzo 2014-Marzo 2015. Tutora: Liliana Favre. Cumplió el plan propuesto en tiempo y forma.

Daniel Duarte. Becario del Programa DELTA G- Ministerio de Educación. Secretaría de Políticas Universitarias Marzo 2014- Marzo 2015. Tutora: Liliana Favre. Cumplió el plan propuesto en tiempo y forma.

Pablo Nicolás Díaz Bilotto. Becario del Programa DELTA G- Ministerio de Educación. Secretaría de Políticas Universitarias marzo 2014- Marzo 2015. Cumplió el plan propuesto en tiempo y forma.

13. DIRECCIÓN DE TESIS.

Durante el período informado he dirigido las siguientes **tesis de grado** en la carrera de Ingeniería de Sistemas en la Facultad de Ciencias Exactas de la Universidad Nacional del Centro de la Provincia de Buenos Aires:

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

Tesistas: Emiliano Galitiello, Belén Rolandi

Directora: Liliana Favre

Presentado en 2015. Calificación: 10 (diez)

[2] “Desarrollo de aplicaciones móviles a partir de una integración de HAXE en MDD”

Tesista: Pablo Nicolás Díaz Bilotto

Directora: Liliana Favre
Presentado en 2015. Calificación: 10 (diez)

[3] “Desarrollo de herramientas de metamodelado formal”
Tesista: Daniel Duarte
Directora: Liliana Favre
Presentado en 2015. Calificación: 10(diez)

[4] “Migración de software C/C++ a plataformas móviles a partir de MDD (Model Driven Development)”
Tesisistas: Maximiliano Duthey, Carolina Spina
Directora: Liliana Favre
Presentado en 2016. Calificación: 10(diez)
Directora: Liliana Favre

14. PARTICIPACIÓN EN REUNIONES CIENTÍFICAS.

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

16. 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 2014-2015 fue de aproximadamente \$10000. Asimismo he recibido Subsidios Institucionales a Investigadores CIC.

17. OTRAS FUENTES DE FINANCIAMIENTO.

18. DISTINCIONES O PREMIOS OBTENIDOS EN EL PERIODO.

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

20. TAREAS DOCENTES DESARROLLADAS EN EL PERÍODO.

Mi dedicación a la docencia es de aproximadamente 10 horas semanales. Durante el período 2014-2015 desempeñé tareas docentes en el Departamento de Computación y Sistemas de la Facultad de Ciencias Exactas en la Universidad Nacional del Centro de la Provincia de Buenos Aires 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 **Desarrollo de software dirigido por modelos** de la carrera de Ingeniería de Sistemas, Facultad de Ciencias Exactas, Universidad Nacional del Centro de la Provincia de Buenos Aires.

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

Evaluadora de informes de avance en el marco del Programa de Investigación Científica, Desarrollo y Transferencia de Tecnologías e Innovaciones (CyTMA2) de la Universidad Nacional de La Matanza, Departamento de Ingeniería e Investigaciones Tecnológicas. Marzo de 2014.

Evaluadora de trabajos en CONAIISI 2014. 2do. Congreso Nacional de Ingeniería Informática/Sistemas de información. San Luis, Argentina.

Evaluadora de Informes de Becas de las convocatorias 2011, 2012 y 2013 de las Becas de Investigación de la UNLPam. Mayo de 2014.

Evaluadora de trabajos del Workshop de Ingeniería de Software del XX Congreso Argentino de Ciencias de la Computación, CACIC 2014, La Matanza, 20 al 24 de octubre de 2014.

Evaluadora Artículo IJSE (International Journal of Software Engineering). ISSN 1687-6954. Enero 2015.

Evaluadora de dos capítulos del libro “Modern Software Engineering Methodologies for Mobile and Cloud Environments”. Editado por Antonio Miguel Rosado da Cruz (Instituto Politécnico de Viana do Castelo, Portugal), Sara Paiva (Instituto Politécnico de Viana do Castelo, Portugal). Advances in Systems Analysis, Software Engineering, and High Performance Computing (ASASEHPC) Book Series Series Editor: Vijayan Sugumaran (Oakland University, USA). ISSN: 2327-3453. Febrero de 2015.

