



resumen

La demanda de software ha venido aumentando en los últimos años y se proyecta un mayor crecimiento en el futuro. Las líneas de producto basadas en modelos (MD-SPL por sus iniciales en inglés) nacen como una alternativa para la generación de artefactos que comparten características comunes, pero donde también pueda manejarse la variabilidad.

# Cupi2 MD-SPL: caso de líneas de producto basadas en modelos

Juan Sebastián Montaña O. • Nicolás López G.

**Desarrollar** aplicaciones de software de la forma tradicional es un proceso que resulta costoso en cuanto a la cantidad de recursos humanos necesarios, tiempo de desarrollo invertido y calidad del producto final. Para dar respuesta a estos problemas, han surgido propuestas que buscan reducir el esfuerzo en tiempo de codificación y dar paso al modelamiento de los sistemas en un mayor nivel de abstracción.

## introducción

MDA (Model Driven Architecture) [9] es un enfoque que permite un mayor grado de abstracción, separando el mundo del problema del de la tecnología que se utilizará para implementar la solución. Esta teoría está basada en la construcción de modelos independientes de la plataforma, los que expresan la lógica del negocio, y que posteriormente se integran con los conceptos específicos de la tecnología seleccionada para la implementación. Adicionalmente, este enfoque facilita la combinación de modelos y generación de código, basándose en las transformaciones de modelo a modelo y de modelo a código. MDA utiliza los modelos como elementos de primera clase en el desarrollo de aplicaciones, a diferencia de otros paradigmas que los utilizan únicamente como elementos de representación, documentación y comunicación.

Otro enfoque muy utilizado cuando se busca construir aplicaciones con características en común es SPL (Software Product Line) [4]. Las





uno  
onu

SPL's permiten el ahorro en tiempo de desarrollo y proveen calidad en las aplicaciones, aprovechando el conjunto de características comunes que éstas comparten. De igual modo, puede manejarse la variabilidad dentro de una SPL, permitiendo que productos que pertenecen a una misma familia, tengan características particulares y que los hagan diferentes de otros productos. [7]

## Antecedentes

En semestres anteriores, Carlos Parra y Kelly Garcés, estudiantes de Maestría en Ingeniería de Sistemas y Computación de la Universidad de los Andes, desarrollaron una herramienta conocida como QualDev MD-SPL (Model Driven Software Product Line) [7], [10]. El propósito de esta herramienta es facilitar el proceso de generación de código a partir de una fusión de los enfoques MDA y SPL. Particularmente, dentro del alcance definido, se estuvo trabajando en una línea de producto para los ejercicios que se utilizan en el proyecto Cupi2 [5].

Cupi2 es un proyecto del grupo de construcción de software de la Universidad de los Andes, puesto en práctica desde el primer semestre del año 2005, cuyo objetivo es “la búsqueda de nuevas maneras para enfrentar el problema de enseñar a programar” [5]. Para este fin, se apoya en ejemplos y ejercicios que se realizan en los diferentes niveles que componen cada uno de los cursos que hacen parte del proyecto. Para cada curso, se trabajan 6 niveles, donde cada uno de estos tiene objetivos particulares a ser cumplidos y trata temas específicos en el área de programación.

La metodología utilizada en estos cursos consiste en entregar a los estudiantes una parte del código de la aplicación, y de acuerdo con un enunciado que contiene los requerimientos que deben cumplirse, ellos completan el ejercicio para que quede funcionando completa y correctamente.

Semestralmente, el proyecto Cupi2 requiere la construcción de varias aplicaciones ejemplo, como material de soporte para cada nivel. La tarea resulta bastante compleja si se tiene en cuenta que en total son 18 los niveles; adicionalmente, pueden inyectarse defectos al construir, de forma manual, cada una de estas aplicaciones, utilizadas por todos los estudiantes que toman estos cursos.

QualDev [11] es un grupo formado por estudiantes de pregrado, maestría, doctorado y profesores, que tiene como propósito trabajar por la mejora de las prácticas de la Ingeniería de Software. En la actualidad, el grupo tiene en curso varios proyectos, particularmente ayudar a las empresas de desarrollo a mejorar sus procesos de construc-





ción y mantenimiento de software. La visión a futuro del grupo es construir líneas de producción de software, basadas en los enfoques MDA y SPL.

En el primer semestre de 2007, se comenzó a evaluar el uso de QualDev MD-SPL generando el código de algunos ejercicios que ya habían sido desarrollados de forma manual, encontrándose algunas fallas en la herramienta. Mucho del código generado presentaba errores de compilación, por lo que el usuario de la herramienta debía realizar algunas correcciones manualmente; el otro problema se encontró en el terreno no funcional: tanto la herramienta como el proceso de generación de los ejercicios no es fácil de manejar, en particular para una persona que no conoce a fondo la teoría de los modelos. De este modo, el propósito del trabajo de grado realizado durante el semestre consistió en mejorar la usabilidad de la herramienta, ocultando la complejidad del ambiente y disminuyendo la curva de aprendizaje, al igual que en la generación de código sin errores, funcional y consistente con lo esperado para cada uno de los niveles soportados.

