

# Generación Automática de código RMI a partir del Modelo de Análisis utilizando Reglas Relations/QVT

Ariel Arsaute, Marcela Daniele, Mariana Frutos, Ariel Gonzalez, Marcelo Uva, Fabio Zorzan.

Departamento de Computación, Facultad de Ciencias Exactas, Físico-Químicas y Naturales,  
Universidad Nacional de Río Cuarto.

Ruta 36 Km. 601 –CP 5800 - Río Cuarto – Córdoba - Argentina Tel. (0358) 4676235  
{aarsaute, marcela, mfrutos, agonzalez, uva, fzorzan}@dc.exa.unrc.edu.ar

## Resumen

Este trabajo realiza una contribución tendiente a la mejora del proceso de desarrollo de software, siguiendo la filosofía de Model Driven Architecture (MDA). Dentro de esta línea de investigación en trabajos previos [1,2,10] se definió la transformación del modelo de CU de sistema al modelo de análisis y del modelo de análisis al modelo de diseño considerando la tecnología de implementación Remote Method Invocation (RMI). Para lograr estas transformaciones se especificaron una serie de reglas de transformación Query/View/Transformation (QVT) que permitieron establecer la correspondencia entre los modelos fuentes y destinos correspondientes. Los modelos de análisis y diseño obtenidos como resultado de aplicar las distintas transformaciones QVT (modelos objetivos), atañen a modelos UML correspondiente a los diagrama de clases de análisis y diseño respectivamente.

En este trabajo se pretende la generación automática de código, tomando como entrada el modelo de diseño resultado y haciendo uso de la herramienta Aceleo que es un generador de código que transforma los modelos en código.

El objetivo final es la realización de aportes tendientes a lograr un producto de software siguiendo los principios de abstracción, automatización y estandarización establecidos por MDA, de manera totalmente automática, o bien automatizando el proceso de transformación lo mayormente posible.

**Palabras clave:** MDA, QVT, Proceso Unificado, UML.

## Contexto

La línea de investigación presentada en este trabajo se desarrolla en el marco del proyecto “Ingeniería de Software: Automatización de Procesos de Desarrollo de Software”, perteneciente a los Proyectos y Programas de Investigación (PPI) de la secretaría de Ciencia y Técnica de la Universidad Nacional de Río Cuarto.

## Introducción

La ingeniería de Software implementa un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento del software. Las metodologías de desarrollo junto con las herramientas de software se han adoptado con éxito en un amplio espectro de aplicaciones industriales. Dentro de las metodologías orientadas a objetos, el Proceso Unificado [3] define actividades y responsabilidades estableciendo quién está haciendo qué, cuándo, y cómo para construir o mejorar un producto de software. Esta metodología, divide el desarrollo del producto de software en fases utilizando UML (Unified Modeling Language) como lenguaje de modelado [4].

### ARQUITECTURA DIRIGIDA POR MODELOS

La Arquitectura Dirigida por Modelos (MDA) [5] se ha concebido para dar soporte a la ingeniería dirigida por modelos de los sistemas software, cuyo objetivo central es resolver el problema producido por el cambio de tecnologías junto con la integración del sistema de software y que todo esto no impliquen un alto costo.

La idea subyacente en MDA es usar modelos, de modo que las propiedades y características de los sistemas queden contenidas en un modelo abstracto independiente de los cambios producidos en la tecnología. MDA proporciona una serie de guías o patrones expresadas como modelos. MDA propone cuatro niveles de abstracción que componen la jerarquía o arquitectura de modelos. Estos son: Modelo independiente de cómputo CIM (Computation Independent Model), Modelo independiente de la plataforma PIM (Platform Independent Model), Modelos específicos de la plataforma PSM (Platform Specific Model), y la aplicación final.

Los modelos CIM describen el entorno en el que se utilizará el sistema, sin referencia directa a su implementación.

Los PIM modelan funcionalidad y estructura de un sistema de información sin considerar detalles tecnológicos de la plataforma en la cual se implementará el sistema.

Los PSM describen los modelos específicos de plataforma, concretamente de la plataforma tecnológica donde se ejecutará el sistema.

MDA plantea el siguiente proceso de desarrollo: de los requisitos se obtiene un modelo independiente de la plataforma (PIM), luego este modelo es transformado con la ayuda de herramientas en uno o más modelos específicos de la plataforma (PSM), y finalmente cada PSM es transformado en código. Por lo tanto, MDA incorpora la idea de transformación de modelos (PIM a PSM, PSM a código) y se necesitan herramientas para automatizar estas tareas. Estas herramientas constituyen uno de los elementos básicos de MDA.

