

# CARRERA DEL INVESTIGADOR CIENTÍFICO Y TECNOLÓGICO

## Informe Científico<sup>1</sup>

PERIODO <sup>2</sup>: 2012-2013

### 1. DATOS PERSONALES

APELLIDO: Favre

NOMBRES: Liliana María

Dirección Particular: Calle: N°:

Localidad: Tandil CP: 7000 Tel:

Dirección electrónica: [liliana.favre@gmail.com](mailto:liliana.favre@gmail.com) / [lfavre@exa.unicen.edu.ar](mailto:lfavre@exa.unicen.edu.ar)

### 2. TEMA DE INVESTIGACION

Tecnologías de software

### 3. DATOS RELATIVOS A INGRESO Y PROMOCIONES EN LA CARRERA

INGRESO: Categoría: Adjunto sin director Fecha: julio de 1998

ACTUAL: Categoría: Adjunto sin director desde fecha: julio de 1998

### 4. INSTITUCION DONDE DESARROLLA LA TAREA

Universidad y/o Centro: 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°: s/n

Localidad: Tandil CP: 7000 Tel: 0249-4439681 int 43

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

<sup>1</sup> Art. 11; Inc. "e"; Ley 9688 (Carrera del Investigador Científico y Tecnológico).

<sup>2</sup> El informe deberá referenciar a años calendarios completos. Ej.: en el año 2014 deberá informar sobre la actividad del período 1°-01-2012 al 31-12-2013, para las presentaciones bianuales.

## 6. EXPOSICION SINTETICA DE LA LABOR DESARROLLADA EN EL PERIODO.

La investigación realizada durante el período 2012-2013 se centró en la definición de procesos rigurosos de evolución de software en desarrollos dirigidos por modelos (MDD- *Model Driven Development*). El objetivo del proyecto es definir procesos rigurosos de evolución de software basados en tres pilares: MDD, técnicas tradicionales de análisis estático y dinámico para ingeniería inversa y, técnicas de verificación basadas en el formalismo algebraico. Se analizó cómo controlar la evolución de sistemas hacia nuevas tecnologías potenciando la productividad a través del desarrollo de herramientas que automaticen actividades esenciales en estos procesos. 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 de tecnologías cohesivas y centradas en modelos. 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. 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*).

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*). ADM lleva a cabo procesos de modernización considerando los principios esenciales de MDA: la representación de artefactos mediante modelos en diferentes niveles de abstracción y las transformaciones entre modelos.

En este contexto, los resultados más relevantes de este período de investigación están vinculados a la definición de procesos de modernización a partir de una integración de técnicas tradicionales de ingeniería inversa con MDA y ADM, la definición de técnicas de metamodelado formal y, la integración de estos resultados con tecnologías móviles.

La validación de resultados se realiza bajo el proyecto de código abierto Eclipse dado que algunos de sus subproyectos proporcionan herramientas y entornos de ejecución alineados con estándares de MDA y ADM. Actualmente se está desarrollando un prototipo de herramienta, integrado a Eclipse, para la ingeniería inversa desde código Java a modelos MDA. También 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. Asimismo se han desarrollado casos de estudio de modernización de software orientado a objetos a fin de adaptarlo a tecnologías móviles.

Los resultados logrados podrían incorporar funcionalidad a las herramientas CASE. Fueron publicados en capítulos de libros, revistas y conferencias internacionales.

## 7. TRABAJOS DE INVESTIGACION REALIZADOS O PUBLICADOS EN ESTE PERIODO.

### 7.1 PUBLICACIONES.

Las publicaciones listadas a continuación muestran resultados originales de investigación logrados en el grupo “Tecnologías de software” que dirijo en la Facultad de Ciencias Exactas de la Universidad Nacional del Centro de la Provincia de Buenos Aires. La publicación [3] es de autoría única por invitación. Las restantes publicaciones muestran resultados de investigación que logré en forma conjunta con otros integrantes del grupo. Se incluye en este informe un resumen de cada publicación y se adjunta una copia de las mismas y documentación respaldatoria.

[1] 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)  
pág. 3-23, ISSN 2090-1801 (versión online). Vol.5, n° 2, julio 2012.

<http://www.ijse.org.eg/>

#### 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).

[2] 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.

<http://www.igi-global.com/book/emerging-technologies-evolution-maintenance-software/55286>

### **Abstract**

System modernization requires the existence of technical frameworks for information integration and tool interoperation 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 use reverse engineering 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.

