

## **Plan de Trabajo Final Carrera Ingeniería de Sistemas**

**Facultad de Ciencias Exactas – UNICEN**

**Tema:**

**Migración de software C/C++ a plataformas móviles a partir de MDD (*Model Driven Development*)**

**Alumnos:**

- **Maximiliano Pablo Duthey (LU: 247684)**
- **Carolina María Spina (LU: 247769)**

**Directora:**

**Lic. Liliana Favre**

## 1. Introducción

Actualmente los dispositivos móviles acompañan a los usuarios en todo momento y lugar, y se prevé que serán el principal medio de acceso a Internet en los próximos años, sin embargo, el desarrollo de aplicaciones de software móviles no está lo suficientemente maduro [34].

La proliferación de diferentes plataformas móviles ha forzado a los desarrolladores a definir enfoques que permitan simplificar el desarrollo de aplicaciones [21]. Los autores remarcan en [11] que dos de los principales desafíos de la ingeniería de software de aplicaciones móviles son por un lado, la creación de interfaces de usuario que abarquen diferentes clases de dispositivos móviles y por otro, brindar aplicaciones reutilizables en múltiples plataformas.

Desarrollar una aplicación de software para un dispositivo móvil implica adoptar y entender las características de estos dispositivos y sus restricciones. Si bien éstos cuentan con características avanzadas también se presentan importantes restricciones. Por ejemplo, incorporan interfaces de entrada más intuitivas, usualmente, pantallas táctiles, bases de datos integradas, soporte multimedia y mecanismos de comunicación y geolocalización. También se presentan importantes restricciones en cuanto al tamaño de la pantalla disponible, capacidad de procesamiento, la utilización de memoria primaria y las bibliotecas de desarrollo disponibles [35]. En [21] se presenta un análisis de los conceptos claves de las plataformas Android, iPhone y QT. En [34] se analizan problemas abiertos en la ingeniería de software de aplicaciones móviles remarcando que hoy en día hay un “ángulo” móvil para todo desarrollo de software.

Un problema actual es la rápida proliferación de plataformas móviles [21]. El alto costo y la complejidad técnica de desplegar una aplicación a un amplio espectro de plataformas, fuerza a desarrollar aplicaciones alineadas a cada tipo de dispositivo. Varias empresas tienen grupos de desarrollo para cada tipo de plataforma redoblando esfuerzos para aplicaciones móviles de funcionalidad similar. En aplicaciones multiplataforma, los desarrolladores prefieren implementar una aplicación por vez, y desplegarla en diversas plataformas con un mínimo esfuerzo. En esta dirección, surgió el proyecto HAXE, que es un lenguaje de programación de propósitos generales diseñado para que los desarrolladores puedan, utilizando un solo lenguaje y un conjunto de librerías, abarcar distintas plataformas de manera eficaz. Actualmente, el compilador de HAXE genera código para las siguientes tecnologías: JavaScript, Flash, NekoVM, PHP, C++ y C# [8] [10].

Otro desafío en estos desarrollos es lograr integrar variedad de información e interoperabilidad de herramientas. Model Driven Development (MDD) es considerado un enfoque promisorio para afrontarlo [5]. 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*) [28].

MDA propone separar la especificación de la funcionalidad del sistema de su implementación sobre una plataforma en una tecnología específica y controlar la evolución del software desde modelos abstractos a implementaciones tendiendo a aumentar el grado de automatización. Una de las características esenciales de MDA es que todos los artefactos involucrados en un proceso de desarrollo se representan a partir del lenguaje de metamodelado MOF (*Meta Object Facility*) [24]. Otro concepto fundamental es el de transformaciones entre modelos. En MDA, una transformación es la

especificación de mecanismos para convertir elementos de un modelo en elementos de otro modelo y el estándar propuesto por OMG para especificar transformaciones es el meta modelo QVT (*Query, View, Transformation*) [31].

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*) [2] que lleva a cabo procesos de modernización considerando los principios esenciales de MDA. ADM define estándares específicos para modernización. 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 [23]. Como todo componente Eclipse, MoDisco puede integrarse con *plugins* o tecnologías disponibles en el ambiente Eclipse. El proceso de modernización incluye tres etapas: ingeniería inversa, reestructuración e ingeniería *forward*. La ingeniería inversa, el proceso de analizar los artefactos de software existentes para extraer información proveer vistas de alto nivel del sistema, es una etapa crucial dentro de la modernización del sistema. En el contexto de ADM, la meta de la ingeniería inversa es descubrir el conocimiento del sistema existente y producir modelos en diferentes niveles de abstracción. Estos modelos serán el punto de partida para el proceso de reestructuración y la posterior generación del nuevo código.