Jurado docente. Concursos regulares. Universidad Nacional de Mar del Plata. Ordenanza Consejo Académico 1164/14. Área Programación. Materia Programación I. Cargo: Profesor Adjunto Dedicación: simple. 26/3/2015.

Jurado docente. Concursos regulares. Universidad Nacional de Mar del Plata. Ordenanza Consejo Académico 1164/14. Área Programación- Materia Programación I: Cargo: Jefe de Trabajos Prácticos. Dedicación: Simple. 26/3/2015.

Jurado docente. Concursos regulares. Universidad Nacional de Mar del Plata. Ordenanza Consejo Académico 1164/14. Área Programación- Materia Programación I. Cargo: Ayudante graduado. Dedicación Simple. 26/3/2015.

Jurado docente. Concursos regulares. Universidad Nacional de Mar del Plata. Ordenanza Consejo Académico 1164/14. Área Programación- Materia Programación I: Cargo: Ayudante

graduado. Dedicación Parcial. Materia: Taller de Programación I. Cargo: Ayudante graduado. Dedicación: Parcial. 26/3/2015.

Jurado docente. Concursos regulares. Universidad Nacional de Mar del Plata. Ordenanza Consejo Académico 1164/14. Área Programación- Asignatura Taller de Programación I. cargo: Profesor Adjunto. Dedicación exclusiva. 8/5/2015.

Jurado docente. Concursos regulares. Universidad Nacional de Mar del Plata. Ordenanza Consejo Académico 1164/14. Área Programación-Asignatura Programación II. Cargo: Profesor Adjunto. Dedicación Simple. 8/5/2015.

Jurado docente. Concursos regulares. Universidad Nacional de Mar del Plata. Ordenanza Consejo Académico 1164/14. Área Programación-Asignatura Programación II. Cargo: Jefe de Trabajos Prácticos. Dedicación Simple. 8/5/2015.

Jurado docente. Concursos regulares. Universidad Nacional de Mar del Plata. Ordenanza Consejo Académico 1164/14. Área Programación- Asignatura Programación II. Cargo: Ayudante graduado. Dedicación Simple. 8/5/2015.

Jurado docente. Concursos regulares. Universidad Nacional de Mar del Plata. Ordenanza Consejo Académico 1490/15 Área: Programación. Asignatura: Taller de Programación III Departamento de Física. Facultad de Ingeniería. Cargo a concursar: Jefe de trabajos prácticos. Dedicación: Parcial. 4/9/2015.

Jurado docente. Concursos regulares. Universidad Nacional de Mar del Plata. Ordenanza Consejo Académico 1490/15 Área: Programación. Asignatura: Taller de Programación I. Departamento de Física. Facultad de Ingeniería. Cargo a concursar: Jefe de trabajos prácticos Dedicación: Parcial. 4/9/2015.

Jurado docente. Concursos regulares. Universidad Nacional de Mar del Plata. Ordenanza Consejo Académico 1490/15 Área Programación. Asignatura: Programación III. Departamento de Física. Facultad de Ingeniería. Cargo a concursar: Ayudante graduado. Dedicación: Exclusiva. 4/9/2015.

Jurado docente. Concursos regulares. Universidad Nacional de Mar del Plata. Ordenanza Consejo Académico 1490/15 Área: Programación. Asignatura: Programación I. Departamento de Física. Facultad de Ingeniería. Cargo a concursar: Ayudante graduado. Dedicación: Simple. 4/9/2015.

Evaluadora de informes de avance en el marco del Programa de Investigación Científica, Desarrollo y Transferencia de Tecnologías e Innovaciones (CyTMA2) de la Universidad Nacional de La Matanza, Departamento de Ingeniería e Investigaciones Tecnológicas. Mayo de 2015.

Evaluadora de informes de finales en el marco del Programa de Investigación Científica, Desarrollo y Transferencia de Tecnologías e Innovaciones (CyTMA2) de la Universidad Nacional de La Matanza, Departamento de Ingeniería e Investigaciones Tecnológicas . Mayo de 2015.