### QVT (Query / View / Transformation)

El planteamiento QVT se basa principalmente en la definición de un lenguaje para las consultas (Queries) sobre los modelos Meta Object Facility (MOF) [6], la búsqueda de un estándar para generar vistas (Views) que revelen aspectos específicos de los sistemas modelados, y finalmente, la definición de un lenguaje para la descripción de transformaciones de modelos MOF.

En este trabajo se plantea la utilización de QVT para definir transformaciones entre modelos. Estas transformaciones describen relaciones entre un metamodelo fuente F y un metamodelo objetivo O, ambos metamodelos deben estar especificados en MOF. Luego esta transformación se utiliza para obtener un modelo objetivo, el cual es una instancia del metamodelo O, a partir de un modelo fuente que es una instancia del metamodelo F. Una característica muy importante de estas transformaciones es que pueden ser bidireccionales y multidimensionales.

La herramienta CASE utilizada para la definición de las reglas QVT es MediniQVT [7]. Esta herramienta implementa la especificación QVT/Relations de la OMG en un poderoso motor QVT. MediniQVT está diseñada para transformaciones de modelos permitiendo un rápido desarrollo, mantenimiento y particularización de reglas de transformación de procesos específicos. La herramienta está integrada a Eclipse y utiliza Eclipse Modeling Framework (EMF) [8] para la representación de modelos.

### Líneas de investigación y desarrollo

Como se mencionó anteriormente, MDA propone basar el desarrollo de software en modelos, separando el diseño de la arquitectura y de las tecnologías de construcción, posibilitando que el diseño y la arquitectura puedan ser alterados independientemente. Una vez definidos los modelos, se definen una serie de transformaciones que generan nuevos modelos, permitiendo ir de modelos más abstractos a otros más concretos. MDA define un framework para procesar y relacionar modelos.

La transformaciones entre modelos se realizan haciendo uso de herramientas automáticas, como herramientas de transformación de modelos, las cuales permiten el refinamiento de los mismos.

Nuestra propuesta consiste en tomar el modelo de diseño UML obtenido como resultado del trabajo anterior, el cual corresponde a una instancia del metamodelo UML. A partir de este modelo pretendemos generar código utilizando Acceleo [9]. Esta herramienta permite la generación de código a través de cualquier tipo de metamodelo compatible con EMF como UML 1, UML 2 e incluso metamodelos adaptados a nuestro dominio específico.

De esta manera se pretende lograr un acercamiento a la automatización de la construcción del software aplicada a aquellos procesos de desarrollo dirigidos por casos de uso.

### Resultados y Objetivos

En esta etapa se trabajará en la generación automática de código a partir del modelo de diseño obtenido como resultado de aplicar las sucesivas transformaciones QVT definidas entre el modelo de casos de uso de sistema y el modelo de diseño, como se observa en la Fig.1.

La aplicación de las transformaciones QVT produce un modelo de diseño que consiste en un diagrama UML de clases de diseño.

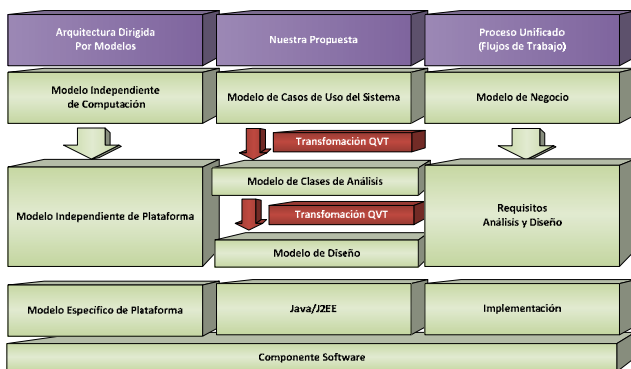


Fig. 1. Transformación modelo CUs a Modelo de Clases de Diseño

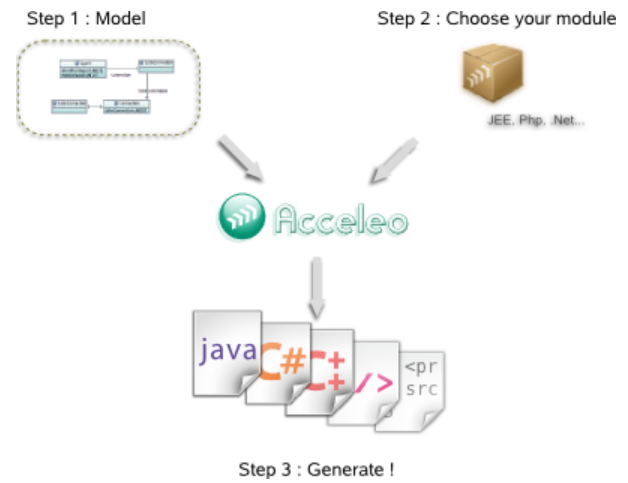


Fig. 2. Generación de código utilizando Acceleo

Este modelo nos resulta conveniente para ser utilizarlo como modelo entrada para la herramienta Acceleo como podemos ver en la Fig.2, obteniendo como resultado el código generado del lenguaje seleccionado.