Se propone en esta tesis trabajar sobre un escenario de modernización: la migración de código C/C++ a plataformas móviles a través del lenguaje HAXE.

## 2. Motivación

Es frecuente la necesidad de migrar componentes y aplicaciones de software desarrollados en C/C++ a plataformas móviles y por ende es conveniente contar con procesos genéricos reutilizables que la soporten. Esto puede lograrse a partir de procesos dirigidos por modelos integrados con el lenguaje multiplataforma HAXE.

En trabajos existentes no se ha encontrado una integración de C/C++ con HAXE a partir de MDA y consideramos de interés definir procesos de migración que los integren teniendo en cuenta la cantidad de software de calidad desarrollado en estos lenguajes que los programadores de aplicaciones móviles deben adaptar. Los procesos deberían incluir el análisis de código existente para extraer modelos y la generación de código a partir de estos modelos para múltiples plataformas móviles.

En MDD, las transformaciones entre modelos se expresan en lenguajes específicos que permiten a los desarrolladores concentrarse en los aspectos conceptuales de las relaciones entre modelos y delegar el proceso de transformación. La generación de modelos a partir de transformaciones entre metamodelos en MDD apunta a generar modelos “correctos por construcción” con respecto a la especificación de metamodelos.

Este enfoque requiere contar con un metamodelo MOF de C/C++, produciendo además el beneficio de otorgar a los desarrolladores MDD este metamodelo y así la posibilidad de abordar otros escenarios de modernización que involucren al lenguaje C/C++.

### 3. Objetivos

Las herramientas CASE basadas en UML o en MDA no soportan procesos de ingeniería forward para HAXE [8] [10] y, por lo tanto, de ingeniería inversa. En esta propuesta se considera valioso tender un puente entre C/C++ y HAXE y enfoques MDD, MDA en particular.

El primer paso en esta dirección es contar con una definición de C/C++ alineada con los estándares MDA, en particular MOF. Se propone como objetivo definir un metamodelo MOF para C/C++ que permita especificar transformaciones a nivel de metamodelos usando el lenguaje de transformaciones ATL [15].

Luego, se propone definir procesos de ingeniería inversa desde C/C++ a modelos y posteriormente la generación de código HAXE a partir de estos modelos.

La propuesta será validada en Eclipse. Se desarrollará como caso de estudio la adaptación a plataformas móviles de diferentes piezas de software escrito en C/C++.

#### 3.1 Métodos y técnicas a emplear

La validación de los procesos de modernización se realizará bajo el proyecto de código abierto Eclipse dado que algunos de sus subproyectos proporcionan herramientas y entornos de ejecución alineados con estándares de MDD, en particular MDA y ADM. El subproyecto EMF (*Eclipse Modeling Framework*) es un framework de modelado que provee infraestructura para el desarrollo de herramientas tales como editores, motores de transformación y soporte para metamodelado [14]. 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 [22]. Dentro del ámbito de Eclipse también se han creado motores de transformación como ATL [15], *QVT Operational* y se avanza en la implementación de *QVT Declarative*, aún en fase de incubación. Acceleo es la implementación en Eclipse de transformaciones modelo-a texto basada en Ecore MOF M2T [1][25].

Para la extracción de modelos se usará el generador de parser ANTLR (ANother Tool for Language Recognition) y se usará MoDisco [29] [23]. Específicamente, de los posibles escenarios de modernización se analizará la migración de software C/C++ a plataformas móviles a través del lenguaje HAXE. HAXE fue diseñado para que los desarrolladores puedan, por medio de un solo lenguaje y un conjunto de librerías, abarcar varias plataformas, y en particular las utilizadas en plataformas móviles, de manera eficaz.

