

CARRERA DEL INVESTIGADOR CIENTÍFICO Y TECNOLÓGICO

Informe Científico

PERIODO: 1/1/2016 al 31/12/2017

1. DATOS PERSONALES

APELLIDO: Favre

NOMBRES: Liliana María

Dirección Particular: Calle:

Localidad: Tandil CP: 7000 Tel:

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 2016 y 2017 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/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 en el contexto de ingeniería dirigida por modelos (MDE- *Model Driven Engineering*), técnicas de ingeniería inversa tradicionales de análisis estático y dinámico, y técnicas de metamodelado formal 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. Se propone en la sección 22 dar continuidad a la investigación en “Modernización de Software”.

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

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.

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]. El objetivo de esta investigación es trabajar en la modernización de software bajo la iniciativa impulsada por OMG (*Object Management Group*) denominada ADM (*Architecture Driven Modernization*) [3].

ADM provee estándares para favorecer la interoperabilidad entre herramientas, factor clave para procesos de modernización con modelos. 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 [32]. 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 [36]. Otro concepto fundamental en MDA es el de transformaciones entre modelos. Todo proceso MDD puede verse como una secuencia de transformaciones de modelos. Una transformación es el proceso de convertir un modelo

origen que conforma a un metamodelo origen en un modelo destino que conforma a un metamodelo destino. Las transformaciones pueden ser de diferentes tipos: Texto-a-Modelo (T2M), Modelo-a-Modelo (M2M) o Modelo-a-Texto (M2T).

Se propuso en esta investigación una integración de ADM con técnicas clásicas de ingeniería inversa de análisis estático y dinámico. 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. Además, la construcción de metamodelos se basó en una integración de técnicas semiformales basadas en metamodelos Ecore con técnicas formales basadas en el formalismo algebraico [20].

Se analizaron diversos escenarios de modernización poniendo el énfasis en la modernización de software orientado a objetos a fin de adaptarlo a tecnologías móviles. Los resultados más relevantes de este período de investigación se vinculan a la definición de procesos de modernización de software orientado a objetos a través de la integración de desarrollos multiplataforma con desarrollos dirigidos por modelos. En este marco se trabajó en la migración de software Java y C++ al lenguaje multiplataforma Haxe [22]. Se definieron nuevos metamodelos para los lenguajes C++ y Haxe y un repertorio de transformaciones para integrar Java y C++ con Haxe.

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

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, tesis de grado de la carrera de Ingeniería de Sistemas y empresarios. Se incluye en este informe un resumen de cada publicación y se adjunta una copia de las mismas y documentación respaldatoria.

8.1 PUBLICACIONES.

CAPÍTULOS

[8.1.1]

Díaz Bilotto, Pablo
(Universidad Nacional del Centro de la Provincia de Buenos Aires)

Favre, Liliana

(Universidad Nacional del Centro de la Provincia de Buenos Aires & Comisión de Investigaciones Científicas de la Provincia de Buenos Aires)

Capítulo 13: "Migrating JAVA to Mobile Platforms through HAXE: An MDD Approach" en "Modern Software Engineering Methodologies for Mobile and Cloud Environments". (DOI:

10.4018/978-1-4666-9916-8.ch013). Editores: Antonio Miguel Cruz, Sara Paiva. ISBN13: 9781466699168|ISBN10: 1466699167|EISBN13: 9781466699175|DOI: 10.4018/978-1-4666-9916-8, páginas 240-268. IGI GLOBAL, USA. 2016.

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

[8.1.2]

Favre, Liliana

(Universidad Nacional del Centro de la Provincia de Buenos Aires & Comisión de Investigaciones Científicas de la Provincia de Buenos Aires)

Martinez, Liliana

(Universidad Nacional del Centro de la Provincia de Buenos Aires)

Pereira, Claudia

(Universidad Nacional del Centro de la Provincia de Buenos Aires)

Volumen X, Capítulo 648 “Model Driven Software Modernization” Capítulo invitado en Encyclopedia of Information Science and Technology, 4 Edition (10 Vol, 8104 páginas) (Editor Medhi Khosrow-Pour). ISBN13: 9781522522553|ISBN10: 1522522557|EISBN13: 9781522522560|DOI: 10.4018/978-1-5225-2255-3. páginas 7447-7458, IGI GLOBAL, USA, 2018.

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

[8.1.3.]

Favre, Liliana

(Universidad Nacional del Centro de la Provincia de Buenos Aires & Comisión de Investigaciones Científicas de la Provincia de Buenos Aires)

Martinez, Liliana

(Universidad Nacional del Centro de la Provincia de Buenos Aires)

Pereira, Claudia

(Universidad Nacional del Centro de la Provincia de Buenos Aires)

Volumen 3, Capítulo 60: "Reverse Engineering of Object-Oriented Code: An ADM Approach" en "Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications" (4 volúmenes, 2087 páginas). Editor Mehdi Khosrow-Pour, DBA. ISBN13: 9781522539230|ISBN10: 1522539239|EISBN13: 9781522539247|DOI: 10.4018/978-1-5225-3923-0, páginas 1479-1502. Information Resources Management Association, IGI GLOBAL, USA, 2018.