## La Herramienta

QualDev MD-SPL es un plug-in de Eclipse que se integra con otras herramientas como EMF (Eclipse Modeling Framework) [6], ATD (ATL Development Tools) [2], AMW (Atlas Model Weaver) [2] y Acceleo [1]. Para mayores referencias sobre las herramientas de QualDev MD-SPL, su arquitectura y detalles de implementación ver [7] y [10].

A continuación se muestra un gráfico con las herramientas que soportan la aplicación.

**Figura 1.**  
**Infraestructura de**  
**QualDevMD-SPL**



Eclipse es utilizado como ambiente de desarrollo; se usa el framework de modelamiento de Eclipse, para apoyar el enfoque MDA, y sobre éste se trabaja con las demás herramientas, que permiten realizar transformaciones modelo a modelo y modelo a código. QualDev MD-SPL hace uso de toda esta infraestructura para permitir la generación de aplicaciones funcionales a partir de la definición de modelos.

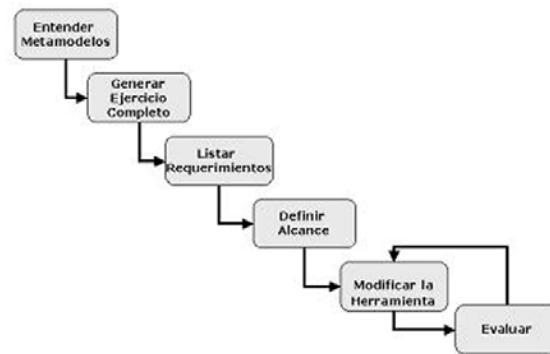




## El Proceso

Dado que la herramienta ya se encontraba en un estado avanzado, fue necesario diseñar un plan de trabajo sobre cómo abordar el problema y realizar los cambios necesarios. Primero se investigó sobre el enfoque MDA y se analizó el trabajo desarrollado en cuanto a los metamodelos que soportan la SPL de Cupi2. A partir de estos estudios y de una primera versión de un tutorial, se generó un ejercicio para ver la herramienta en funcionamiento y conocer los productos generados. Después se obtuvo un listado con las funcionalidades que debía cumplir la herramienta y a partir de esto se definió el alcance que tendría el proyecto, dada la complejidad de las funciones y el tiempo disponible. Al final, de manera iterativa se realizaban cambios en la herramienta y eran evaluados para comprobar la correctitud de la solución. Dentro del alcance del proyecto no se contempló la verificación formal de los modelos, simplemente se realizó validación de la herramienta a través de la generación de ejercicios que se habían desarrollado en forma manual. Las dos aplicaciones (el ejercicio creado por un asistente graduado y el ejercicio generado usando la herramienta) se comparaban, comprobando que no hubiera errores y que todas las funcionalidades se realizaran de forma correcta. El proceso completo se muestra en la figura 2.

**Figura 2.**  
Estrategia de  
solución



## La Solución

Algunas etapas del proceso de generación de los ejercicios eran un poco confusas, toda vez que la herramienta no reflejaba los cambios que el usuario realizaba o el procedimiento era extenso y poco entendible. Para atacar estos problemas, se diseñó un nuevo proceso, en el que se automatizaban pasos y se aseguraba mostrar al usuario el reflejo de sus acciones.

Por otro lado, se revisó la parte funcional de la herramienta, donde se corrigieron fallas que estaban generando errores en las aplicaciones finales.

Todo este trabajo estuvo enfocado en cumplir con las promesas planteadas por MDA y SPLs, continuando con el uso que se tenía de estas teorías cuando la herramienta fue





diseñada. El proceso de generación de las aplicaciones estuvo guiado en cada etapa por la definición de modelos, donde las características comunes se escogían en diferentes etapas del proceso. A través del diseño de modelos del mundo, transformaciones y modelos de la tecnología, soportándose en las herramientas nombradas anteriormente, un usuario completaba la generación de una aplicación Cupi2.

## Los Resultados

Para la validación final del proyecto, se crearon y evaluaron escenarios de calidad, basados en la teoría planteada por Bass, Clements y Kazman [3], que permitieran la obtención de resultados concretos, medibles y comparables. Se diseñaron escenarios para medir la usabilidad de la herramienta, teniendo como parámetros de evaluación el tiempo empleado en generar un ejercicio completo, el total de cambios que el usuario debía realizar manualmente a lo largo del proceso y la cantidad de errores de código encontrados en la aplicación generada.