Evaluadora de presentación de proyectos de investigación de la Universidad Nacional de La Matanza, Departamento de Ingeniería e Investigaciones Tecnológicas. Mayo de 2015.

Evaluadora de trabajos en CONAIISI 2015. 3er. Congreso Nacional de Ingeniería Informática/Sistemas de información. Buenos Aires, Argentina.

Evaluadora de trabajos en CACIC 2015. XXI Congreso Argentino de Ciencias de la Computación CACIC 2015, Junín, Argentina.

Evaluadora de Informes de Avance de proyectos de investigación de la Universidad Nacional de La Matanza, Departamento de Ingeniería e Investigaciones Tecnológicas. 2016.

Integrante del Advisory Board de la revista internacional “Journal of Cases on Information Technology” (JCIT), Information Resources Management Resources Association, ISSN 1048-7717, USA 2004-2016.

Evaluadora de artículo de Journal of Systems and Software, Elsevier (ISSN 0164-1212), 2016.

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

1. Introducción

Gran parte de los sistemas de información vitales en organizaciones de nuestro medio fueron implementados hace varios años con tecnologías que no están alineadas con los actuales objetivos estratégicos de las organizaciones. Estos sistemas, conocidos como *legacy* (heredados), involucran software, hardware, procesos de negocio 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 y sólo el código engloba la historia de su evolución. Además, la entrada en escena de nuevas tecnologías, por ejemplo móviles, motiva la creciente demanda de modernización de sistemas desarrollados más recientemente, por ejemplo, con tecnologías orientadas a objetos [15].

Nuevos enfoques de desarrollo de software enmarcados en lo que se referencia como MDD (*Model Driven Development*) podrían dar respuesta a esta demanda, sin embargo, no están lo suficientemente difundidos en nuestro medio que en general opta por nuevos desarrollos que conllevan un alto riesgo y no son rentables [9] [37].

El objetivo de esta investigación es trabajar en la modernización de software dirigido por modelos bajo la iniciativa impulsada por OMG (*Object Management Group*) denominada ADM (*Architecture Driven Modernization*) [2]. ADM provee estándares para favorecer la interoperabilidad entre herramientas, factor clave para procesos de modernización con modelos. Se propone en esta investigación una integración de ADM con técnicas clásicas de ingeniería inversa y técnicas rigurosas de metamodelado que permita controlar la evolución de software hacia nuevas o actuales tecnologías [36] [17].

ADM lleva a cabo procesos de modernización aplicando los principios y técnicas de desarrollo dirigido por modelos y en particular, los principios esenciales de MDA (*Model Driven Architecture*): la representación de artefactos mediante modelos en diferentes niveles de abstracción que conforman metamodelos y las transformaciones entre modelos [26]. 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*), 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 [26]. 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*) [32]. Otro motor de transformaciones ampliamente usado y alineado a MDA es ATL (*Atlas Transformation Language*) [4].

Los estándares (metamodelos) más relevantes en el contexto de ADM son KDM (*Knowledge Discovery Metamodel*), ASTM (*Abstract Syntax Tree Metamodel*) y SMM (*Software Metrics Metamodel*) [21] [3] [34]. KDM es la base para la modernización de software y representa no sólo al código sino al sistema completo. ASTM es la especificación de los elementos de modelado para expresar árboles de sintaxis abstracta (*Abstract Syntax Tree - AST*). KDM y ASTM son dos especificaciones de modelado complementarias. KDM establece una especificación que permite representar información semántica sobre el sistema de software, mientras que ASTM establece una especificación para representar la sintaxis del código fuente por medio de AST.

El proceso de modernización incluye tres etapas: ingeniería inversa, reestructuración e ingeniería directa. En el contexto de ADM, la meta de la ingeniería inversa es descubrir el conocimiento del sistema existente y producir modelos en diferentes niveles de abstracción que conformen a ASTM y KDM. Estos modelos serán el punto de partida para el proceso de reestructuración y posteriores procesos de ingeniería directa.

