UNIMAR
ISSN: 0120-4327, ISSN Electrónico: 2216-0116,
Universidad Mariana, San Juan de Pasto, Nariño, Colombia, 2015.

Una visión a la propuesta de Arquitectura Dirigida por Modelos*

Luis Muñoz**✉
Guillermo Solarte ***
Douglas Hernández Heredia****

Cómo citar este artículo / To reference this article / Para citar este artigo: Muñoz, L., Solarte, G. y Hernández, D. (2015). Una visión a la propuesta de Arquitectura Dirigida por Modelos. *Revista UNIMAR, 33*(2), 271-279.

Fecha de recepción: 02 de junio de 2015
Fecha de revisión: 06 de agosto de 2015
Fecha de aprobación: 30 de noviembre de 2015

RESUMEN

Este documento presenta un acercamiento a la iniciática *Model Driven Architecture* (MDA, Arquitectura Dirigida por Modelos) propuesta por *Object Management Group* (OMG). MDA separa la lógica de la aplicación del negocio. Esta iniciática es esencial a la hora de tener un enfoque de desarrollo de software dirigido por modelos. El artículo se centra en los principales conceptos presentados en la propuesta MDA, identificando ventajas y desventajas cuando se hace la aplicación del estándar. En la parte final del documento se puede observar algunos casos de estudio que utilizaron este estándar para desarrollar software basado en modelos.

Palabras clave: Arquitectura Dirigida por Modelos, estándar, desarrollo de software, software.

A vision to the proposal of Model Driven Architecture

ABSTRACT

This paper shows an approach to the initiatory Model Driven Architecture (MDA) proposed by Object Management Group (OMG). MDA separates the application logic business. This initiation is essential in having development approach model-driven software. The article focuses on the main concepts presented in the proposal MDA, identifying advantages and disadvantages when applying the standard. In the final part of the document you can see some case studies that used this standard for developing model-based software.

Key words: Model Driven Architecture, standard, software development, software.

Uma visão a proposta de Model Driven Architecture

RESUMO

Este artigo apresenta uma abordagem para a iniciação *Model Driven Architecture* (MDA, Arquitetura Dirigida por Modelos) proposta pelo *Object Management Group* (OMG). MDA separa a lógica de negócios do aplicativo. Esta iniciação é essencial em ter abordagem de desenvolvimento de software orientado à modelo. O artigo centra-se sobre os principais conceitos apresentados na proposta MDA, identificando vantagens e desvantagens ao aplicar as normas. Na parte final do documento você pode ver alguns estudos de casos que utilizaram esse padrão para o desenvolvimento de software baseado em modelo.

Palavras-chave: Model Driven Architecture, padrão, desenvolvimento de software, software.

* Artículo de Revisión de Tema.

**✉Ingeniería de Sistemas M.Sc. Profesor Auxiliar de la Universidad Tecnológica de Pereira, Risaralda, Colombia. Correo electrónico: lemunozg@utp.edu.co
*** Ingeniero de Sistemas. Profesor Asociado de la Universidad Tecnológica de Pereira, Risaralda, Colombia. Correo electrónico: roberto@utp.edu.co
**** Estudiante de Ingeniería de Sistemas y Computación, Universidad Tecnológica de Pereira, Risaralda, Colombia. Correo electrónico: douhernandez@utp.edu.co

I. Introducción

La evolución de las tecnologías de software y el deseo de la industria de adaptarse a esos cambios, ha planteado nuevos paradigmas para su desarrollo. Por lo general las aplicaciones requieren nuevas implementaciones que se apropien de los conceptos de las nuevas y futuras tecnologías que faciliten su integración con otros sistemas y que resuelvan problemas de código con un buen rendimiento. Con las anteriores necesidades la realidad es que los costos que las empresas de software pagan por lograr la portabilidad, interoperabilidad e independencia de la plataforma para sus aplicaciones, se traduce actualmente en un aumento descontrolado de tiempo y costos de desarrollo.

Los ingenieros de software construyen modelos de sistemas de información utilizando diagramas con alto nivel de abstracción, en ocasiones lograr una buena abstracción de un objeto puede ser difícil para los ingenieros de sistemas, por esto surgen Arquitecturas Dirigidas por Modelos (MDA) esto es una iniciativa de The Object Management Group (Group., 2015), un grupo muy importante en el ámbito del desarrollo de software. Esta arquitectura pretende cumplir con las siguientes fases. Propone, en una primera fase, la definición de modelos de alto nivel de abstracción independientes de cualquier hardware, sistema operativo o plataforma que finalmente lo soporte (Modelo Independiente de la Plataforma, PIM). Esta característica intenta proteger parte de los cambios de la organización en el desarrollo del software, aislando la lógica de la aplicación y las reglas del negocio, de la tecnología y las plataformas específicas sobre las cuales el sistema se implementa.

En una segunda fase, el PIM es transformado en un modelo plataforma específica de los modelos (PSM), lo anterior se puede ilustrar en la Figura 1. Que visualiza aspectos generales de transformación. En aquellos casos en los que intervengan varias plataformas en la implementación, se generara un PSM por cada una de ellas. De este modo, la implementación del sistema se alcanza tras aplicar reiterativamente varios ciclos de transformación entre modelos; entendido el término implementación en el contexto de MDA como un PSM más, el cual incluye la información necesaria para construir el sistema y ponerlo a funcionar.

