

INFORME BECA DE ENTRENAMIENTO CIC

Becario: Maximiliano Duthey

Directora: Lic. Liliana Favre

Período: 1/11/2014 al 31/10/2015

1. Plan de trabajo de la Beca de Entrenamiento

Modernización de software

Existe actualmente en la industria del software una creciente demanda de soporte para la modernización de sistemas a fin de lograr ventajas competitivas para las organizaciones. Nuevos enfoques de desarrollo de software dirigidos por modelos como la arquitectura MDA (Model Driven Architecture) (<http://www.omg.org/mda>) podrían dar respuesta a esta demanda. MDA no es en sí misma una especificación de una tecnología sino un plan para lograr especificaciones de tecnologías cohesivas y centradas en modelos que tiendan a aumentar el grado de automatización de procesos de desarrollo de software [1]. MDA distingue los siguientes modelos: modelo independiente de la computación (Computation Independent Model o CIM), modelo independiente de la plataforma (Platform Independent Model o PIM), modelo específico a la plataforma (Platform Specific Model o PSM) y modelo específico a la implementación. 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 en MDA es la especificación de mecanismos para transformar elementos de un modelo en elementos de otro modelo. El estándar propuesto por OMG para especificar transformaciones es el metamodelo QVT (Query, View, Transformation) (<http://www.omg.org/spec/QVT/1.1/>). Otro motor de transformaciones ampliamente usado y alineado a MDA es ATL (Atlas Transformation Language) (<http://www.eclipse.org/atl/documentation/>).

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 entre los que se encuentran KDM (Knowledge Discovery Metamodel) (<http://www.omg.org/spec/kdm>) y ASTM (Abstract Syntax Tree Metamodel) (<http://www.omg.org/spec/ASTM>). Estos estándares constituyen dos especificaciones de modelos complementarias, KDM define una especificación que permite representar información semántica acerca de un sistema de software, mientras que ASTM establece una especificación para representar la sintaxis del código fuente por medio de árboles de sintaxis abstracta. ASTM actúa como la base de bajo nivel para modelar software dentro del ecosistema de estándares del OMG, mientras que KDM sirve como puerta de entrada a los modelos de más alto nivel. MoDisco (Modeling Discovery) (<http://www.eclipse.org/Modisco>) 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.

Se propone como objetivo general de este trabajo analizar procesos basados en MDA y ADM para la modernización de software orientado a objetos. En particular, se propone analizar procesos de que integren técnicas tradicionales de análisis estático y dinámico con los estándares ADM. La propuesta se validará a través de la modernización de código orientado a objetos a modelos MDA que permitan posteriormente procesos de ingeniería directa (*forward engineering*).

Referencias:

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

[2] ADM 2014. Architecture-Driven Modernization Task Force.
<http://www.omgwiki.org/admtf/doku.php>

2. Breve exposición de la tarea realizada

Las actividades de investigación se desarrollaron en el marco del proyecto “Tecnologías de Software” perteneciente al Instituto INTIA de la Facultad de Ciencias Exactas en la UNCPBA. Se propuso como objetivo general de esta Beca de Entrenamiento analizar procesos de modernización de software basados en técnicas tradicionales de análisis estático y dinámico y técnicas relacionadas a los estándares definidos por OMG (*Object Management Group*) para MDA (*Model Driven Architecture*) y ADM (*Architecture Driven Modernization*) [3] [1] [2].

Con respecto a las técnicas tradicionales de análisis estático y dinámico se analizaron diferentes propuestas y en particular el enfoque propuesto en [4]. Los autores presentan una visión general de las técnicas que han sido recientemente investigadas y aplicadas en el campo de la ingeniería inversa de código orientado a objetos describiendo los algoritmos utilizados para la recuperación de diagramas UML a partir de código, tales como diagramas de clases, de estado y de secuencia.

Por otra parte, se analizaron estándares vinculados a desarrollos dirigidos por modelos (*Model Driven Development* o MDD) [2]. MDD es considerado un enfoque promisorio para afrontar la modernización de software orientado a objetos. Una realización específica de MDD, propuesta por OMG es la arquitectura MDA (*Model Driven Architecture*). 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*). 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*).

Como una evolución de los principios esenciales de MDA, OMG está involucrado en la especificación desarrollo de estándares para la modernización de sistemas legados a través de la iniciativa ADM (*Architecture Driven Modernization*) que lleva a cabo procesos de modernización considerando los principios esenciales de MDA. Entre los estándares esenciales que se analizaron se destacan los metamodelos KDM (*Knowledge Discovery Metamodel*) y ASTM (*Abstract Syntax Tree Metamodel*). KDM define un metamodelo para el descubrimiento de conocimiento de software independiente de cualquier lenguaje de implementación y plataforma de ejecución. Su meta principal es la de proporcionar una forma común para el intercambio de metadatos de aplicaciones posibilitando de esta forma interoperabilidad entre herramientas. ASTM establece una especificación para representar árboles de sintaxis abstracta, es decir proveer una definición de elementos utilizados para componer modelos de árboles de sintaxis abstracta.

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 y 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 sistema. En esta beca se puso el énfasis en la etapa de ingeniería inversa.

Para lograr los objetivos se propuso una combinación de técnicas básicas de análisis estático con técnicas basadas en MDA como metamodelado y transformación de metamodelos. 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 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. EMF incluye el metamodelo Ecore, el cual es una implementación del estándar MOF. Otro subproyecto Eclipse es MMT (*Model-to-Model Transformation*), originalmente conocido como M2M, que soporta transformaciones entre modelos. Dentro del ámbito de Eclipse también se han creado motores de transformación como ATL (*Atlas Transformation Language*), *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. 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. Como todo componente Eclipse, MoDisco puede integrarse con *plugins* o tecnologías disponibles en el ambiente Eclipse. Para la extracción de modelos se usó el generador de parser ANTLR (*ANother Tool for Language Recognition*).

Se resumen las actividades realizadas para concretar este plan:

1. Estudio del 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 y el lenguaje de transformaciones ATL.*

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

4. *Análisis de procesos para la ingeniería inversa de código orientado a objetos basados en técnicas tradicionales de análisis estático y dinámico y técnicas basadas en MDA y ADM..*

5. *Definición de un proceso para la ingeniería inversa que, partiendo de código C/C++ extraiga modelos que conformen al metamodelo Ecore del lenguaje C++*

6.. *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 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 3.1 el plan presentado. Además se adjunta un informe preliminar del Trabajo Final que detalla mi participación en las distintas etapas en 3.2.

Referencias:

[1] ADM 2015. Architecture-Driven Modernization Task Force.

<http://www.omgwiki.org/admtf/doku.php>

[2] Brambilla, M.; Cabot, J.; Wimmer, M. Model-Driven Software Engineering in Practice.

Morgan & Claypool Publishers (2012)

[3] 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>

[4] Tonella, P., Potrich, A., 2005. *Reverse Engineering of object-oriented Code.*

Monographs in Computer Science. Heidelberg: Springer-Verlag.

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

Maximiliano Pablo Duthey - Becario de Entrenamiento CIC