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

[8.1.4]

Favre, Liliana.

(Universidad Nacional del Centro de la Provincia de Buenos Aires & Comisión de Investigaciones Científicas de la Provincia de Buenos Aires)

"A framework for modernizing non-mobile software: A Model Driven Engineering Approach". Capítulo 7 en "Protocols and Applications for the Industrial Internet of Things". Editores: Cristian González García, Vicente García-Díaz), B. Cristina Pelayo García-Bustelo, Juan Manuel Cueva Lovelle. 356 páginas. DOI: 10.4018/978-1-5225-3805-9.ch007, ISBN13: 9781522538059|ISBN10: 1522538054|EISBN13: 9781522538066|DOI: 10.4018/978-1-5225-3805-9, páginas 192-224, Information Resources Management Association, IGI GLOBAL, USA, 2018.

Abstract

New paradigms such as pervasive computing, cloud computing, and the internet of things (IoT) are transforming the software industry and the business world. Organizations need to redesign their models and processes to be sustainable. Smartphones are at the core of these paradigms, letting us locate and easily interact with the world around us. Frequently, the development of mobile software requires of the adaption of valuable and tested non-mobile software. Most challenges in this kind of software modernization are related to the diversity of

platforms on the smartphones market and to the need of systematic and reusable processes with a high degree of automation that reduce time, cost, and risks. This chapter proposes a modernization framework based on model-driven engineering (MDE). It allows integrating legacy code with the native behaviors of the different mobile platform through cross-platform languages. Realizations of the framework for the migration of C/C++ or Java code to mobile platforms through the Haxe multiplatform language.

Keywords: Model Driven Engineering; Model Driven Architecture; Model Transformation; Mobile Computing; Internet of Things; Haxe; Atlas Transformation Language

[8.1.5.]

Favre, Liliana

(Universidad Nacional del Centro de la Provincia de Buenos Aires & Comisión de Investigaciones Científicas de la Provincia de Buenos Aires)

Martinez, Liliana

(Universidad Nacional del Centro de la Provincia de Buenos Aires)

Pereira, Claudia

(Universidad Nacional del Centro de la Provincia de Buenos Aires)

Volumen 1, Capítulo 17: "Reverse Engineering of Object-Oriented Code: An ADM Approach" en "Intelligent Systems: Concepts, Methodologies, Tools, and Applications" (4 Vol., 2351 páginas). ISBN13: 9781522556435|ISBN10: 1522556435|EISBN13: 9781522556442|DOI: 10.4018/978-1-5225-5643-5, páginas 424-447, Information Resources management Association, IGI GLOBAL, USA, 2018

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

REVISTAS

[8.1.6]

Martinez, Liliana

(Universidad Nacional del Centro de la Provincia de Buenos Aires)

Pereira, Claudia

(Universidad Nacional del Centro de la Provincia de Buenos Aires)

Favre, Liliana

(Universidad Nacional del Centro de la Provincia de Buenos Aires & Comisión de Investigaciones Científicas de la Provincia de Buenos Aires)

“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.4(3): 34-44 (2017)

<http://ijimai.org/journal/>

Abstract

Software technology is constantly evolving and therefore the development of applications requires adapting software components and applications in order to be aligned to new paradigms such as Pervasive Computing, Cloud Computing and Internet of Things. In particular, many desktop software components need to be migrated to mobile technologies. This migration faces many challenges due to the proliferation of different mobile platforms. Developers usually make applications tailored for each type of device expending time and effort. As a result, new programming languages are 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 Architecture-Driven Modernization (ADM) proposed by the Object Management Group (OMG) can help to manage a huge diversity of mobile technologies. The Architecture-Driven Modernization Task Force (ADMTF) was formed to create specifications and promote industry consensus on the modernization of existing applications. In this work, we propose a migration process from C/C++ software to different mobile platforms that integrates ADM standards with Haxe. We exemplify the different steps of the process with a simple case study, the migration of “the Set of Mandelbrot” C++ application. The proposal was validated in Eclipse Modeling Framework considering that some of its tools and run-time environments are aligned with ADM standards.

Keywords

Architecture-Driven Modernization, Haxe, Migration, Metamodeling, Mobile Platform, Model-Driven Development

CONFERENCIAS INTERNACIONALES

[8.1.7]

Favre, Liliana

(Universidad Nacional del Centro de la Provincia de Buenos Aires & Comisión de Investigaciones Científicas de la Provincia de Buenos Aires)

Duarte, Daniel

(Universidad Nacional del Centro de la Provincia de Buenos Aires)

“Formal MOF Metamodeling and Tool Support”. Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development. MODELSWARD 2016. Páginas 99-110. Roma, Italia, 19 de Febrero de 2016. ISBN: 978-989-758-168-7 SCITEPRESS (Science and Technology Publications). Publicado en IEEE Xplore en junio de 2017.