MoDisco (*Modeling Discovery*) es un componente GMT (*Eclipse Generative Modeling Technology*) para la ingeniería inversa dirigida por modelos que puede considerarse la implementación oficial de estándares de ADM para la modernización de sistemas [27]. El análisis de la modernización de software basado en ADM es un tema de investigación actual y MoDisco evoluciona en esa dirección con los aportes de la comunidad MDD [8]. Hasta el momento MoDisco provee solamente la ingeniería inversa de diagramas de clase y no soporta la modernización de software desarrollado en C o C++.

2. Objetivos

Teniendo en cuenta lo anteriormente expuesto se propone como objetivo general de este proyecto analizar procesos de modernización dirigidos por modelos basados en la integración del enfoque ADM, con técnicas de ingeniería inversa tradicionales de análisis estático y dinámico y técnicas de verificación. Se continuará con el análisis de la modernización de software desarrollado en C, C++ y Java a fin de adaptarlo a tecnologías móviles y se analizarán otros escenarios de modernización.

3. Metodología

Para lograr los objetivos del proyecto, se propone integrar técnicas semiformales y formales de metamodelado con técnicas clásicas de ingeniería inversa basadas en análisis estático y dinámico.

Las técnicas semiformales de metamodelado se basarán en metamodelos estándares de ADM tales como KDM y ASTM. Las técnicas de verificación formal se basarán en lenguajes formales algebraicos.

La validación de los procesos de modernización 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 MDD, en particular MDA y ADM [14]. El subproyecto EMF (*Eclipse Modeling Framework*) es un framework de modelado que provee infraestructura para el desarrollo de herramientas tales como editores, motores de transformación y soporte para metamodelado. EMF incluye el metamodelo Ecore, el cual es una implementación del estándar MOF. Otro subproyecto Eclipse es MMT (*Model-to-Model Transformation*), originalmente conocido como M2M, que soporta transformaciones entre

modelos. Dentro del ámbito de Eclipse también se han creado motores de transformación como ATL, *QVT Operational* y se avanza en la implementación de *QVT Declarative*, aún en fase de incubación. Además, se utilizará MoDisco para validar la propuesta de modernización. Específicamente, de los posibles escenarios de modernización se analizará la migración de software C, C++ y Java a plataformas móviles a través del lenguaje Haxe [10]. Haxe fue diseñado para que los desarrolladores puedan, por medio de un solo lenguaje y un conjunto de librerías, abarcar varias plataformas, y en particular las utilizadas en plataformas móviles, de manera eficaz.

4. Grado de avance

Siguiendo el plan de actividades propuesto en proyectos previos, se investigaron formalismos y teorías para analizar, administrar y controlar la evolución del software en el contexto de MDA y ADM. Por otra parte, se continuó con la coordinación del desarrollo de prototipos de herramientas para validar los resultados logrados.

Con respecto a la investigación básica, se trabajó en la definición de procesos de modernización en el contexto de ADM y en la evolución de metamodelos en el marco provisto por MOF. Se definió un framework para la ingeniería inversa de código orientado a objetos que integra técnicas tradicionales de ingeniería inversa tales como análisis estático y dinámico con MDA y ADM. El framework provee un conjunto de modelos y elementos para representar sistemas con diferentes niveles de abstracción enmarcándose dentro de la infraestructura propuesta por MDA y los estándares de modernización de ADM. Se propuso un proceso de ingeniería inversa para extraer diagramas UML de alto nivel a partir de código que consta de tres etapas. Una etapa de obtención del modelo en la que se obtiene una representación del programa mediante un árbol de sintaxis abstracta, una etapa de análisis estático, en la que se aplican las técnicas presentadas en [36] sobre el modelo del código para derivar información relevante y, por último, una etapa de transformaciones en la que se realizan diferentes transformaciones sobre el modelo de entrada para obtener los modelos (más) abstractos del sistema, objetivo de este proceso.

A fin de controlar la evolución del software en procesos basados en MDA, se definieron en el marco de este proyecto técnicas de metamodelado formal. Específicamente dentro del marco de la evolución de metamodelos MOF se definió un DSL (*Domain Specific Language*) para metamodelado formal denominado NEREUS basado en el formalismo algebraico [17].