Dado que la herramienta se diseñó para ser utilizada por profesores o asistentes graduados, los escenarios se validaron con dos estudiantes de maestría. Los resultados obtenidos usando la versión inicial de la herramienta, con su respectivo manual de usuario, comparado con la versión que contenía las modificaciones se muestran a continuación:

	Antes	Después
<b>Total cambios por realizar</b>	8	0
<b>Total errores generados</b>	13	0
<b>Tiempo Empleado</b>	4,57 horas	0,625 horas

**Tabla 1**

Puede verse que con el nuevo tutorial y última versión de la herramienta, el tiempo utilizado para generar un ejercicio disminuyó de forma notable. Esto se debe a que el proceso de generación de ejercicios fue rediseñado para automatizar bastantes pasos, al igual que se eliminaron los errores que existían. Así mismo, es posible observar que no se generó ningún error en la aplicación final y que el usuario no debía hacer ningún cambio para tener funcionando el ejercicio.

A través de la implementación de una SPL basada en modelos para el proyecto Cupi2, pueden verse las ventajas de usar los enfoques explicados con anterioridad. Definitivamente, se evidencia disminución en el tiempo de desarrollo de las aplicaciones; la cuestión ya no es trabajar en codificación sino modelar el mundo del problema, manejar rasgos opcionales que permitan diferenciar los productos y componer los modelos entre sí, para finalmente obtener el código fuente de una aplicación lista para ser utilizada. Otra ventaja de haber usado estos enfoques es la posibilidad de reutilización de artefac-





tos, dado que los activos (metamodelos, transformaciones y plantillas de código) son usados para soportar el proceso de generación de todas las aplicaciones.

Cabe resaltar que los casos de prueba realizados no permiten llegar a conclusiones concretas, simplemente el propósito fue verificar la viabilidad de la herramienta realizada. Existe bastante trabajo aún por realizar, como seguir proponiendo mejoras de la herramienta en cuanto a usabilidad, generar nuevos ejercicios para los niveles soportados y fortalecer la herramienta permitiendo la generación de ejercicios para nuevos niveles.

Finalmente, y retomando los principios planteados por estos paradigmas de MDA y SPL, que cada día cobran más fuerza y ganan adeptos, el nivel de abstracción es mayor; la tecnología es por completo independiente del mundo del problema; se pueden manejar características comunes y también administrar la variabilidad en los productos; y, las aplicaciones son generadas de forma rápida, eficaz y con estándares de calidad. Todas estas ventajas demuestran el potencial que se logra al usar las metodologías mencionadas, quedando expuesta la motivación para continuar trabajando y desarrollar habilidades que permitan una industria mucho más competitiva.

## Referencias

- [1] Acceleo: MDA generator – Home. [www.acceleo.org/pages/home/en](http://www.acceleo.org/pages/home/en) última visita Junio 2007
- [2] ATL Project. [www.eclipse.org/m2m/atl/](http://www.eclipse.org/m2m/atl/) última visita Junio 2007
- [3] Bass L, Clements P, Kazman R, Software Architecture in Practice, Addison-Wesley, 1997
- [4] Clements, P, Northrop, L. Software Product Lines: Practices and Patterns. Addison-Wesley, 2002.
- [5] Cupi2. En línea: <http://cupi2.uniandes.edu.co/> Última visita Junio 2007
- [6] Eclipse Modeling - EMF – Home. [www.eclipse.org/emf/](http://www.eclipse.org/emf/) última visita Junio 2007
- [7] Garcés, K. Administración de variabilidad en una línea de productos basada en modelos. Documento de Tesis de Maestría en Ingeniería de Sistemas. Universidad de los Andes, Bogotá – Colombia, 2007.
- [8] Miller J, Mukerji J. MDA Guide, 2003.
- [9] Model Driven Architecture (MDA). Reference site <http://www.omg.org/mda>.
- [10] Parra, C. Cupi2: una línea de productos basada en modelos. Documento de Tesis de Maestría en Ingeniería de Sistemas. Universidad de los Andes, Bogotá – Colombia, 2007.
- [11] QualDev Group. En línea: <http://qualdev.uniandes.edu.co/> Última visita Junio 2007

**Juan Sebastián Montaña Ortega.** Estudiante último semestre Ingeniería de Sistemas y Computación Universidad de los Andes. Miembro activo del proyecto QualDev, Universidad de los Andes.

**Nicolas López Giraldo.** Ingeniero y MSc, Ingeniería de Sistemas y Computación, Uniandes. Profesor Instructor del departamento de Ingeniería de Sistemas y Computación de la Universidad de los Andes. Líder del proyecto QualDev, Universidad de los Andes.