[3] 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 978-953-51-01581. pág. 55-82. Intech Open Access Publisher. 2012.  
<http://www.intechopen.com/books/reverse-engineering-recent-advances-and-applications>

### **Abstract**

The progress in the last decade in scalability and incremental verification of basic formal methods could be used as a complement of static and dynamic analysis with tests, assertions and partial formal specification. In this light, this chapter describes MDA reverse engineering of object-oriented code that is based on the integration of traditional compiler techniques such as static and dynamic analysis, metamodeling techniques based on MDA standards and, partial formal specification. We propose to exploit the source code as the most reliable description of both, the behavior of the software and the organization and its business rules. Different principles of reverse engineering are covered, with special emphasis on consistency, testing and verification. We propose a formal metamodeling technique to control the evolution of metamodels that are the essence to achieve interoperability between different software artifacts involved in reverse engineering processes. Rather than requiring that users of transformation tools manipulate formal specification, we want to provide formal semantic to graphical metamodeling notations and develop rigorous tools that permit users to directly manipulate metamodels they have created. As an example, we analyze the reverse engineering of Java code however the bases of our approach can be easily applied to other object-oriented languages.

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

“Architecture Driven Modernization for Software Reverse Engineering Technologies”

Capítulo 12 del libro “Progressions and Innovations in Model Driven Software Engineering”

(Editores Garcia-Diaz, V.; Cueva Lovelle, J.M.; Pelayo Garcia-Bustelo, B.; Sanjuán Martínez, O.) pág. 288-307

Serie: Advances in Systems Analysis, Software Engineering, and High Performance Computing. IGI Global, USA. Junio de 2013.

DOI: 10.4018/978-1-4666-4217-1, ISBN13: 9781466642171, ISBN10: 1466642173, EISBN13: 9781466642188

<http://www.igi-global.com/book/progressions-innovations-model-driven-software/74194>

### **Abstract**

Modernization of legacy systems 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. The new OMG (Object Management Group) initiative for modernization aligned with this requirement is Architecture-Driven Modernization (ADM). Reverse engineering techniques play a crucial role in system modernization. In this chapter, the authors describe the state-of-the-art in the model-driven modernization area, reverse engineering in particular, and discuss about existing tools and future trends. In addition, they describe a framework to reverse engineering models from object-oriented code that distinguishes three different abstraction levels linked to models, metamodels, and formal specifications. As an example, this chapter shows how to reverse engineering use case diagrams from Java code in the ADM context focusing on transformations at metamodel level. The authors validate their approach by using Eclipse Modeling Framework.

[5] 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.) pág.177-200

Serie: Advances in Systems Analysis, Software Engineering, and High Performance Computing (Asashepc). IGI Global, USA. Agosto de 2013.

DOI: 10.4018/978-1-4666-4494-6, ISBN13: 9781466644946, ISBN10: 146664494X, EISBN13: 9781466644953

<http://www.igi-global.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.

[6] Améndola, Federico; Favre, Liliana

“Science & Engineering Software Migration: Moving from desktop to mobile applications”  
V International Conference on Computational Methods for Coupled Problems in Science and Engineering. COUPLED PROBLEMS 2013. S. Idelsohn, M. Papadrakakis and B. Schrefler (Eds). Abstract- pág. 1418-1430, 17-19 junio, Ibiza, España. <http://congress.cimne.com/coupled2013>

### **Abstract**

The proliferation of mobile devices over the last years provides opportunities and challenges for solving problems in Science & Engineering. Among other novel features, mobile devices contain global positioning sensors, wireless connectivity, built-in web browsers and photo/video/voice capabilities that allow providing highly localized, context aware applications. Mobile phones have become as powerful as any desktop computer in terms of applications they can run. However, the software development in mobile computing is still not as mature as it is for desktop computer and the whole potential of mobile devices is wasted. A current problem in the engineering community is the adaptation of desktop applications for mobile technologies.

To take advantage of new platform technologies, existing software must evolve. A number of solutions have been proposed to deal with this problem such as redevelopment, which rewrites existing applications, or migration, which moves the existing system to a more flexible environment while retaining the original system data and functionality. A good solution should be to restore the value of the existing software, extracting knowledge and exploiting investment in order to migrate to new software that incorporates the new technologies.

On the one hand, traditional reverse engineering techniques can help in the software migration to mobile applications. They are related to the process of analyzing available software with the objective of extracting information and providing high-level views on the underlying code. On the other hand, to achieve interoperability with multiple platforms the migration needs of technical frameworks for information integration and tool interoperability such as the initiative of the Object Management Group (OMG) called Model Driven Architecture (MDA). The outstanding ideas behind MDA are separating the specification of the system functionality from its implementation on specific platforms and managing the software evolution from abstract models to implementations increasing the degree of automation.

The objective of this paper is to describe a reengineering process that allow moving existing desktop applications for solving engineering problems of multidisciplinary character to mobile platforms. Our research aims to simplify the creation of applications for mobile platforms by integrating traditional reverse engineering techniques, such static and dynamic analysis, with MDA. We validated our approach by using the open source application platform Eclipse, EMF (Eclipse Modeling Framework), EMP (Eclipse Modeling Project) and the Android platform.