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)

[8.1.8]

Favre, Liliana

(Universidad Nacional del Centro de la Provincia de Buenos Aires & Comisión de Investigaciones Científicas de la Provincia de Buenos Aires)

Martinez, Liliana

(Universidad Nacional del Centro de la Provincia de Buenos Aires)

Pereira, Claudia

(Universidad Nacional del Centro de la Provincia de Buenos Aires)

“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.(Editores M. Papadrakakis, V. Papadopoulos,G. Stefanou,V. Plevris) ISBN:978-618-82844-0-1 5 - 10 JUNE 2016 Crete Island, Greece Trabajo 4096. 15 páginas 8162-8176, 2016

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++.

[8.1.9]

Favre, Liliana

(Universidad Nacional del Centro de la Provincia de Buenos Aires & Comisión de Investigaciones Científicas de la Provincia de Buenos Aires)

Bricker, Federico

(Dokko Group)

“Migrating software to mobile technology: A Model Driven Engineering Approach”. Proceedings of the VII International Conference on Coupled Problems in Science and Engineering Rhodes Islad, Greece June 12 – 14, (1300 páginas) Edited by: Manolis Papadrakakis, Eugenio Oñate, Bernhard A. Schrefler. Publicado por International Center for Numerical Methods in engineering (CIMNE), páginas 912-923, Barcelona, España, 2017.

Abstract

Nowadays, software industry evolves to tackle new approaches aligned with new paradigms such as Ubiquitous Computing, Mobile Computing, Cloud Computing or IoT (Internet of

Things). Future embedded and ubiquitous systems will operate continuously on mobile devices such as smartphones which come with a variety of sensors (GPS, accelerometer, digital compass, microphone, and camera) enabling a wide range of mobile applications related to the mentioned paradigms. Mobile computing originated advances in the electronic miniaturization and communication technology facilitating the development of distributed processing to create networks of billions of devices such as 3G, 4G and 5G. In summary, Mobile Computing is crucial to harnessing the potential of Ubiquitous Computing, Cloud Computing and IoT and, without the existence of smartphones these paradigms would not exist. The development of Science & Engineering applications requires adapting desktop software components to these new paradigms. There are many of these desktop applications still critical and their complete replacement is dangerous. Model Driven Engineering (MDE) is a promising approach to solve this problema. Technical frameworks for information integration and tool interoperability aligned to MDE can help to manage a huge diversity of new technologies and mobile platforms. On the other hand, 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 easily allows adapting the native behavior of different platforms such as Android, iOS and BlackBerry, in a straightforward way. Our goal focuses on the migration of non-mobile software to mobile platforms as a mean of software modernization. In this work, we describe a migration process from C/C++ software to new technologies that integrates MDE with the Haxe language. The process includes three phases: reverse engineering, restructuring/refactoring and forward engineering. It follows model-driven principles: all artifacts involved in the process are viewed as models that conform to standard metamodels and the migration process is divided in smaller steps focusing in specific activities that can be automated thanks to the chaining of model transformations. All the involved software artifacts share a common format allowing interoperability and can be reused, modified for evolution purposes or extended for other purposes. An important issue of our approach is to provide support for reasoning and verification in the modernization of systems that are critical to safety, security and economic profits. The approach was validated in the Eclipse Modeling Framework (EMF) that provides tools and run-time environments aligned with MDE principles. Case studies in Science & Engineering domain that show the advantages of our approach were developed.

Keywords

Model Driven Development; Model Driven Engineering; Software Modernization; Mobile Development; Haxe; Eclipse Modeling Framework

8.2 TRABAJOS EN PRENSA Y/O ACEPTADOS PARA SU PUBLICACIÓN.**[8.2.1]**

Favre, Liliana. "Migrating software towards mobile technologies: An integration of Model Driven Engineering with Cross-Platform Development". Encyclopedia of Organizational Knowledge, Administration and Technologies", IGI GLOBAL, USA. Propuesta aceptada.marzo de 2018. <https://www.igi-global.com/publish/call-for-papers/call-details/3118>

8.3 TRABAJOS ENVIADOS Y AÚN NO ACEPTADOS PARA SU PUBLICACIÓN.

8.4 TRABAJOS AÚN NO ENVIADOS PARA SU PUBLICACIÓN.

Se prevé enviar para su evaluación en los próximos meses dos trabajos relacionados al proyecto:

- *Model-Driven Cross-Platform Development*

Describe una infraestructura para la integración de desarrollos multiplataforma y desarrollos dirigidos por modelos con la finalidad de modernizar software *legacy* a fin de adaptarlo a nuevas tecnologías móviles.

- *Reasoning about MOF Metamodeling*

Describe los últimos avances en metamodelado formal y la evolución de artefactos de desarrollo especificados por metamodelos MOF en el marco del DSL (*Domain Specific Language*) Nereus.

También continúo con la redacción de un libro de autoría sobre modernización de software basada en *Model Driven Engineering*. Presentará los resultados originales del enfoque de esta investigación.