#### **4. Cronograma de actividades**

Se proponen las siguientes etapas para concretar este plan:

*1. Estado actual del conocimiento y búsqueda bibliográfica.*

*1.1. Análisis de estándares de metamodelado vinculados a MDA, técnicas de metamodelado y herramientas de soporte para metamodelado.*

*1.2. Análisis de técnicas de ingeniería inversa basadas en análisis estático y técnicas de modernización basadas en MDA y ADM.*

*2. Análisis de herramientas para el desarrollo: el framework ANTLR para generar analizadores léxico y sintáctico, el lenguaje de transformaciones ATL, el lenguaje HAXE y su metamodelo, MoDisco y Acceleo*

*3. Definición del metamodelo Ecore del lenguaje C/C++*

*4. Análisis de procesos basados en MDA y ADM para la modernización de software orientado a objetos. Definir un proceso para la modernización de código orientado a objetos a modelos.*

*5. Definición de un proceso de ingeniería directa que tienda un puente hacia HAXE y diversas plataformas móviles a partir de un enfoque MDD.*

*6. Desarrollo de un caso de estudio que valida la propuesta.*

*7. Redacción del informe final*

##### **4.1 Grado de avance**

Esta propuesta de tesis surge a partir de trabajos realizados por los alumnos como becarios de entrenamiento CIC en el período 2014-2015. El tema de la beca de entrenamiento de Maximiliano Pablo Duthey es "Modernización de software" y el tema de Carolina María Spina es "Desarrollo de software dirigido por modelos y aplicaciones móviles". Se propone en este plan integrar los resultados logrados a partir de las investigaciones previas realizadas en el marco de estas becas en una propuesta de un proceso de migración de software C/ C++ a plataformas móviles.

Se cuenta con un significativo avance en las etapas 1, 2, 3 y 4 previéndose concretar el resto de las actividades en el transcurso de los próximos 5 meses.

## 5. Bibliografía

- [1] Acceleo (2015). Obeo. Acceleo Generator. <http://www.eclipse.org/acceleo/>
- [2] ADM (2015). Architecture-driven modernization task force. <http://www.adm.org>
- [3] ASTM (2011). OMG Architecture-driven Modernization: Abstract Syntax Tree Metamodel (ASTM), versión 1.0, <http://www.omg.org/spec/ASTM>.
- [4] Bowen, J., Hinze, A. (2011). Supporting mobile application development with model-driven emulation. *Journal of the ECEASST*, Volumen 45 (pp. 1-5).
- [5] Brambilla, M. Cabot, J., Wimmer, M (2012). *Model-Driven Software Engineering in Practice*, Synthesis Lectures on Software Engineering. Morgan & Claypool Publishers.
- [6] Braun, P., Eckhaus, R. (2008). 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), Los Alamitos: IEEE Computer Society.
- [7] Bruneliere, H. Cabot, J. Dupé, G., Madiot, F. (2014). MoDisco: a Model Driven Reverse Engineering Framework. *Information and Software Technology* 56 (pp. 1012-1032).
- [8] Cannasse, N. (2014). HaXe. Too Good to be True? GameDuell Tech Talk. <http://www.techtalk-berlin.de/news/read/nicolas-cannasse-introducing-haxe/>
- [9] CASE MDA. (2015). Committed companies and their products. [www.omg.org/mda/committed-products.htm](http://www.omg.org/mda/committed-products.htm)
- [10] Dasnois, B. (2011). *HaXe 2 Beginner's Guide*. Packt Publishing.
- [11] Dehlinger, J., Dixon, J. (2011). Mobile application software engineering: Challenges and research directions. *Proceedings of the Workshop on Mobile Software Engineering* (pp. 29-32), Berlin Heidelberg: Springer-Verlag.
- [12] Díaz Bilotto, Pablo; Favre, Liliana (en prensa) Migrating JAVA to Mobile Platforms through HAXE: An MDD Approach," Aceptado como capítulo del libro "Modern Software Engineering Methodologies for Mobile and Cloud Environments". Editores: Antonio Miguel Cruz , Sara Paiva. IGI GLOBAL, USA. 2016
- [13] Dunkel, J., Bruns, R. (2007). Model-driven architecture for mobile applications. In *Business Information Systems, W. Abramowicz, Ed., Lecture Notes in Computer Science*, Volumen 4439 (pp. 464-477), Berlin Heidelberg: Springer-Verlag.
- [14] EMF (2015) Eclipse Modeling Framework (EMF) Retrieved April 15, 2015 from <http://www.eclipse.org/modeling/emf/>
- [15] Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I (2008). ATL: A model transformation tool. *Science of Computer Programming* 72, 1 (pp. 31-39).
- [16] Jouault, F., Allilaire, F., Bézivin, J., Kurtev, I., Valduriez, P: (2006) ATL: a QVT-like transformation language. Companion to the 21st ACM SIGPLAN Symposium on Object-oriented programming systems, languages, and applications (New York, NY, USA, 2006), OOPSLA '06 (pp. 719-720), New York: ACM Press.
- [17] Jouault, F., Kurtev, I. (2005). Transforming models with ATL. In *Satellite Events at the MoDELS 2005 Conference (Berlin, 2006)*, Lecture Notes in Computer Science, vol. 3844, Berlin Heidelberg: Springer Verlag (pp. 128-138).