[7] Améndola, Federico; Favre, Liliana

"Adapting CRM Systems for Mobile Platforms: An MDA Perspective"

Proceedings 14th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2013) July 1 - 3, 2013 Honolulu, Hawaii, USA. IEEE Press, pág. 323-328. ISBN 9780769550053.

<http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6598367>

**Abstract**

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.

[8] 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). Proceedings 9th International Conference on Evaluation of Novel Approaches to Software Engineering, (ENASE 2014) pág. 188-195 DOI: 10.5220/0004894201880195. Copyrightc SCITEPRESS.

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

**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.

**7.2 TRABAJOS EN PRENSA Y/O ACEPTADOS PARA SU PUBLICACIÓN.**

9] Favre, Liliana; Martinez, Liliana; Pereira, Claudia.

"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. Artículo invitado en 2013. Fecha prevista de la publicación 2014.  
DOI: 10.4018/978-1-4666-6359-6, ISBN13: 9781466663596, ISBN10: 1466663596,  
EISBN13: 9781466663602

### **Abstract**

Modernization of legacy systems 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. For many years, traditional reengineering has been a solution to software modernization. However, reengineering often fails due to it involves ad hoc and non-standardized processes. 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. This chapter focuses on modernization of object-oriented applications in order to facilitate the adaptation of legacy applications to new technologies.

[10] Favre, Liliana; Martinez, Liliana; Pereira, Claudia  
“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. Aceptado en 2013. Fecha prevista de publicación  
2014. 12 pág. DOI: 10.4018/978-1-4666-5888-2, ISBN13: 9781466658882, ISBN10:  
1466658886, EISBN13: 978146665889

### **Abstract**

Software modernization, understood as technological and functional evolution of legacy systems, provides principles, methods, techniques and tools to support the transformation from an existing software system to a new one that satisfies new requirements.

Modernization is related to different processes such as migration, software refactoring, architecture restructuring, and mainly reverse engineering. Reverse engineering techniques allow supporting an integral part of the software modernization, specifically, the process of analyzing available software artifacts such as requirements, design, architectures, code or byte code, with the objective of extracting information and providing high-level views on the underlying system. Thus, software modernization starts from an existing implementation and requires an evaluation of every part of the system that could be transformed or implemented anew from scratch.

Reverse engineering involves (re)discovering the functional, structural and behavioral semantics of a given artifact in order to document, maintain, improve or migrate them. Software modernization requires new technical frameworks for information integration and tool interoperability such as the Model Driven Development (MDD). MDD refers to a range of development approaches based on the use of software models as first class entities. The most well-known realization of MDD is the OMG standard Model Driven Architecture (MDA) (MDA, 2013). 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 increasing the degree of automation of model transformations and achieving interoperability with multiple platforms, programming languages and formal languages. The essence of MDA is the Meta Object Facility Metamodel (MOF) that allows different kinds of software artifacts to be used together in a single project. Models play a major role in MDA which distinguishes Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM). Some authors also distinguish Implementation Specific Model (ISM) as a description (specification) of the system in source code. To express transformations, OMG defined the MOF 2.0 Query, View, Transformation (QVT) metamodel.

Software modernization can be summarized as follows. First, information is extracted out of the system artifacts. Second, this information is analyzed in order to take adequate modernization decisions and finally, the information must be transformed to new artifacts. These steps are supported by metamodels to describe existing systems, discoverers to automatically create models of these systems and, tools to understand and transform complex models created out of existing systems. OMG is involved in the definition of standards to successfully modernize existing information systems. In this direction, OMG Architecture-Driven Modernization Task Force (ADM TF) is developing a set of specifications and promoting industry consensus on modernization of existing applications.

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 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.

### **7.3 TRABAJOS ENVIADOS Y AUN NO ACEPTADOS PARA SU PUBLICACION.**

### **7.4 TRABAJOS TERMINADOS Y AUN NO ENVIADOS PARA SU PUBLICACION.**

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. Se prevé concretar su redacción a mediados del año 2015.