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 definieron metamodelos Ecore para el lenguaje multiplataforma Haxe y para el lenguaje C/C++. Cabe destacar que no existían metamodelos Ecore alineados a MOF 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.
- b) 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. Se

definieron varias transformaciones que pueden considerarse una contribución a la comunidad MDD, por ejemplo las transformaciones C/C++2Haxe o Java2Haxe.

- c) 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 modelos a nivel de PSM (Platform-Specific Models) en UML [42].
- d) 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. MOF. Se concretó el desarrollo de un parser de Nereus y su integración con herramientas de prueba formal para actividades de testing y verificación. Actualmente se encuentra en ejecución un plan para el desarrollo de herramientas de metamodelado formal que soporten la construcción rigurosa de nuevos metamodelos.

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

El lenguaje de metamodelado formal Nereus puede ser utilizado en cursos básicos para la enseñanza de las bases de la programación orientada a objetos. 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).

Teniendo en cuenta que la semántica del lenguaje Nereus fue dada por traducción al lenguaje CASL (*Common Algebraic Specification Language*) y que CASL está conectado a diferentes demostradores de teoremas a través de HETS [26], es posible la realización de pruebas formales sobre una especificación Nereus. Además, Nereus permite especificar metamodelos expresados como diagramas que incluyen clases y sus diferentes relaciones (generalización, dependencia, asociaciones, agregación y composición) y puede ser utilizado en la enseñanza de especificaciones formales en el marco de la ingeniería de software. En esta dirección se prevé presentar en el transcurso de este año un trabajo concluido que propone este enfoque: *“Introducing Formal Specification to Software Engineering Students: A Formal Language and Tool Support”*.

11.2 DIVULGACIÓN

12. DIRECCIÓN DE BECARIOS Y/O INVESTIGADORES.

12.1. Dirección de proyectos

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 y extendido para el año 2018

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 2016-2017 a los siguientes profesores:

Liliana Martínez.

Jefe de trabajos prácticos- Dedicación exclusiva- Facultad de Ciencias Exactas- UNCPBA- 2016

Profesora Adjunta. Dedicación exclusiva. Facultad de Ciencias Exactas. UNCPBA.2017-2018

Claudia Pereira

Jefe de trabajos prácticos- Dedicación exclusiva- Facultad de Ciencias Exactas- UNCPBA. 2016

Profesora Adjunta. Dedicación exclusiva. Facultad de Ciencias Exactas. UNCPBA.2017-2018

Federico Améndola

Ayudante de primera, dedicación simple- Facultad de Ciencias Exactas- UNCPBA 2016-2018

13. DIRECCIÓN DE TESIS.

Durante el período informado he dirigido a los siguientes **tesistas 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:

Maximiliano Duthey

Título de la tesis: “Migración de software C/C++ a plataformas móviles a partir de MDD (Model Driven Development)”

Directora: Liliana Favre

Presentado en 2016. Calificación: 10(diez)

Carolina Spina

Título de la tesis: “Migración de software C/C++ a plataformas móviles a partir de MDD (Model Driven Development)”

Directora: Liliana Favre

Presentado en 2016. Calificación: 10(diez)

Cabe destacar que la Tesis “*Migración de software C/C++ a plataformas móviles a partir de MDD (Model Driven Development)*” fue desarrollada a través de Becas de Entrenamiento CIC de los alumnos Carolina Spina y Maximiliano Duthey bajo mi dirección.

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. 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 2016-2017 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

- Integrante del Advisory Board de la revista internacional “Journal of Cases on Information Technology” (JCIT), Information Resources Management Resources Association, ISSN 1048-7717, USA 2004-2017-2018
- Evaluadora de artículos de Journal of Systems and Software, Elsevier (ISSN 0164-1212), 2016-2017-2018
- Evaluadora de artículos Computers & Security, Elsevier (ISSN 0167-4048), 2016-2017
- Integrante de la Comisión Evaluadora de la Categorización del Programa de Incentivos a Docentes-Investigadores según Resolución 1543/2015 del Ministerio de Educación (artículo 17). Regional Metropolitana- Área: Ingeniería. Noviembre de 2017.
- Evaluadora de trabajos en CONAIISI 2016. 4to Congreso Nacional de Ingeniería Informática/Sistemas de información. Salta, Argentina.