Acceleo es una herramienta de transformación de modelo a texto, con un lenguaje basado en plantillas o templates. Teniendo en cuenta los estándares definidos por EMF, Acceleo brinda un enfoque MDA simple permitiendo la generación de archivos a partir de modelos UML, MOF o EMF

Acceleo brinda un asistente para la generación automática de las mismas. Primeramente se debe indicar el metamodelo a ser utilizado. Luego se debe seleccionar el modelo en sí. Acceleo está basado en los principales estándares MDA, lo que garantiza su compatibilidad e interoperabilidad con otras aplicaciones, como GMF. Es compatible con XMI, fue diseñado para trabajar con cualquier metamodelo y permite extender la funcionalidad ofrecida mediante la importación de librerías de Java, que pueden utilizarse para agregar funcionalidad a las plantillas y la generación de código. Una vez seleccionado el metamodelo y el modelo de entrada, se debe indicar las plantillas utilizadas en el proceso de generación. Luego se guarda la cadena indicando su nombre y la ubicación donde se alojará.

El proceso de transformación comenzará con la ejecución de una de las cadenas de generación creadas. El archivo producto de la transformación será generado en el formato que se haya indicado.

El resultado buscado está estrechamente vinculado al logro de un acercamiento hacia la automatización de las actividades que componen el proceso de desarrollo de software, transformado modelos ya validados como lo son las Plantillas Genéricas de descripción, análisis y diseño de casos de uso, con el fin de obtener un producto de software de calidad, cumpliendo con los requerimientos iniciales.

### Formación de Recursos Humanos

Durante el desarrollo de esta línea de investigación han logrado obtener el título Magister en Ingeniería de Software tres integrantes del grupo de trabajo. Otro integrante está actualmente trabajando es su tesis de Magister. Dos grupo de alumnos están realizando su trabajo final de fin de carrera de Analista en Computación, otro grupo ha logrado obtener el título de Licenciado en Ciencias de la Computación.

También, se han formado ayudantes de segunda en las asignaturas de Análisis y Diseño de Sistemas, Ingeniería de Software, Base de Datos y Proyecto.

Los temas abordados en esta línea de investigación brindan un fuerte aporte al proceso de perfeccionamiento continuo de los autores de carreras de computación en Universidades Nacionales como del exterior.

### Referencias

- [1] Ariel Arsaute, et al. “Hacia una integración de MDA y el Proceso Unificado a través de reglas de transformación QVT”. 13th Argentine Symposium on Software Engineering, ASSE 2012.
- [2] Ariel Arsaute, et al. “Transformación del modelo de análisis al modelo de diseño utilizando QVT”. XIV Workshop de Investigadores en Ciencias de la Computación 2012.
- [3] Jacobson, I. El Proceso Unificado de Desarrollo de software. Addison -Wesley, EE.UU. 2000.
- [4] UML, Unified Modeling Language (UML) Resource Page. <http://www.omg.org/#UML2.0>
- [5] Miller, J., Mukerji, J., MDA Guide Version 1.0.1 Document number omg/2003-06-01, Disponible en: <http://www.omg.com/mda>, 2003
- [6] Object Management Group, “Meta Object Facility (MOF)2.0Query/View/Transformation Specification” <http://www.omg.org/docs/ptc/05-11-01.pdf>.
- [7] ikv++: medini QVT. <http://www.ikv.de/>, ultimo acceso Agosto 2011
- [8] Eclipse Modeling Framework, URL: <http://www.eclipse.org/modeling/emf/>, ultimo acceso Abril 2009.
- [9] Acceleo. Home: <http://www.acceleo.org/>. Última visita Marzo 2012.
- [10] N. Debnath, F. A. Zorzan, G. Montejano and D. Riesco, “Transformation of BPMN Subprocesses Based in SPEM Using QVT”, 2007 IEEE INTERNATIONAL CONFERENCE on ELECTRO/INFORMATION TECHNOLOGY, May 17-20, 2007, Marriott O Hare, Chicago, IL, USA. <http://www.eit-conference.org/eit2007/>