### **7.5 COMUNICACIONES.**

### **7.6 INFORMES Y MEMORIAS TECNICAS.**

## **8. TRABAJOS DE DESARROLLO DE TECNOLOGÍAS.**

### **8.1 DESARROLLOS TECNOLÓGICOS.**

### **8.2 PATENTES O EQUIVALENTES.**

### **8.3 PROYECTOS POTENCIALMENTE TRANSFERIBLES, NO CONCLUIDOS Y QUE ESTAN EN DESARROLLO.**

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

a) Se está desarrollando 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. Se prevé una presentación formal del prototipo de herramienta en el transcurso del corriente año.

b) Los resultados logrados en el marco de este proyecto para la modernización de software orientado a objetos han sido validados para aplicaciones móviles. Actualmente se trabaja en la integración del lenguaje multiplataforma HAXE a este enfoque. 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. Específicamente, en el marco del proyecto se proveerá de un metamodelo HAXE alineado con los estándares de MDA. Este metamodelo permitirá especificar transformaciones a nivel de metamodelos y soportar tanto procesos de generación de código como de ingeniería inversa, por ejemplo para la migración de componentes orientadas a objetos a tecnologías móviles. Se prevé concretar el desarrollo del metamodelo HAXE en el transcurso de este año.

Con respecto a tecnologías móviles, también puede remarcarse que se definió, dentro del proyecto, un plan para migrar al framework GLIESE para desarrollo de video juegos en ámbitos educativos al lenguaje HAXE. GLIESE fue implementado en el marco de este proyecto y permite la generación de código en C++ . La migración a Haxe permitiría integrarlo con múltiples plataformas, particularmente móviles. Se han establecido contactos con una empresa tandilense que desarrolla video juegos para dispositivos móviles Dokko Group (<http://en.dokkgroup.com.ar/>).

c) 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. En el transcurso de este año se concretará el parser de NEREUS, implementado usando ANTLR (*ANother Tool for Language Recognition*). Teniendo en cuenta que la semántica del lenguaje NEREUS fue dada por traducción al lenguaje CASL (*Common Algebraic Specification Language*), se desarrolla un

traductor de NEREUS a CASL que permitirá integrarlo con las herramientas provistas por COFI (*The Common Framework Initiative for Algebraic Specification and Development*).

#### **8.4 OTRAS ACTIVIDADES TECNOLÓGICAS CUYOS RESULTADOS NO SEAN PUBLICABLES**

### **9. SERVICIOS TECNOLÓGICOS**

### **10. PUBLICACIONES Y DESARROLLOS EN:**

#### **10.1 DOCENCIA**

1) 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).

En este contexto, se está terminando la redacción de un trabajo para enviar a una revista que describe al *framework* GLIESE para desarrollo de juegos en el ámbito educativo y su utilización como soporte para la enseñanza de análisis y diseño de algoritmos. Se prevé enviarlo para su evaluación en los próximos meses.

Asimismo se prevé dar continuidad a esta investigación extendiendo a GLIESE a fin de que provea un soporte para la simulación de soluciones basadas en técnicas de búsqueda heurística en juegos en los que el oponente del usuario es la computadora. Se considera que los resultados de esta investigación pueden aportar a la enseñanza de la programación, tanto al permitir trabajar a alumnos desde niveles tempranos de su formación con aplicaciones producto, como así también para enseñar a construir soluciones algorítmicas basadas en búsquedas heurísticas, lo cual aún es un desafío.

2) Se dirigió el siguiente trabajo de alumnos de la carrera de Ingeniería de Sistemas, presentado como trabajo de cátedra en las 41 JAIIO

“Una implementación de la variante TSPPDL del Problema del Viajante”

Autores: David López, Javier Marsicano.

Directora: Liliana Favre

Publicado como artículo en el Simposio EST 2012 (15 Concurso de Trabajos Estudiantiles)  
41 JAIIO. Trabajo Finalista, 2012.

## **10.2 DIVULGACIÓN**

### **11. DIRECCION DE BECARIOS Y/O INVESTIGADORES.**

#### **11.1. Dirección de proyectos**

##### **11.1.1) Proyecto del Programa de Incentivos 03/C217.**

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:* Federico Améndola (2012)

Período 2010- 2012

##### **11.1.2) 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 carátula del proyecto de incentivos acreditado y vigente hasta el 2015

**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	

### **Resumen del proyecto 03/C249**

*El objetivo general del proyecto “Tecnologías de software” es integrar técnicas orientadas al cliente, semiformales y formales en procesos de evolución de software basados en MDD (Model Driven Development). Estas técnicas abarcan diferentes procesos dentro del desarrollo, como por ejemplo técnicas para la elicitación y modelado de requisitos así como también técnicas de evolución de software En este contexto, se proponen dos líneas de investigación:*

#### **Línea 1: Evolución de software en MDD**

*Directora: Liliana Favre*

*El objetivo de la Línea 1 es definir procesos rigurosos de evolución de software basados en tres pilares: el enfoque MDA, técnicas tradicionales de análisis estático y dinámico para ingeniería inversa y técnicas de verificación basadas en el formalismo algebraico. Se pretende analizar cómo controlar la evolución de sistemas hacia nuevas tecnologías potenciando la productividad a través del desarrollo de herramientas que automaticen actividades esenciales en estos procesos.*