- Jurado docente. Carrera Académica. Área Ciencia de la Computación. Facultad de Ingeniería. Olavarría, UNCPBA. 2016
- Evaluadora de trabajos en CACIC 2016. XXII Congreso Argentino de Ciencias de la Computación CACIC 2016, San Luis, Argentina.
- Evaluadora de Informes de Avance de proyectos de investigación de la Universidad Nacional de la Matanza, Departamento de Ingeniería e Investigaciones Tecnológicas. 2016.
- 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 2016
- Evaluadora de presentación de proyectos de investigación de la Universidad Nacional de La Matanza, Departamento de Ingeniería e Investigaciones Tecnológicas. Mayo de 2016
- Evaluadora de proyectos de investigación de la Universidad Tecnológica Nacional, Rectorado. 2016.
- Evaluadora de Informes de Avance de proyectos de investigación de la Universidad Nacional de la Matanza, Departamento de Ingeniería e Investigaciones Tecnológicas. Abril de 2017.
- 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. Abril de 2017
- Evaluadora de presentación de proyectos de investigación de la Universidad Nacional de La Matanza, Departamento de Ingeniería e Investigaciones Tecnológicas. Abril de 2017
- Evaluadora de proyectos de investigación de la Universidad Tecnológica Nacional, Rectorado. 2017.
- Evaluadora de trabajos en CONAIISI 2017. 4to Congreso Nacional de Ingeniería Informática/Sistemas de información. Santa Fe, Argentina.
- Evaluadora de trabajos en CACIC 2017. XXIII Congreso Argentino de Ciencias de la Computación CACIC 2017, La Plata, Argentina.
- Jurado docente carrera académica. Facultad de Ingeniería. Universidad Nacional del Centro, Olavarría. RJE 6816/17. Mayo de 2018.
- Evaluadora de Informes de Avance de proyectos de investigación de la Universidad Nacional de la Matanza, Departamento de Ingeniería e Investigaciones Tecnológicas. 2018.
- 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. 2018

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

Título: Modernización de software

Resumen

Nuevos paradigmas como *Cloud Computing* y “*The Internet of Things*” (IoT) están transformando la industria de software y el mundo de los negocios. Las empresas deben rediseñar sus procesos y modelos a fin de lograr ventajas competitivas y ser sustentables. La computación móvil (*Mobile Computing*) y la miniaturización electrónica permitieron el advenimiento de estos nuevos paradigmas. Así como los smartphones desplazaron a las cámaras fotográficas, al GPS, a los reproductores de música y a las billeteras, hoy están en el centro de la escena de IoT que involucra personas y objetos operando sinérgicamente siendo la principal interfaz que conecta a las personas con internet. El desarrollo de software alineado a estos nuevos paradigmas requiere adaptar software *legacy*. Los desafíos en este tipo de migraciones tienen que ver con la proliferación de plataformas móviles de desarrollo y con la carencia de procesos reutilizables y sistemáticos con un alto grado de automatización que reduzcan riesgos, tiempo y costos. Con respecto al primer desafío, la situación ideal es trabajar con desarrollos multiplataforma (*Cross-Platform*) que permitan implementar una aplicación y desplegarla automáticamente sobre diversas plataformas. Por otra parte, MDE (*Model Driven Engineering*), una disciplina de desarrollo de software que enfatiza en el uso de modelos en diferentes niveles de abstracción y transformaciones de modelos, puede aportar a la industrialización de la modernización de software. Se propone en esta investigación combinar MDE y desarrollos multiplataforma a fin de adaptar software *legacy* a nuevas tecnologías tales como *Mobile Computing*, *IoT* y *Cloud Computing*.

Palabras claves: Modernización de software, Model-Driven Engineering, Cloud Computing, Mobile Computing, The Internet of Things-IoT

1. Introducción

Nuevos paradigmas como *Cloud Computing* o *Internet of Things* (IoT) están transformando la industria de software y el mundo de los negocios. Las empresas deben rediseñar sus procesos y modelos a fin de lograr ventajas competitivas y ser sustentables. La computación móvil y la miniaturización electrónica permitieron el advenimiento de estos nuevos paradigmas. Así como los smartphones desplazaron a las cámaras fotográficas, al GPS, a los reproductores de música y a las billeteras hoy están en el centro de la escena de IoT que involucra personas y objetos operando sinérgicamente. Los smartphones desempeñan un rol central en el contexto de IoT al ser la principal interfaz que conecta a las personas con internet.

Frecuentemente el desarrollo de software alineado a estos nuevos paradigmas requiere adaptar software *legacy* (heredado). Los desafíos en este tipo de migraciones tienen

que ver con la proliferación de plataformas móviles de desarrollo y con la carencia de procesos reutilizables y sistemáticos con un alto grado de automatización que reduzcan riesgos, tiempo y costos. Con respecto al primer desafío, la situación ideal es trabajar con desarrollos multiplataforma (*Cross-Platform*). Por ejemplo Haxe es un lenguaje multiplataforma *open-source* que permite producir aplicaciones y código fuente para varias plataformas diferentes desde un simple código base. La idea detrás de Haxe es permitir a los desarrolladores elegir la mejor plataforma para un específico desarrollo [22].

Por otra parte, MDE (*Model Driven Engineering*), una disciplina de desarrollo de software que enfatiza en el uso de modelos en diferentes niveles de abstracción y transformaciones de modelos, puede aportar a la industrialización del desarrollo de procesos de modernización de software [9].

