Aplicando UML y DSL en el enfoque MDA

[1]**Daniel A. Giulianelli**
dgiulian@unlam.edu.ar

[2]**Claudia F. Pons**
cpons@lifia.info.unlp.edu.ar

[1]**Pablo M. Vera**
pablovera@unlam.edu.ar

[1]**Rocío A. Rodriguez**
rrodri@unlam.edu.ar

[1]**Victor F. Fernández**
vfernandez@unlam.edu.ar

[1] **Universidad Nacional de La Matanza (UNLaM)**
Departamento de Ingeniería e Investigaciones Tecnológicas
Florencio Varela 1903, San Justo, Buenos Aires, Argentina

[2] **Universidad Nacional de La Plata (UNLP)**
Facultad de Informática LIFIA-Laboratorio de Investigación y Formación en Informática Avanzada
Calle 50 y 150 La Plata, Buenos Aires, Buenos Aires, Argentina

RESUMEN

El enfoque MDA (Model Driven Architecture) está basado en dos elementos: los modelos y las transformaciones, mediante los cuales se dirige el proceso de desarrollo. Cada transformación arrojará un resultado más detallado hasta obtener el código necesario para implementar la solución en una plataforma específica.

En este paper se plantea la utilización de dos lenguajes de modelado aplicados al enfoque de MDA. Un lenguaje de propósito general como lo es UML (Unified Modeling Language) aplicado en un nivel alto de abstracción y DSL (Domain Specific Lenguage) aplicado a un nivel más bajo, cercano a la implementación de la solución final en una determinada plataforma.

Palabras Clave: Modelado, MDA, UML, DSL

CONTEXTO

Este proyecto forma parte de la línea de investigación “Ingeniería de Software”.

La Institución que coordina el Proyecto es la Universidad Nacional de La Matanza (UNLaM). Forman parte de la línea de investigación las siguientes instituciones:

- Universidad Nacional de La Plata (UNLP)
- Universidad Nacional del Centro de la Provincia de Buenos Aires (UNICEN)
- Universidad Nacional del Sur (UNS)
- Universidad Nacional de Lomas de Zamora (UNLZ)

Los Organismos que contribuyen al financiamiento del proyecto son:

- El Ministerio de Educación a través del Programa de Incentivo al Docente Investigador.

- La UNLaM a través del programa CyTMA (Ciencia y Tecnología La Matanza).

1. INTRODUCCION

Actualmente los sistemas son muy disimiles uno de otro es por ello que al modelar un sistema que pertenezca a un determinado dominio, UML (Unified Modeling Language)[6] resulta ser muy amplio y complejo de adaptarse a las características particulares de dicho dominio. Al momento de modelar el sistema, será necesario analizar el vocabulario de UML (simbología e incluso diagramas que pueden ser aplicados) y extender el lenguaje por ejemplo por medio de estereotipos que permitan modelar características no nomencladas.

En algunos dominios particulares extender a UML resulta complejo, por ejemplo en sistemas: de control, microcontroladores, aplicaciones paralelas, distribuidas, móviles, etc.

DSL (Domain Specific Lenguage)[2] ha sido creado con la idea de poder modelar características particulares de dominios específicos como los que se mencionaban anteriormente.

En este artículo se propone modelar al sistema con el enfoque MDA (Model-Driven Architecture) [4], [5] utilizando a UML como lenguaje de modelado que permite analizar al sistema desde un punto de abstracción alto y a DSL en un nivel más bajo de abstracción y más cercano a la codificación específica en una determinada plataforma.

1.1 MDA

Model-Driven Architecture (MDA) es un enfoque ampliamente aceptado para el desarrollo de sistemas de software complejos. "Es una iniciativa del Object Management Group (OMG), que representa un nuevo paradigma de desarrollo de software donde los modelos guían todo el proceso de desarrollo…"[1]

MDA propone el uso de modelos en todas las fases de desarrollo, desde la especificación y análisis hasta la implementación. La transformación de modelos es la base de MDA; comenzando por un modelo independiente de la plataforma el objetivo es lograr, en cada paso, modelos más específicos.

"MDA fue establecida como una arquitectura para el desarrollo de aplicaciones; tiene como objetivo proporcionar una solución para los cambios de negocio y de tecnología, permitiendo construir aplicaciones independientes de la implementación; representa un nuevo paradigma en donde se utilizan modelos del sistema, a distinto nivel de abstracción, para guiar todo el proceso de desarrollo". [3]

La figura 1 muestra el esquema de MDA.