#### **Línea 2: Ingeniería de requisitos orientada a objetivos en MDD**

*Directora: Carmen Leonardi*

*El objetivo principal de la Línea 2 es la integración de estrategias de requisitos con técnicas y modelos de objetivos para poder comenzar el proceso de adquisición y modelado de requisitos desde una perspectiva del negocio, más general, en el contexto de procesos de desarrollo basados en MDD.*

### **11.2. Dirección de auxiliares graduados**

He dirigido en tareas de investigación en el período 2012-2014 a los siguientes auxiliares graduados:

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

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

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

### **11.3. Dirección de becarios**

Federico Améndola- **Becario de estudio CIC- 2012-2013**  
Tema: “Desarrollo de software dirigido por modelos y aplicaciones móviles”  
Directora: Liliana Favre

#### **11.4 Becarios alumnos**

Actualmente soy tutora de 3 becarios del programa DELTA G (“Proyecto de estímulo a la graduación de estudiantes de carreras de ingeniería”) otorgada por la Secretaría de Políticas Universitarias (SPU) quienes se encuentran desarrollando su tesis de grado bajo mi dirección en la Carrera de Ingeniería de Sistemas de la UNCPBA.

Daniel Duarte (Becario Delta G 2014)  
Tutora: Liliana Favre

Emiliano Galitiello (Becario Delta G 2014)  
Tutora: Liliana Favre

Pablo Nicolás Díaz Bilotto (Becario Delta G 2014)  
Tutora: Liliana Favre

##### **11.4.1. Solicitud de becas de investigación para alumnos (aún sin evaluación)**

Solicitud de beca de entrenamiento CIC  
Tema: Desarrollo de software dirigido por modelos y aplicaciones móviles  
Postulante: Carolina Spina  
Dirección propuesta: Liliana Favre

Solicitud de beca de entrenamiento CIC  
Tema: Modernización de software  
Postulante: Maximiliano Duthey  
Dirección propuesta: Liliana Favre

Solicitud de beca CIN (Becas de estímulo a las vocaciones científicas)  
Tema: Juegos serios aplicados al análisis y diseño experimental de algoritmos  
Postulante: Joaquín Bengoechea  
Dirección propuesta: Liliana Favre

## **12. DIRECCION DE TESIS.**

### **12.1 Dirección de tesis de grado**

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 en 2012.  
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. En desarrollo.

Directora: Liliana Favre

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

Pablo Nicolás Díaz Bilotto

Trabajo de tesis de la carrera de Ingeniería de Sistemas. Facultad de Ciencias Exactas. Universidad Nacional del Centro de la Pcia. de Buenos Aires. En desarrollo.

Directora: Liliana Favre

“Desarrollo de herramientas de metamodelado formal”

Daniel Duarte

Trabajo de tesis de la carrera de Ingeniería de Sistemas. Facultad de Ciencias Exactas. Universidad Nacional del Centro de la Pcia. de Buenos Aires. En desarrollo.

Directora: Liliana Favre

### **13. PARTICIPACION EN REUNIONES CIENTIFICAS.**

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

### **15. SUBSIDIOS RECIBIDOS EN EL PERIODO.**

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 2012-2013 fue de aproximadamente \$6000.

### **16. OTRAS FUENTES DE FINANCIAMIENTO**

### **17. DISTINCIONES O PREMIOS OBTENIDOS EN EL PERIODO.**

### **18. ACTUACION EN ORGANISMOS DE PLANEAMIENTO, PROMOCION O EJECUCION CIENTIFICA 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.

### **19. TAREAS DOCENTES DESARROLLADAS EN EL PERIODO.**

Durante los años 2012 y 2013 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

## **20. OTROS ELEMENTOS DE JUICIO NO CONTEMPLADOS EN LOS TITULOS ANTERIORES.**

Integrante de la Comisión Evaluadora de Informes de becas de doctorado y maestría convocatoria 2009 de la Universidad Nacional de La Pampa. Res. 117/12. mayo de 2012.

Integrante de la Comisión Evaluadora de Informes de becas de doctorado y maestría convocatoria 2010 de la Universidad Nacional de La Pampa. Res. 116/12. Mayo de 2012.

Evaluadora de trabajos del Workshop de Ingeniería de Software del XVIII Congreso Argentino de Ciencias de la Computación, CACIC Bahía Blanca. 8 al 12 de octubre de 2012.

Jurado docente concurso de Profesores Asignatura "Sistemas y organizaciones" correspondiente a la Especialidad de Sistemas en Universidad tecnológica Regional Buenos Aires, setiembre de 2012.