Los principios Model-Driven pueden resumirse como sigue: todos los artefactos involucrados en un proceso MDE pueden ser vistos como modelos que conforman a un metamodelo particular, el proceso en sí mismo puede ser visto como una secuencia de transformaciones de modelos y toda la información extraída es representada en una forma estándar a través de metamodelos. Es decir, modelos, metamodelos y transformaciones son cruciales en MDE. En este enfoque, MDD (*Model Driven Development*) refiere a procesos de ingeniería directa (*forward engineering*) que usan modelos y transformaciones como artefactos primarios. Una realización específica de MDD es la arquitectura MDA propuesta por OMG [32].

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 modelado 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 [36].

Los procesos MDA pueden verse como una secuencia de transformaciones de modelos. Una transformación es el proceso de convertir un modelo origen que conforma a un metamodelo origen en un modelo destino que conforma a un metamodelo destino. Las transformaciones pueden ser de distinto tipo: T2M (Texto-a-Modelo), M2M (Modelo-a-Modelo), o M2T (Modelo-a-Texto) y soportan diferentes procesos de ingeniería inversa (*reverse engineering*), reestructuración/refactoring e ingeniería directa (*forward engineering*). El estándar relacionado a transformaciones de modelos es el metamodelo MOF 2.0 *Query, View, Transformation (QVT)* [41].

Actualmente existe una creciente demanda de soporte para la modernización de sistemas y OMG está involucrado en el desarrollo de estándares para la modernización de software a través de la iniciativa ADM (*Architecture Driven Modernization*) [3]. ADM sigue los principios esenciales de MDA. Los estándares (metamodelos) más relevantes en el contexto de ADM son KDM (*Knowledge Discovery Metamodel*) [27],

ASTM (*Abstract Syntax Tree Metamodel*) [4] y SMM (*Software Metrics Metamodel*) [44]. En el contexto de MDE, se referencia como *Model-Driven Software Modernization* (MDSM) a una forma de reingeniería para la evolución funcional y tecnológica de sistemas de software [9].

En esta investigación se propone combinar MDE, MDSM y desarrollos multiplataforma a fin de adaptar software legado a nuevas tecnologías tales como *Cloud Computing*, *IoT* y *Mobile Computing*.

2. Objetivo general

Analizar diferentes escenarios de migración de software *legacy* hacia nuevas tecnologías (*Cloud Computing*, *Internet of Things* y *Mobile Computing*) cubriendo las fases de ingeniería inversa (*reverse engineering*) y directa (*forward engineering*) a partir de la integración de desarrollos de ingeniería dirigida por modelos (*Model-Driven Engineering*) con desarrollos multiplataforma (*Cross-Development*).

3. Metodología

Para lograr los objetivos del proyecto, se propone integrar técnicas de metamodelado semiformales y formales con técnicas clásicas de ingeniería inversa basadas en análisis estático y dinámico. Las técnicas semiformales de metamodelado se basarán en estándares (metamodelos) de MDA y ADM tales como MOF, KDM y ASTM.

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 MDE, en particular MDA y ADM. El subproyecto EMF (*Eclipse Modeling Framework*) es un framework de modelado que provee una infraestructura para el desarrollo de herramientas tales como editores, motores de transformación y soporte para metamodelado [19]. 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 [31]. 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 [5] [41]. 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 [35].

Específicamente, de los posibles escenarios de modernización, se propone analizar la migración de software a diversas plataformas a través de metamodelos de lenguajes multiplataforma que pueden verse como una especificación intermedia para diversas plataformas.

Se construirán metamodelos MOF rigurosamente integrando técnicas semiformales de especificación de metamodelos Ecore con técnicas rigurosas basadas en el

formalismo algebraico, específicamente en el lenguaje de metamodelado formal Nereus resultado de este proyecto.

4. Grado de avance

Esta investigación dará continuidad a la realizada en el proyecto “Tecnologías de software” en el período 2016-2017. En la misma, se investigaron formalismos y teorías para analizar, administrar y controlar la evolución del software en el contexto de MDA y ADM.

Específicamente, se trabajó en la definición de procesos de modernización de software desarrollado en C/C++ o Java a fin de adaptarlo a diversas plataformas móviles a través del lenguaje multiplataforma Haxe. Se definieron procesos en el contexto de ADM y sus estándares en particular KDM. Asimismo se experimentó con la migración a nivel de modelos de código en el contexto de MDA y sus estándares, en particular metamodelos MOF. Se definieron metamodelos para los lenguajes C/C++ y Haxe y varias transformaciones por ejemplo C/C++2Haxe o Java2Haxe.

Se extendieron resultados previos relacionados a la definición del lenguaje Nereus de metamodelado formal respecto a la evolución de artefactos de desarrollo descritos por metamodelos MOF.

Se avanzó en dirección a lograr productos innovadores transferibles con la empresa tandilense Dokko Group.

Las publicaciones incluidas en la documentación respaldatoria describen estos avances.

5. Actividades propuestas para el período 2018-2019

De acuerdo al objetivo del proyecto se propone analizar diferentes escenarios de migración de software *legacy* para adaptarlo a nuevas tecnologías (*Cloud Computing*, *IoT* y *Mobile Computing*). Se propone usar desarrollos MDE basados en los estándares de ADM y MDA para aumentar el grado de automatización en las etapas de ingeniería inversa de software y de ingeniería directa hacia nuevas tecnologías.