- [18]KDM. OMG (2011). Architecture-Driven Modernization: Knowledge Discovery MetaModel (KDM), versión 1.3, <http://www.omg.org/>
- [19] Kim, H. K. (2008). Frameworks of process improvement for mobile applications. *Engineering Letters* 16, 4 (pp. 550-555).
- [20] Kramer, D., Clark, T., Oussena, S. (2010). MobDSL: A domain specific language for multiple mobile platform deployment. *Networked Embedded Systems for Enterprise Applications (NESEA)*, 2010 IEEE International Conference (pp. 1-7), Los Alamitos: IEEE Press.
- [21] Lettner, M., Tschernuth, M., Mayrhofer, R. (2012). Mobile platform architecture review: Android, Iphone, Qt. In *Computer Aided Systems Theory EUROCAST 2011*, R. Moreno-Díaz, F. Pichler, and A. Quesada-Arencibia, *Lecture Notes in Computer Science*, Volumen 6928 (pp. 544-551). Berlin Heidelberg: Springer-Verlag.
- [22] MMT (2015). Model-to-Model Transformation. Eclipse Modeling Framework. Retrieved April 15, 2015, <https://www.eclipse.org/mmt/>
- [23] MoDisco. (2012). Model discovery. Retrieved April 15, 2015 <http://www.eclipse.org/MoDisco>
- [24] MOF (2011). OMG Meta Object Facility (MOF) core specification version 2.4.1. OMG Document Number:formal/2011-08-07. <http://www.omg.org/spec/MOF/2.4.1>
- [25] MOFM2T (2008). MOF Model to Text Transformation Language, Version 1.0. <http://www.omg.org/spec/MOFM2T/1.0/>
- [26] OCL (2014). OMG Object constraint language (OCL), version 2.4. <http://www.omg.org/spec/OCL/2.4>
- [27] OCLinEcore (2015) OCLinEcore Editor <http://wiki.eclipse.org/MDT/OCLinEcore>.
- [28] OMG MDA (2014). MDA guide version rev. 2.0 OMG Document ormsc/2014-06-01. <http://www.omg.org/cgi-bin/doc?ormsc/14-06-01>
- [29] Parr, T. (2013) *The Definitive ANTLR 4 Reference*. Pragmatic Bokkshelf, Second Edition,
- [30] Pérez Castillo, R., García Rodríguez, I., Gómez Cornejo, R., Fernández Roper, M., Piattini, M. (2013). ANDRIU. A Technique for Migrating Graphical User Interfaces to Android. *Proceedings of The 25th International Conference on Software Engineering and Knowledge Engineering (SEKE 2013)* (pp. 516-519) Boston: Knowledge Systems Institute.
- [31] QVT (2012). QVT: MOF 2.0 query, view, transformation:Version 1.1. OMG Document Number: formal/2011-01-01. <http://www.omg.org/spec/QVT/1.1/SMM>
- [32] Steinberg, D., Budinsky, F., Paternostro, M., Merks, E. (2009). *EMF: Eclipse Modeling Framework*, 2 ed. Addison-Wesley, Boston, MA.
- [33] Thompson, C., Schmidt, D., Turner, H., White, J. (2011). Analyzing Mobile Application Software Power Consumption via Model-Driven Engineering, *Proceedings of PECCS 2011* (pp. 101-113)
- [34] Wasserman, A. I. (2010). Software engineering issues for mobile application development. *Proceedings of the FSE/SDP workshop on Future of software engineering research (FoSER '10)*, (pp. 397-400) New York, NY: ACM.

Avalo la presente solicitud de evaluación

Liliana Favre

Firma de alumnos:

Maximiliano Pablo Duthey

Carolina María Spina