Evaluadora de 4 proyectos de investigación de la Secretaría de Ciencia y Técnica y Estudios Regionales de la Universidad Nacional de Jujuy en la disciplina Ingeniería. RES. SecTER 143/12.

Evaluadora Docente Titular Res. Consejo Directivo 2356/2012 en la Carrera Académica de profesores en las asignaturas "Arquitectura de las computadoras", "Sintaxis y semántica del lenguaje" y "Teoría de control". 1 de noviembre de 2012.

Evaluadora Docente Titular Res. Consejo Directivo 2355/2012 en la Carrera Académica de profesores en las asignaturas "Comunicaciones", "Matemática discreta", "Matemática Superior" "Paradigmas de programación" y "Simulación". 2 de noviembre de 2012.

Integrante de la Comisión Evaluadora de Informes de presentaciones de becas de doctorado y maestría convocatoria 2013 de la Universidad Nacional de La Pampa. Res. 373/12. Noviembre de 2012.

Integrante de la Comisión Evaluadora de Informes de becas de doctorado y maestría convocatoria 2010 de la Universidad Nacional de La Pampa. Res. 105/13 Mayo de 2013.

Integrante de la Comisión Evaluadora de Informes de becas de doctorado y maestría convocatoria 2011 de la Universidad Nacional de La Pampa. Res. 107/13. Mayo de 2013.

Integrante de la Comisión Evaluadora de Informes de becas de doctorado y maestría convocatoria 2012 de la Universidad Nacional de La Pampa. Res.106/13. Mayo de 2013.

Evaluadora en el "Programa Becas Estímulo Vocaciones Científicas-2013" en la Ciudad de Santa Rosa (La Pampa) los días 27 y 28 de junio de 2013. Resolución CIN C.E. 846/13.

Evaluadora artículos Encyclopedia of Information Science and Technology, 3rd Edition (Editor Medhi Khosrow-Pour). IGI Global, USA, 2013.

Evaluadora Artículo IJSE (International Journal Of Software Engineering) ISSN 1687-6954. Enero 2013.

Integrante del Advisory Board de la revista internacional "Journal of Cases on Information Technology" (JCIT), Information Resources Management Resources Association, ISSN 1048-7717, USA, 2013

Evaluadora Titular de los Proyectos "Proyectos Orientados a la Investigación Regional" POIRe 2013, aprobados por Resolución del CS N° 419/12 y Resolución del CS N° 158/13.

Evaluadora 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 . Agosto de 2013.

Evaluadora de trabajos del Workshop de Ingeniería de Software del XIX Congreso Argentino de Ciencias de la Computación, CACIC 2013, Mar del Plata. 21 al 25 de octubre de 2013.

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.

## **21. TITULO Y PLAN DE TRABAJO A REALIZAR EN EL PROXIMO PERIODO.**

### **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* (legados), 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 englobando sólo el código 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. Nuevos enfoques de desarrollo de software dirigidos por modelos 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 no son rentables y conllevan un alto riesgo. El objetivo de esta investigación es definir rigurosas técnicas de modernización de software basadas en la realización de MDD propuesta por OMG (*Object Management Group*) denominada MDA (*Model Driven Architecture*) y ADM (*Architecture Driven Modernization*). Se propone una integración de MDA y ADM con técnicas clásicas de ingeniería inversa y métodos formales que permita controlar la evolución de sistemas *legacy* hacia nuevas o actuales tecnologías. La idea es potenciar la productividad mediante la definición de las bases para el desarrollo de herramientas que aumenten el grado de automatización en procesos de modernización de software.

**Palabras claves:** ADM; MDA; Ingeniería inversa; Reingeniería de software; Modernización de software; Metamodelado; Especificación formal

## 1. Introducción

Actualmente existe una creciente demanda de soporte para la modernización de sistemas que deben adaptarse a nuevas demandas [4]. En el contexto de MDA, OMG está involucrado en el desarrollo de estándares para la modernización de sistemas a través de la iniciativa ADM (*Architecture Driven Modernization*) [17,1]. ADM lleva a cabo procesos de modernización considerando los principios esenciales de MDA: la representación de artefactos mediante modelos en diferentes niveles de abstracción y las transformaciones entre modelos. Uno de los estándares implementados en el contexto de ADM es el metamodelo KDM (*Knowledge Discovery Metamodel*) [16] que garantiza interoperabilidad e intercambio de datos entre diferentes herramientas de modernización. Otros estándares son ASTM (*Abstract Syntax Tree Metamodel*) y SMM (*Software Metrics Metamodel*) [1, 4]

Las CASE MDA comerciales proveen un soporte incipiente para la modernización de sistemas y en particular para los procesos involucrados como los de ingeniería inversa [7,6]. Sus principales limitaciones con respecto a este enfoque están vinculadas a la acotada especificación de plataformas y puentes entre plataformas y a la falta de alineación a estándares de ADM. Si bien 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 co-evolución de metamodelos, es uno de los grandes desafíos en MDD [8, 15, 23, 25] y en particular ADM.

