

INFORME BECA DE ENTRENAMIENTO CIC

Becaria: Carolina María Spina

Directora: Lic. Liliana Favre

Período: 1/11/2014 al 31/10/2015

1. Plan de trabajo de la Beca de Entrenamiento

Tema: Desarrollo de software dirigido por modelos y aplicaciones móviles

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.

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 como 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, la utilización de memoria primaria y las bibliotecas de desarrollo disponibles.

La proliferación de diferentes plataformas móviles ha forzado a los desarrolladores a definir enfoques que permitan simplificar el desarrollo de aplicaciones. Un 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[1]. 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*) (<http://www.omg.org/mda>).

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*) (<http://www.omg.org/spec/MOF/2.4.1>). 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) (<http://www.omg.org/spec/QVT/1.1/>).

Un problema actual es el desarrollo de una aplicación para múltiples plataformas móviles. En esta dirección, surgió HAXE, un lenguaje de programación open source multiplataforma que se integra con diferentes plataformas (<http://haxe.org>) que evoluciona

con el aporte de la comunidad de usuarios a quienes brinda la posibilidad de generar aplicaciones confiables para diversas plataformas móviles.

Se propone en esta propuesta tender un puente entre HAXE y enfoques MDD, MDA en particular a fin de lograr desarrollo de aplicaciones para múltiples plataformas móviles. Para lograr los objetivos propuestos se propone una combinación de técnicas básicas de MDA como metamodelado y transformación de metamodelos con el lenguaje HAXE. Se validará la propuesta bajo el proyecto Eclipse dado que algunos de sus subproyectos proporcionan herramientas y entornos de ejecución alineados con MDA. Se propone como caso de estudio la adaptación a diferentes plataformas móviles de GIMP (GNU Image Manipulation Program) (www.gimp.org). Se eligió esta aplicación dada la escasez de herramientas libres de diseño gráfico para dispositivos móviles

Referencia:

[1] Brambilla, M.; Cabot, J.; Wimmer, M. Model-Driven Software Engineering in Practice. Morgan & Claypool Publishers (2012)

2. Breve exposición de la tarea realizada

Las actividades de investigación se desarrollaron en el marco del proyecto “Tecnologías de software” (INTIA-UNCPBA). La investigación realizada se centró en la definición de procesos de la migración de software orientado a objetos a plataformas móviles a partir de la integración de desarrollos dirigidos por modelos (MDD- *Model Driven Development*) y el lenguaje multiplataforma HAXE [2] [3].

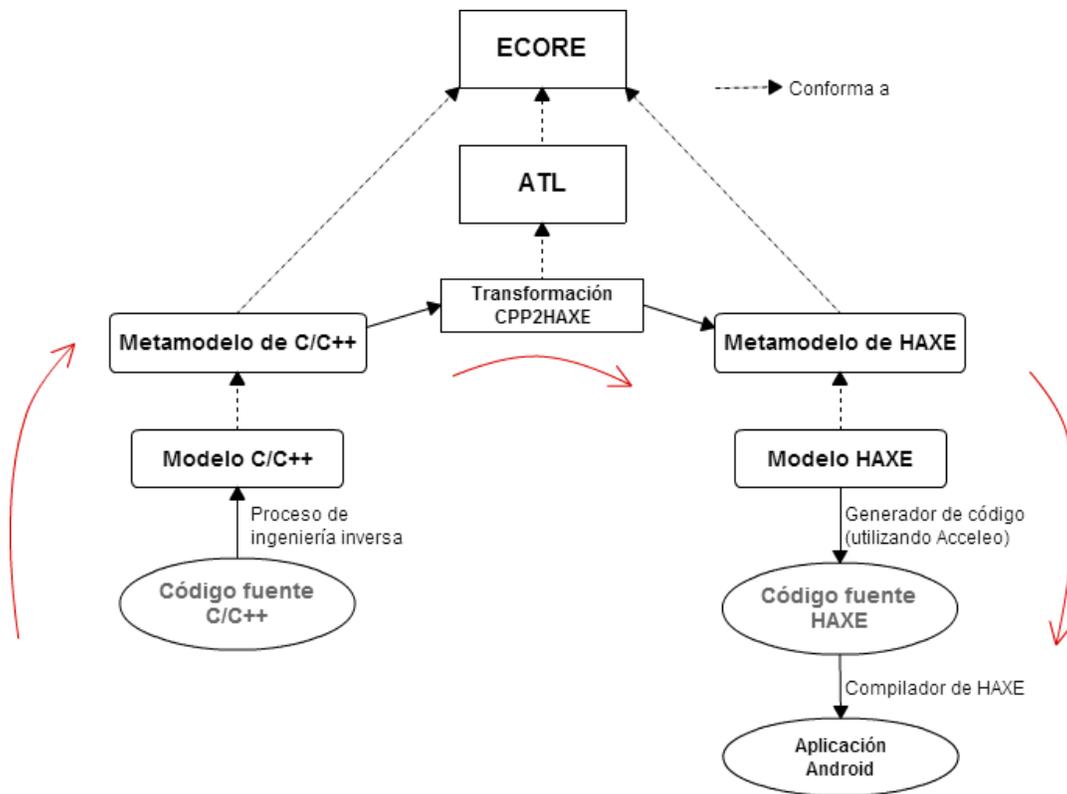
La proliferación de diferentes plataformas móviles ha forzado a los desarrolladores a definir enfoques que permitan simplificar el desarrollo de aplicaciones. Un desafío en estos desarrollos es lograr integrar variedad de información e interoperabilidad de herramientas [4]. Un enfoque promisorio para afrontar estos desafíos son los desarrollos dirigidos por modelos (MDD) [2]. Luego, una de las primeras actividades realizadas fue analizar las bases de MDD, el cual 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*) [6]. 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). El lenguaje de transformación de modelos ATL (*Atlas Transformation Language*) es el más usado debido a su grado de madurez [5].