Con respecto a la validación de los resultados se realizaron las siguientes actividades:

a) Se desarrolló un prototipo de herramienta, integrado a Eclipse, para la ingeniería inversa desde código Java a modelos expresados en UML. Este prototipo tiene la particularidad de integrar técnicas clásicas de ingeniería inversa con lenguajes para metamodelado y transformaciones de modelos alineados con los estándares de MDA.

b) Los resultados logrados en el marco de este proyecto para la modernización de software orientado a objetos han sido validados para aplicaciones móviles. Se definió un metamodelo Ecore de Haxe y un proceso para la migración de Java a plataformas móviles [12].

c) Se desarrollaron herramientas de metamodelado formal que proveen soporte para la validación de metamodelos MOF a la comunidad MDD. Se implementó un analizador del lenguaje NEREUS usando ANTLR (*ANother Tool for Language Recognition*) [31]. Teniendo en cuenta que la semántica del lenguaje NEREUS fue dada por traducción al lenguaje CASL (*Common Algebraic Specification Language*), se trabajó en el desarrollo de un traductor de

NEREUS a CASL que permitirá integrarlo con las herramientas provistas por el entorno HETS (*Heterogeneous Tool Set*) [18] [30].

5. Actividades propuestas para el período 2016-2017

Se propone dar continuidad a la investigación desarrollada durante los años 2013-2015 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 integrantes del proyecto y la colaboración de tesistas de grado, pasantes, becarios o auxiliares de docencia participantes en el proyecto.

Se propone trabajar específicamente en la definición de procesos de modernización de software desarrollado en C, C++ y Java. Se trabajará en el contexto de ADM y estándares relacionados como ASTM y KDM. Se extenderán resultados previos vinculados a la ingeniería inversa de modelos UML desde código Java. Asimismo, se definirán procesos para la ingeniería inversa de código C y C++. Estos procesos requieren de los metamodelos de los lenguajes C y C++, que han se han definido en el marco de este proyecto y actualmente están siendo refinados para adaptarlos a los estándares de ADM ASTM y KDM. Se propone definir cómo recuperar modelos C++ (instancias de ASTM) y modelos KDM que permitan a la vez recuperar modelos UML en el marco del *framework* MoDisco.

Se continuará con la definición de procesos de migración a plataformas móviles a través del lenguaje Haxe tanto desde código Java como desde C y C++.

También, en el marco del proyecto se extenderán resultados previos relacionados a definición del DSL NEREUS para metamodelado. Se propone continuar con la integración de estos resultados con CASL. CASL está conectado a diferentes demostradores de teoremas a través de HETS y posibilita la realización de pruebas formales sobre la especificación de dichos metamodelos [30]. La idea central de HETS es proveer un *framework open-source* para integración de métodos formales y pruebas de teoremas. HETS puede conectarse con diversos demostradores automatizados de teoremas (Automatic Theorem Provers – ATP) tales como Isabelle y Spass. Se analizará la evolución de artefactos de desarrollos descritos por metamodelos MOF y se definirán procesos para administrar la evolución de los mismos a partir de este enfoque.

Se propone avanzar en dirección a lograr productos innovadores transferibles. Se han establecido contactos con una empresa tandilense que desarrolla aplicaciones para dispositivos móviles Dokko Group (<http://en.dokkogroup.com.ar/>) con miras al desarrollo de trabajos conjuntos en el área.

6. Bibliografía

- 1) Acceleo. Obeo. Acceleo Generator. <http://www.eclipse.org/acceleo/>, 2015.
- 2) ADM. Architecture-Driven Modernization. <http://adm.omg.org> , 2015.
- 3) ASTM. Abstract Syntax Tree Metamodel, version 1.0, OMG Document Number: formal/2011-01-05 www.omg.org/spec/ASTM 2011.