MoDisco (Modeling Discovery) [5, 18] es un componente GMT (Eclipse Generative Modeling Technology) [11] para la ingeniería inversa dirigida por modelos que puede considerarse la implementación oficial de estándares de ADM para la modernización de sistemas. 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. Es un framework genérico que permite diferentes formas de extensión, por ejemplo complementando la capa de tecnologías para soportar otros

lenguajes o plataformas y brindando facilidades para desarrollar los metamodelos correspondientes en EMF Ecore [11], descubridores (discovers) de modelos y transformaciones modelo-a-modelo en ATL [2] entre otras facilidades.

## 2. Objetivos

Teniendo en cuenta lo anteriormente expuesto se propone como objetivo general de este trabajo analizar procesos de evolución de software basados en la integración del enfoque ADM y técnicas relacionadas (como metamodelado MOF, KDM y ASTM). Se propone integrar ADM con técnicas de ingeniería inversa tradicionales de análisis estático y dinámico y técnicas de verificación basadas en el formalismo algebraico. Se pretende que los resultados obtenidos permitan extender la funcionalidad de las CASE existentes.

## 3. 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 los estándares de metamodelado de ADM. Las técnicas formales se basarán en lenguajes formales algebraicos, en particular el lenguaje NEREUS y las herramientas provistas por COFI (*The Common Framework Initiative for Algebraic Specification and Development*) [12, 9]. 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 como de directa [13,24].

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 MMT (*Model-to-Model Transformation*), originalmente conocido como M2M, que soporta transformaciones entre modelos [11]. 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 [22].

Específicamente, la validación de los procesos de modernización se realizará sobre el framework MoDisco.

Se propone desarrollar casos de uso de modernización de aplicaciones de escritorio en C++ y Java a fin de adaptarlas a tecnologías móviles usando como lenguaje multiplataforma HAXE [14].

## 4. Actividades propuestas para el período 2014-2015

El plan de investigación propone dar continuidad a la investigación desarrollada durante los años 2011-2012 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 en el marco provisto por MOF y en la definición de procesos de modernización en el contexto de ADM.

Se analizará la evolución de artefactos de desarrollos descritos por metamodelos MOF y se definirán procesos para administrar la evolución de los mismos [13, 19]. Este análisis se basará en resultados previos de este proyecto, en particular en la definición de un DSL (*Domain Specific Language*) formal para metamodelado. Se propone definir un framework basado en MDA que combine técnicas formales de *matching* de metamodelos y aproximaciones basadas en operadores de reusabilidad.

Asimismo se trabajará en la definición de procesos de modernización de sistemas en el contexto de ADM y estándares relacionados como ASTM y KDM. Se adaptarán resultados previos vinculados a la ingeniería inversa de modelos UML desde código Java. Asimismo se adaptarán técnicas fundamentales en procesos de evolución de software como *refactoring*, patrones de diseño y *traceability* al contexto de ADM.

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

a) Se está desarrollando 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. Durante el corriente año se prevé una presentación formal del prototipo de herramienta para la ingeniería inversa de código Java a fin de extraer modelos UML, en particular la ingeniería inversa de diagramas de clase, de secuencia y casos de uso. Se prevé dar continuidad al desarrollo, tanto para que soporte otros tipos de diagramas como de plataformas.

b) Los resultados logrados en el marco de este proyecto para la modernización de software orientado a objetos han sido validados para aplicaciones móviles [10, 27]. Actualmente se trabaja en la integración del lenguaje multiplataforma HAXE a este enfoque [14]. 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. Específicamente, en el marco del proyecto se proveerá de un metamodelo HAXE alineado con los estándares de MDA. Este metamodelo permitirá especificar transformaciones a nivel de metamodelos y soportar tanto procesos de generación de código como de ingeniería inversa, por ejemplo para la migración de componentes orientadas a objetos a tecnologías móviles. Se prevé concretar el desarrollo del metamodelo HAXE en el transcurso de este año.