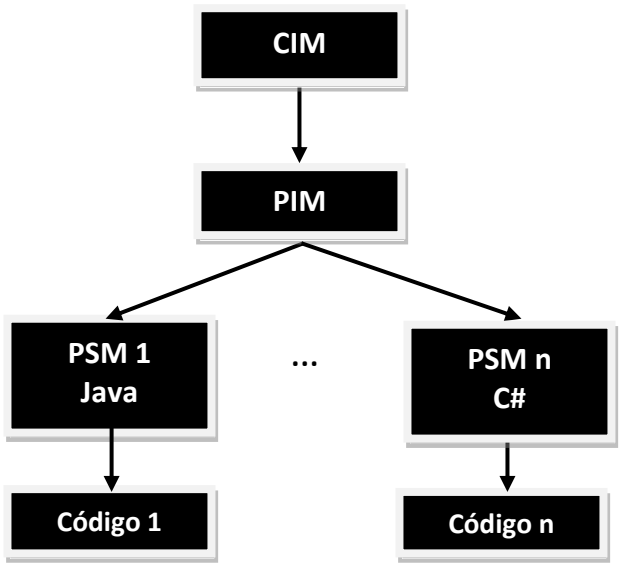

Figura 1- Esquema MDA

En el esquema de la figura 1, se visualizan distintos componentes del enfoque MDA: CIM,

PIM, PSM. A continuación se definen cada uno de estos [4]:

- **CIM:** Es un modelo independiente de lo computacional. No muestra detalles de la estructura de un sistema. Muchas veces es denominado modelo de dominio, y para su especificación se utiliza un vocabulario que es familiar a los practicantes del dominio en cuestión.

  Se asume que el usuario principal del CIM, (el practicante del dominio) no tiene conocimiento sobre los modelos o artefactos que se utilizan para realizar la funcionalidad para la cual se articulan los requerimientos en el CIM.

  El CIM juega un rol importante al unir la brecha entre:
  - aquellos que son expertos en el dominio y sus requerimientos
  - aquellos que son expertos en el diseño y construcción de artefactos

  Ya que juntos satisfacen los requerimientos del dominio.

- **PIM:** Es un modelo independiente de la plataforma. Exhibe un grado específico de independencia de la plataforma de modo que puede ser adecuado para utilizarlo con varias plataformas distintas.

- **PSM**: Es un modelo específico de la plataforma. Combina las especificaciones del PIM con los detalles que indican como ese sistema utiliza un tipo particular de plataforma.

"La idea clave subyacente es que, si se trabaja con modelos, se obtendrán importantes beneficios tanto en productividad, portabilidad, interoperatividad como mantenimiento. Podemos dividir el proceso MDA en tres fases; en la primera, se construye un modelo independiente de la plataforma (PIM), este es un modelo de alto nivel del sistema, independiente de cualquier tecnología; luego, se transforma el modelo anterior a uno o más modelos específicos de la plataforma (PSM), este modelo es de más bajo nivel que el PIM y describe al sistema de acuerdo con una tecnología de implementación determinada; por último, se genera el código fuente a partir de cada PSM. La división entre PIM y PSM está vinculada al concepto de plataforma que no está, aún, claramente definido. MDA, además, presenta un modelo independiente de los aspectos computacionales (CIM) que describe al sistema dentro de su ambiente y muestra lo que se espera de él sin exhibir detalles de cómo será construido". [3]

1.2 Lenguajes de Modelado: UML – DSL

Se cuenta con dos lenguajes de modelado los cuales persiguen distintos propósitos:

- UML - Lenguaje de propósito general
- DSL - Lenguaje de propósito específico

Los lenguajes de dominio específico, como por ejemplo DSL, son una alternativa a UML para modelar aplicaciones. A diferencia de UML no tienen estructuras generales sino que para modelar cada tipo de aplicación se debe definir un DSL específico con las entidades que se necesiten modelar. Esto hace que el lenguaje sea más acotado y específico.

UML al ser de propósito general se vale de estereotipos y profiles para poder adaptarse lo más posible a dominios específicos mientras que un DSL nace específicamente para dicho dominio.

Los DSL al ser más acotados son más propicios para la generación de código.
Sin embargo ambas tecnologías pueden convivir ya que por ejemplo es posible utilizar UML para modelar de forma genérica independiente de la plataforma (PIM platform independient model) y tener un DSL de más

bajo nivel ya dependiente de la plataforma (PSM platform specific model) que permita de forma más sencilla la generación de código. Por lo tanto se podrían tener varios PSM modelados en DSL para cada plataforma sobre la cual se desee generar código (ver figura 2).

2. LINEAS DE INVESTIGACIÓN Y DESARROLLO

Esta línea de investigación se enfoca en utilizar dos lenguajes de modelado (UML-DSL) aprovechando las ventajas de ambos aplicándolos a distintas etapas del modelado.