- 4) ATL. Atlas Transformation Language (ATL) Documentation
www.eclipse.org/ATL/documentation, 2015.
- 5) Barbier, P., Casteran, E., Cariou, E., le Goer, O. Adaptive software based on correct-by construction metamodels, Capítulo 13. Progressions and Innovations in Model Driven Software Engineering, V. García Díaz, J. M. Cueva Lovelle, B. C. Pelayo García-Bustello, O. Sanjuán Martínez, Eds. Hershey, PA: IGI Global, pp. 308-325, 2013.
- 6) Barbier, F., Deltombe, G., Parisy, O., Youbi, K. Model Driven Engineering: Increasing Legacy Technology Independence. Proceedings of Second India Workshop on Reverse Engineering (IWRE, 2011) Thiruvananthapuram, India: CSI ed. pp. 5-10 , 2011
- 7) Bowen, J., Hinze, A. (2011). Supporting mobile application development with model driven emulation. Journal of the ECEASST, Volumen 45 (pp. 1-5).
- 8) Brambilla, M., Cabot, J., Wimmer, M. Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering. Morgan & Claypool Publishers, 2012.
- 9) Bruneliere, H., Cabot, J., Dupe, G. , Madiot, F. MoDisco: a Model Driven Reverse Engineering Framework. Information and Software Technology, 2014.
- 10) Cannase, N.. Haxe. Too Good to be True? GameDuell Tech Talk. <http://www.techtalk-berlin.de/news/read/nicolas-cannasse-introducing-Haxe/> 2015.
- 11) Cánovas Izquierdo, J., García Molina, J. A domain specific language for extracting models in software modernization. Model Driven Architecture - Foundations and Applications, Lecture Notes in Computer Science , Volume 5562, Berlin, Heidelberg: Springer-Verlag, pp. 82-97, 2009.
- 12) Díaz Bilotto, P., and Favre, L. Migrating JAVA to Mobile Platforms through HAXE: An MDD Approach, Capítulo en "Modern Software Engineering Methodologies for Mobile and Cloud Environments". Antonio Miguel Cruz , Sara Paiva (eds. Hershey, PA: IGI GLOBAL, USA , pp.. 240-268, 2016.
- 13) Duthey, M. , Spina, C. "Migración de software C/C++ a plataformas móviles a partir de MDD (Model Driven development)" Trabajo de tesis Ingeniería de Sistemas (L. Favre directora), UNCPBA, 2016
- 14) Eclipse. The Eclipse Modeling Framework. <https://eclipse.org/modeling/emf/> 2016.
- 15) Favre, L. Model Driven Architecture and Reverse Engineering Technologies: Strategic Directions and System Evolution (Premier Source Reference) ISBN 9781615206490. IGI GLOBAL, 2010.
- 16) Favre, L., Martinez, L., Pereira, C. Capítulo 16: Reverse Engineering of Object-Oriented Code: An ADM Approach. Handbook of Research on Innovations in Systems and Software Engineering," (Editores Garcia-Diaz, V.; Cueva Lovelle, J.M.; Pelayo Garcia-Bustelo, B.; Sanjuán Martínez, O.) IGI GLOBAL, USA. 2014. DOI: 10.4018/978-1-4666-5888-2, ISBN13: 9781466658882, ISBN10: 1466658886, EISBN13: 9781466658899. Vol II, pp. 366-389, 2015
- 17) Favre, L., Duarte, D. Formal MOF Metamodeling and Tool Support. Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development.