Con respecto a tecnologías móviles, y relación a lo expuesto previamente, también puede remarcarse que se definió, dentro del proyecto, un plan para migrar al framework GLIESE para desarrollo de video juegos en ámbitos educativos al lenguaje HAXE. GLIESE permite la generación de código en C++ y HAXE permitiría integrarlo con múltiples plataformas, particularmente móviles.

c) 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 para metamodelado formal denominado NEREUS basado en el formalismo algebraico [12]. 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. En el transcurso de este año se concretará el parser de NEREUS, implementado usando ANTLR (*ANother Tool for Language Recognition*) [21]. Teniendo en cuenta que la semántica del lenguaje NEREUS fue dada por traducción al lenguaje CASL (*Common Algebraic Specification Language*), se desarrolla un traductor de NEREUS a CASL que permitirá integrarlo con las herramientas provistas por COFI [3, 9]. Se prevé concretarlo en este período

Actualmente he organizado cursos optativos en la carrera de Ingeniería de Sistemas de la UNCPBA vinculados al proyecto y se prevé continuar con el dictado de otros cursos que aporten a la capacitación de recursos humanos en modernización de software y motiven a alumnos de la carrera a participar en este proyecto.

#### 4. Referencias y bibliografía principal

1. ADM (2014). Architecture-Driven Modernization Task Force. [www.omg.org/mda](http://www.omg.org/mda)
2. ATL (2014). ATL Documentation. [www.eclipse.org/m2m/atl/documentation](http://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. Brambilla, M., Cabot, J., Wimmer, M. (2012). Model Driven Engineering in Practice. Morgan & Claypool Publishers.
5. Bruneliere, H., Cabot, J., Dupé, G., Madiot, F. (2014) MoDisco: a Model Driven Reverse Engineering Framework. Information and Software Technology, Elsevier, Volume 56, Issue 8, pág. 1012–1032.
6. Canfora, G., Di Penta, M. (2007). New Frontiers of Reverse Engineering. Future of Software Engineering. Proceedings of Future of Software Engineering (FOSE 2007) pág. 326-341. Los Alamitos: IEEE Press.
7. CASE MDA (2014). [www.omg.org/mda/committed-products.htm](http://www.omg.org/mda/committed-products.htm)
8. 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), pág. 222-231, IEEE Press.
9. COFI (2014). The Common Framework Initiative for algebraic specification and development [http://www.informatik.uni-bremen.de/cofi/wiki/index.php/CoFI\\_2014](http://www.informatik.uni-bremen.de/cofi/wiki/index.php/CoFI_2014)
10. Dunkel, J., Brun, R. (2007). Model-Driven Architecture for Mobile Applications. Lecture Notes in Computer Science 4439, pág. 464-477, Springer.
11. Eclipse (2014). The eclipse modeling framework. <http://www.eclipse.org/emf/>
12. Favre, L. (2009) A Formal Foundation for Metamodeling. Ada-Europe 2009: Lecture Notes in Computer Science 5570, Springer-Verlag, pág. 177-191.
13. Favre, L. (2010) Model Driven Architecture for Reverse Engineering Technologies: Strategic Directions and System Evolution. IGI Global, Premier Source Reference
14. HAXE (2014) [www.haxe.org](http://www.haxe.org)
15. Jackson, E., Levendovszky, T., Balasubramanian, D. (2011) Reasoning about Metamodeling with Formal Specifications and Automatic Proofs, in MODELS 2011.
16. KDM (2012). Knowledge Discovery Metamodel, version 1.3, OMG Document Number: formal/2012-05-08. <http://www.omg.org/spec/KDM/1.3>
17. MDA. (2014) Model Driven Architecture <http://www.omg.org/mda>
18. MoDisco (2014). <http://www.eclipse.org/Modisco>
19. MOF (2011) MOF: Meta Object Facility (MOF™) Core Specification Version 2.4.1, OMG Document Number: formal/2011-08-07. [www.omg.org/spec/MOF/2.4.1](http://www.omg.org/spec/MOF/2.4.1)
20. OMG (2014) [www.omg.org](http://www.omg.org)

21. Parr, Terence. (2013) The Definitive ANTLR 4 Reference (1st ed.), Pragmatic Bookshelf, p. 328, ISBN 978-1-93435-699-9
22. QVT (2014). QVT: MOF 2.0 Query, View, Transformation. Version 1.1, OMG Document Number: [www.omg.org/spec/QVT/1.2/](http://www.omg.org/spec/QVT/1.2/)
23. Rose, L. M. (2011). Structures and Processes for Managing Model-Metamodel Co-evolution. PhD. Thesis University of York
24. Tonella, P., Potrich, A. (2005). Reverse Engineering of Object Oriented Code. Monographs in Computer Science. Heidelberg: Springer-Verlag.
25. Vermolen, S., Wachsmuth, G., Visser, E. (2011). Reconstructing Complex Metamodel Evolution. SLE 2011, Lecture Notes in Computer Science 6940, pág. 201-222, Springer.
26. EMF- Eclipse Modeling Framework (2014) [www.eclipse.org](http://www.eclipse.org)
27. Wasserman, A. (2010). Software engineering issues for mobile application development. Proceedings of the FSE/SDP workshop on Future of software engineering research, FoSER '10, pages 397- 400, New York, USA, ACM.