Se propone trabajar con una integración de desarrollos dirigidos por modelos y desarrollos multiplataforma. Esto implicará definir metamodelos de lenguajes multiplataforma y transformaciones específicas de acuerdo a los estándares de ADM y MDA.

Dependiendo de las características de la modernización se propone abordarla bajo dos enfoques: uno alineado a ADM, a nivel de modelos independientes de una plataforma (*Platform Independent Model- PIM*) en KDM y otro, a nivel de modelos específicos de una plataforma (*Platform Specific Model- PSM*). En el primer enfoque, considerando que desde KDM es posible generar modelos UML a nivel de PIM, y que no existen en general generadores de código desde UML para lenguajes multiplataformas (por ejemplo Haxe), será necesario abordar esta actividad.

Teniendo en cuenta las características de los desarrollos MDE y que la demanda de los desarrolladores se centra en las especificaciones de metamodelos y transformaciones, se propone proveer un repositorio de los mismos que signifique una contribución para otros desarrolladores MDE.

Además, continuando con investigaciones previas se extenderán resultados relacionados a definición del DSL Nereus para metamodelado formal. 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 [26]. 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 de teoremas (*Automatic Theorem Provers – ATP*) tales como Isabelle y Spass. Se definirán procesos para administrar la evolución de metamodelos MOF especificados algebraicamente. Se propone analizar la definición de componentes reutilizables de metamodelado y la aplicación de operadores de reutilización para adaptarlas a fin de construir nuevos metamodelos o evolucionar existentes.

Las actividades propuestas las desarrollaría en el marco del proyecto “Tecnologías de software” con el aporte de investigadores, tesistas o pasantes.

Se propone avanzar en dirección a lograr productos innovadores transferibles a partir de los contactos ya establecidos con una empresa tandilense que desarrolla aplicaciones para dispositivos móviles Dokko Group (<http://en.dokkogroup.com.ar/>).

6. Referencias y bibliografía

1. Acceleo. Obeo. Acceleo Generator. <http://www.eclipse.org/acceleo/>, 2018.
2. Acerbis, R., Bongio, A., Brambilla, M., Butti, S. Model-Driven Development Based on OMG's IFML with WebRatio Web and Mobile Platform. Engineering the Web in the Big Data Era. Lecture Notes in Computer Science (vol. 9114, pp. 605-608). Berlin: Springer-Verlag, 2015.
3. ADM. Architecture-Driven Modernization. <http://adm.omg.org> , 2018.
4. ASTM. Abstract Syntax Tree Metamodel, version 1.0, OMG Document Number: formal/2011-01-05 www.omg.org/spec/ASTM, 2011.
5. ATL. Atlas Transformation Language (ATL) www.eclipse.org/ATL/documentation, 2018.
6. Barbier, P., Casteran, E., Cariou, E., le Goaer, O. Adaptative 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.
7. 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.

8. Bowen, J., Hinze, A. Supporting mobile application development with model driven emulation. *Journal of the ECEASST*, Volumen 45 (pp. 1-5), 2011.
9. Brambilla, M., Cabot, J., Wimmer, M. *Model Driven Software Engineering in Practice. Synthesis Lectures on Software Engineering*. Morgan & Claypool Publishers, 2012.
10. Braun, P., Eckhaus, R. Experiences on model-driven software development for mobile applications. In *Proceedings of Engineering of Computer-Based Systems, IEEE International Conference and Workshop on the Engineering of computer Base Systems* , (pp. 490-493), Washington, DC, USA: IEEE Computer Society, 2008.
11. Bruneliere, H., Cabot, J., Dupe, G. , Madiot, F. *MoDisco: a Model Driven Reverse Engineering Framework*. Information and Software Technology, 2014.
12. Canfora, G., Dipenta, M., Ceruelo, L. Achievements and Challenges in Software Reverse Engineering. *Communications of the ACM*, 54 (4), 142-151, 2011.
13. Cannase, N.. Haxe. Too Good to be True? GameDuell Tech Talk. <http://www.techtalk-berlin.de/news/read/nicolas-cannasse-introducing-Haxe/>, 2018.
14. 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.
15. Dasnois, B. *Haxe 2 Beginner's Guide*. Packt Publishing, 2011
16. Dehlinger, J., & Dixon, J. Mobile application software engineering: Challenges and research directions. In *Proceedings of the Workshop on Mobile Software Engineering* (pp. 29-32), Berlin: Springer-Verlag, 2011.
17. 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
18. Ejarque, J., Micsik, A., Badia, R. Towards Automatic Application Migration to Clouds. In *Proceedings IEEE 8th Int. Conf. on Cloud Computing (CLOUD)* (pp. 25-32), New York, USA, 2015.
19. Eclipse. The Eclipse Modeling Framework. <https://eclipse.org/modeling/emf/>, 2018.
20. Favre, L. A Formal Foundation for Metamodeling. In *Proceedings of 14th Ada-Europe International Conference. Lecture Notes in Computer Science* (Vol 5570, pp.177-191). Berlin: Springer-Verlag, 2009.
21. Favre, L. *Model Driven Architecture and Reverse Engineering Technologies: Strategic Directions and System Evolution (Premier Source Reference)* ISBN 9781615206490. IGI GLOBAL, 2010.
22. Haxe. The Haxe Language Retrieved April 25, 2017 from <https://haxe.org/>, 2018.
23. Islam, N., Want, R. Smartphones: Past, present and future. *Pervasive Computing*, 13(4), 82-92, 2014.