- MODELSWARD 2016. pp. 99-110. Roma, Italia, 19 de Febrero de 2016. ISBN: 978-989-758-168-7 SCITEPRESS (Science and Technology Publications).
- 18) HETS, Heterogeneous Tool Set.
http://www.informatik.uni.remen.de/agbkb/forschung/formal_methods/CoFI/HETS/, 2016.
 - 19) Islam, N., Want, R. Smartphones: Past, present and future. *Pervasive Computing*, 13(4), 82-92, 2014
 - 20) Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I. ATL: A Model Transformation Tool, *Science of Computer Programming*, vol. 72, n.o 1-2, pp. 31-39, 2008.
 - 21) KDM . Knowledge Discovery Metamodel, version 1.3, OMG Document Number: formal/2011-08-04. www.omg.org/spec/KDM/1.3, 2011.
 - 22) Kramer, D., Clark, T., Oussena, S. (2010). MobDSL: A domain specific language for multiple mobile platform deployment. *Networked Embedded Systems for Enterprise Applications (NESEA)*, 2010 IEEE International Conference (pp. 1-7), Los Alamitos: IEEE Press.
 - 23) Lettner, M., Tschernuth, M., Mayrhofer, R. (2012). Mobile platform architecture review: Android, Iphone, Qt. In *Computer Aided Systems Theory EUROCAST 2011*, R. Moreno-Díaz, F. Pichler, and A. Quesada-Arencibia, *Lecture Notes in Computer Science*, Volumen 6928 (pp. 544-551). Berlin Heidelberg: Springer-Verlag.
 - 24) MMT (2015). Model-to-Model Transformation. Eclipse Modeling Framework. 2016 <https://www.eclipse.org/mmt/>
 - 25) Martinez, L., Pereira, C., Favre, L. Recovering Sequence Diagrams from Object-Oriented Code: An ADM approach'. *Proceedings 9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2014)*. DOI: 10.5220/0004894201880195 SCITEPRESS, pp. 188-195, 2014
 - 26) MDA, Object Management Group Model Driven Architecture (MDA) MDA Guide rev. 2.0, OMG Document ormsc/2014-06-01, 2014
 - 27) MoDisco. Model Discovery. <http://www.eclipse.org/MoDisco>, 2015
 - 28) MOF, OMG Meta Object Facility Core Specification, version 2.4.2, <http://www.omg.org/spec/MOF/2.4.2>, 2014.
 - 29) MOFM2T (2008). MOF Model to Text Transformation Language, Version 1.0. <http://www.omg.org/spec/MOFM2T/1.0/>
 - 30) Mossakowski, T., Maeder, C., Codescu, M. HETS User Guide, version 0.99, http://www.informatik.uni-bremen.de/agbkb/forschung/formal_methods/CoFI/HETS/, 2014.
 - 31) Parr, T. *The Definitive ANTLR 4 Reference* (1st ed.) , Pragmatic Bookshelf, 2013.
 - 32) QVT. QVT: MOF 2.0 Query, View, Transformation, v.1.2, OMG Document: formal/2011-01- 01, www.omg.org/spec/QVT/1.2, 2011.
 - 33) Rolandi, B., Galitiello, E. Ingeniería inversa en el contexto de MDA (Model Driven Architecture). Trabajo de tesis de la carrera de Ingeniería de Sistemas, UNCPBA, 2016.

- 34) SMM. Software Metrics Meta-Model, Version 1.0, OMG Document Number: formal/2012-01-05. <http://www.omg.org/spec/SMM/1.1>, 2012.
 - 35) Sprinkle, J., Rumpe, B., Vangheluwe, H., Karsai, G. Metamodelling: State of the Art and Research Challenges. H. Giese et al. (Eds.), Lecture Notes in Computer Science 6100, Heidelberg, Springer-Verlag, pp. 57-76, 2010.
 - 36) Tonella, P., Potrich, A. Reverse Engineering of Object-Oriented Code. Monographs in Computer Science. Heidelberg: Springer-Verlag, 2005.
 - 37) Ulrich, W., NewComb, P. Information System Transformation. Architecture Driven Modernization Case Studies. MK/OMG Press. Elsevier, 2010.
 - 38) Umuhoza, E., Ed-douibi, H., Brambilla, M., Cabot, J., Bongio, A. Automatic Code Generation for Cross-platform, Multi-Device Mobile Apps: Some Reflections from an Industrial Experience. In 3rd International Workshop on Mobile Development Lifecycle, MobileDeli 2015.
 - 39) UML. OMG Unified Modeling Language™ (OMG UML) Version 2.5, OMG Document Number: formal/2015-03-01. <http://www.omg.org/spec/UML/2.5>, 2015.
 - 40) XMI. XML Metadata Interchange (XMI) Specification. Version 2.5.1. OMG Document Number: formal/2015-06-07. <http://www.omg.org/spec/XMI/2.5.1>, 2015.
-