Posteriormente se experimentó con el lenguaje HAXE y se analizó y refinó su metamodelo Ecore. HAXE es un lenguaje orientado a objetos, aunque con características funcionales y fuertemente tipado. Permite, empleando las abstracciones correctas mantener un único código base que puede ser compilado para diferentes plataformas móviles. Se ha desarrollado un metamodelo Ecore de HAXE en su versión 3.1.3. que permite su integración a procesos basados en MDA. Desde la óptica de MDA, las instancias del metamodelo corresponden a un modelo específico de la plataforma.

Otra de las actividades cruciales fue la definición de un metamodelo Ecore de C/C++. Este metamodelo permitió definir transformaciones modelo-a-modelo (M2M) entre C/C++ y Haxe.

Se analizaron, además procesos de ingeniería directa a partir de transformaciones Modelo-a-Texto (M2T). A partir de las bases previas se definió un proceso para la migración de software C/C++ a plataformas móviles a través del lenguaje HAXE. La siguiente figura resume las etapas del proceso propuesto:



1. Especificar un metamodelo del lenguaje origen (C/C++).
2. Especificar un metamodelo del lenguaje destino (HAXE).

3. Ingeniería inversa de código C/C++ para creación del modelo que conforme al metamodelo correspondiente.
4. Transformación del modelo del lenguaje origen (C/C++) a un modelo que conforme al metamodelo del lenguaje destino (HAXE).
5. Realizar ingeniería directa desde el modelo del lenguaje destino (HAXE) a texto.
6. Compilación del código fuente a plataformas móviles.

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

Se resumen las actividades realizadas para concretar este plan:

1. *Estudio del estado actual del conocimiento y búsqueda bibliográfica.*

Análisis de estándares de metamodelado vinculados a MDA, técnicas de metamodelado y herramientas de soporte para metamodelado e ingeniería directa.

2. *Experimentación con el lenguaje HAXE*

3. *Análisis y refinamiento del metamodelo Ecore del lenguaje HAXE*

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

5. *Análisis de procesos para la ingeniería directa, transformaciones Modelo-a-Modelo (M2M) y transformaciones Modelo-a-Texto (M2T)*

6. *Transformación del modelo del lenguaje C/C++ a un modelo que conforme al metamodelo HAXE*

7. *Definición de un proceso de ingeniería directa desde el lenguaje HAXE a Texto para la posterior compilación del código fuente a plataformas móviles.*

8. *Desarrollo de casos de estudio*

Los resultados logrados en este trabajo se integraron con los resultados logrados en otra beca de entrenamiento CIC desarrollada en el mismo período por Maximiliano Duthey en un Trabajo Final de la carrera de Ingeniería de Sistemas denominado "Migración de software C/C++ a plataformas móviles a partir de MDD (*Model Driven Development*)". Se adjunta en 3.1. el Plan de Trabajo Final aceptado por el Departamento de Computación y Sistemas (FCEX, UNCPBA). Además, en 3.2. se adjunta un informe preliminar del Trabajo Final que detalla mi participación en las distintas etapas.

Referencias:

- [1] Acceleo (2015). *Obeo. Acceleo Generator*. Retrieved April 15, 2015, from <http://www.eclipse.org/acceleo/>
- [2] Brambilla, M.; Cabot, J.; Wimmer, M. (2012) *Model-Driven Software Engineering in Practice*. Morgan & Claypool Publishers
- [3] Dasnois, B. (2011). *HaXe 2 Beginner's Guide*. Packt Publishing.
- [4] Dehlinger, J., & Dixon, J. (2011). Mobile application software engineering: Challenges and research directions. In *Proceedings of the Workshop on Mobile Software Engineering* (pp. 29-32), Berlin Heidelberg: Springer-Verlag.
- [5] Jouault, F., Allilaire, F., Bézivin, J., & Kurtev, I (2008). ATL: A model transformation tool. *Science of Computer Programming* 72, 1 (pp. 31-39).
- [6] 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>

3. Documentación complementaria

- 3.1. Plan de Trabajo Final aceptado por el Departamento de Computación y Sistemas. Facultad de Ciencias Exactas. UNCPBA
- 3.2. Informe preliminar del Trabajo Final

Avalo el presente informe

Liliana Favre- Directora Beca de Entrenamiento CIC

Carolina María Spina - Becaria de Entrenamiento CIC