24. 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.
25. Joshi, P., Nivangune, A., Kumar, R., Kumar, S., Ramesh, R., Pani, S., Chesum, A. Understanding the Challenges in Mobile Computation Offloading to Cloud through Experimentation. In *2nd ACM Int. Conf. on Mobile Software Engineering and Systems (MOBILESoft)*, (pp.158-159), Florence, Italy, 2015.
26. HETS, Heterogeneous Tool Set. http://www.informatik.uni.remenn.de/agbkb/forschung/formal_methods/CoFI/HETS/, 2018.
27. KDM . Knowledge Discovery Metamodel, version 1.3, OMG Document Number: formal/2011-08-04. www.omg.org/spec/KDM/1.3, 2011.
28. Kim, H. Frameworks of process improvement for mobile applications. *Engineering Letters*, 16(4) pp. 550-555, 2008.
29. Kramer, D., Clark, T., Oussena, S. MobDSL: A domain specific language for multiple mobile platform deployment. In *Proceedings of IEEE Int. Conf. on Networked Embedded Systems for Enterprise Applications (NESEA)*, (pp. 1-7), Perth, Australia, 2010.
30. Lettner, M., Tschernuth, M., Mayrhofer, R. 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, 2012.
31. MMT. Model-to-Model Transformation. Eclipse Modeling Framework. 2016 <https://www.eclipse.org/mmt/>, 2015.
32. MDA, Object Management Group Model Driven Architecture (MDA) MDA Guide rev. 2.0, OMG Document ormsc/2014-06-01, 2014.
33. Miranda, J., Makitalo, N., Garcia-Alonso, J., Berrocal, J., Mikkonen, T., Canal, C.,
34. Murillo, J. From the Internet of Things to the Internet of People. *IEEE Internet Computing*, 19(2), 40–47. doi: 10.1109/MIC.2015.24, 2015.
35. MoDisco. Model Discovery. <http://www.eclipse.org/MoDisco>, 2018.
36. MOF, OMG Meta Object Facility Core Specification, version 2.4.2, <http://www.omg.org/spec/MOF/2.4.2>, 2014.
37. MOFM2T. MOF Model to Text Transformation Language, Version 1.0. <http://www.omg.org/spec/MOFM2T/1.0/> , 2008.
38. 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.
39. Parr, T. *The Definitive ANTLR 4 Reference* (1st ed.) , Pragmatic Bookshelf, 2013.
40. Pérez Castillo, R., García-Rodríguez de Guzmán, I. Piattini, M. "Knowledge Discovery Metamodel-ISO-IEC 19506: A standard to modernize" *Computer Standards & Interfaces* 33 (2011) 519-532, 2011.

41. 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.
42. 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.
43. Sánchez Cuadrado, J., Cánovas, J., García Molina, J. Applying model driven engineering in small software enterprises. *Science of Computer Programming* 89, 176-198, 2014.
44. Selim, G., Cordi, J., Dingel, J. How is ATL Really Used? Language Feature Use in the ATL Zoo. 2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems, IEEE, 2017.
45. SMM. Software Metrics Meta-Model, Version 1.0, OMG Document Number: formal/2012-01-05. <http://www.omg.org/spec/SMM/1.1>, 2012.
46. 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.
47. Stankovic, J. Research Directions for the Internet of Things. *IEEE Internet of Things Journal*, 1(1), 3–9, 2014.
48. Tonella, P., Potrich, A. Reverse Engineering of Object-Oriented Code. *Monographs in Computer Science*. Heidelberg: Springer-Verlag, 2005.
49. Ulrich, W., NewComb, P. Information System Transformation. *Architecture Driven Modernization Case Studies*. MK/OMG Press. Elsevier, 2010.
50. 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.
51. Ulrich, W.M., Newcomb, P. *Information Systems Transformation: Architecture Driven Modernization Case Studies*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2010.
52. 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.
53. Vaupel, S., Taentzer, G., Harries, J.P., Stroh, R., Gerlach, R., Guckert, M. Model-Driven Development of Mobile Applications Allowing Role-Driven Variants. *Lecture Notes in Computer Science* 8767, Springer-Verlag, 2014.
54. 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.
55. Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M.. Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, 1(1), 22–32. doi:10.1109/JIOT.2014.2306328, 2014.

56. Zodik, G. Future technologies supporting the convergence of Mobile, wearables , and IoT. 2015 2nd ACM International Conference on Mobile Software Engineering and Systems, IEEE, 2015.