Para ello se persigue los siguientes objetivos:

- Modelar una aplicación bajo el esquema de MDA (Model Driven Architecture).
- Utilizar UML para modelar el PIM
- Utilizar DSL para modelar el PSM
- Desarrollar una herramienta que permita generar automáticamente código a partir de cada PSM construido.

3. RESULTADOS OBTENIDOS / ESPERADOS

En el presente artículo se plantea que UML y DSL no son excluyentes sino que pueden complementarse y aplicarse al enfoque de MDA, utilizándose UML en un nivel mayor de abstracción y DSL en un nivel más cercano a la implementación.

Actualmente se están modelando diversos sistemas aplicando la presente metodología.

Como trabajo futuro se construirá una herramienta que permita generar código a partir de cada PSM.

4. FORMACION DE RECURSOS HUMANOS

El proyecto en particular está conformado por:

- Director
- Co-Director
- Dos Investigadores Principales
- Tres Investigadores Formados
- Dos Investigadores en Formación
- Cuatro Alumnos

Los proyectos de UNLaM vinculados con la presente línea de trabajo son tres y nuclean alrededor de 40 personas

- Directores: Cuatro
- Co-Directores: Dos
- Investigadores Principales: Cinco
- Investigadores Formados: Seis
- Investigadores en formación: Cinco
- Alumnos: Dieciséis

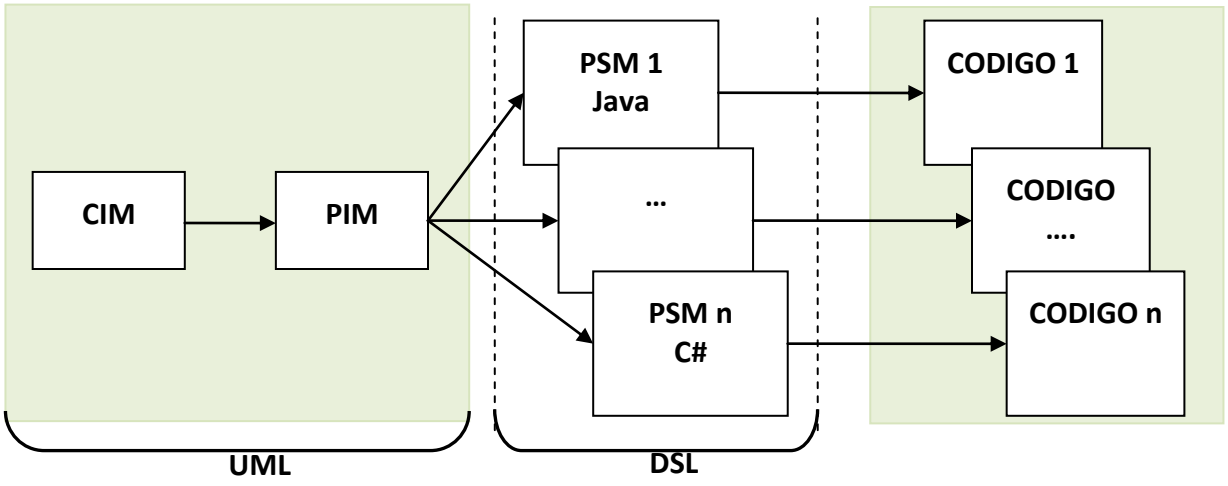

Figura 2- Aplicando UML y DSL al enfoque MDA

En relación a la línea de investigación se vinculan:

- Tesis de Posgrado en Curso: Cinco
- Tesis de Posgrado Aprobadas durante el año 2009: Dos
- Tesinas de Grado (Proyecto Final de Carrera) aprobadas durante el año 2009: Once
- Tesinas de Grado (Proyecto Final de Carrera) en curso: Siete

5. BIBLIOGRAFIA

[1] Lopez E, Gonzalez G, Lopez M, Iduañate R, Proceso de Desarrollo de Software Mediante Herramientas MDA, 2007 http://www.iiisci.org/Journal/CV$/risci/pdfs/C476AI.pdf

[2] Microsoft - MSDN, About Domain-Specific Languages, 2008 http://msdn.microsoft.com/en-us/library/bb126278(v=VS.90).aspx

[3] Neil Carlos, Pons Claudia, Aplicando MDA al Diseño de un Datawerhouse Temporal, 2006. http://caeti.uai.edu.ar/archivos/240_APLICANDO_MDA_AL_DISENIO_DE_UN_DATAWAREHOUSE_TEMPORAL.PDF

[4] OMG, MDA Guide Version 1.0.1, 2003 http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf

[5] OMG, Model Driven Architecture, 2009 http://www.omg.org/mda/

[6] OMG, Unified Modeling Language, Infrastructure, Version 2.2 (2009). http://www.omg.org/spec/UML/2.2/Infrastructure/PDF/