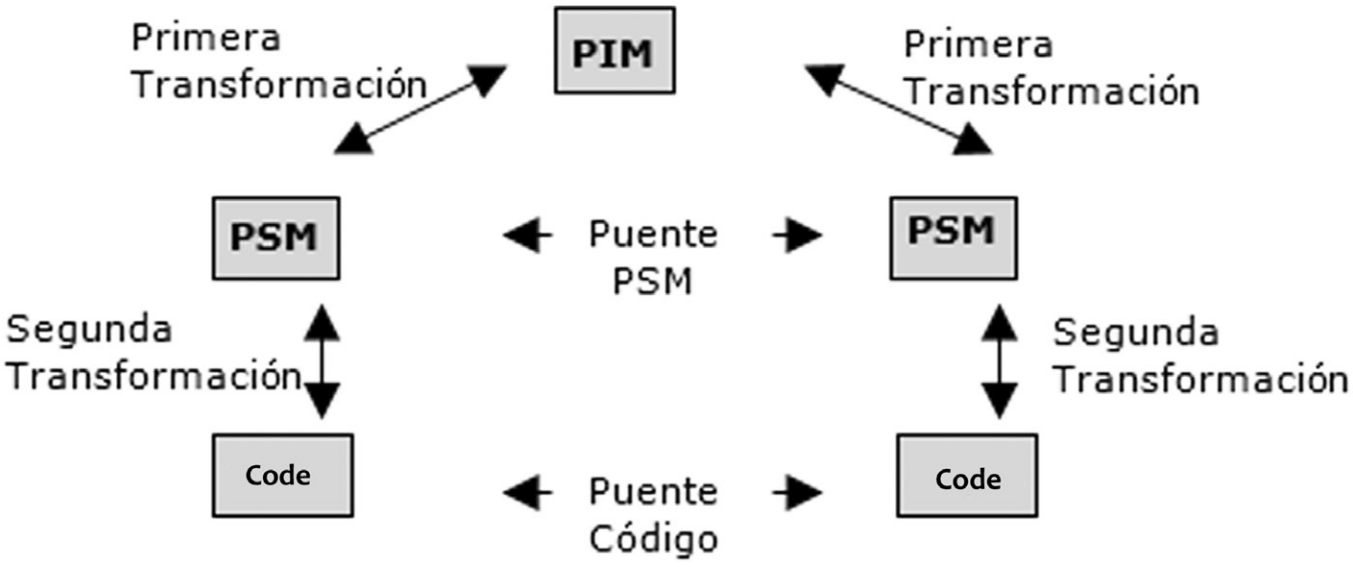


Figura 1. Transformación de modelos de PIM a PSM.

Revista ***UNIMAR*** *- 33 (2)- rev. UNIMAR.-* pp. 271-279.
ISSN: 0120-4327, ISSN Electrónico: 2216-0116,
Universidad Mariana, San Juan de Pasto, Nariño, Colombia, 2015.

La estrategia propuesta sugiere un diseño descendente, en el que los modelos que representan diferentes niveles de abstracción del sistema son gradualmente transformados (fusionados y/o refinados) hasta que se genera el código de la implementación como se ilustra en la Figura 1. En definitiva, una aproximación orientada a la creación de nuevos sistemas y a la generación de código para plataformas actuales y futuras.

Se plantean varios interrogantes que deben ser profundizados en el contexto de MDA, como son: ¿cómo mejorar los procesos de la reutilización dentro del contexto de MDA?, ¿qué beneficios puede aportar MDA frente a otros modelos?, ¿cómo tratar etapas del ciclo de vida del producto software sin alterar la relación entre los modelos y el código generado?

En esa medida, este artículo pretende realizar delimitación conceptual de la MDA en la ingeniería de software identificando diferentes tipos de modelos, sus transformaciones, operaciones, ventajas y desventajas.

Estos conceptos propuestos por la MDA (Kleppe, 2014) constituyen, el punto de partida para el desarrollo de un amplio espectro de técnicas que pretenden formalizar y automatizar los procesos implicados en el desarrollo de software.

Es importante considerar, al menos como objetivo ideal desde la propuesta de MDA por parte de OMG, definir varios lenguajes y herramientas de transformación de modelos como resultado de un intenso trabajo de investigación y desarrollo, tanto desde la industria como desde el mundo académico.

Sin embargo, todavía es necesario seguir investigando para conocer mejor la naturaleza de las transformaciones y las características deseables de un lenguaje estándar de transformación. Por ello, el interés actual está centrado principalmente en la experimentación con los lenguajes existentes a través de la escritura de definiciones de transformaciones para casos reales (Arando, 2006), en este sentido, marcos teóricos como son: el modelo de características y la taxonomía de transformaciones de modelos son muy útiles para comparar y evaluar las decisiones de desarrollo del software (Pressman, 1999).

2. Identificación de las plataformas de modelos

La idea principal del desarrollo utilizando MDA (Group., 2015) es la separación de la especificación del sistema de los aspectos puntuales de la implementación del mismo de acuerdo a la plataforma en la cual se desee realizar el desarrollo.

Para este fin, el marco de trabajo MDA especifica tres puntos de vista, los cuales a su vez generan un modelo que representa los resultados de la aplicación de cada punto de vista. Los tres puntos de vista MDA son:

- **Punto de vista independiente de computación**. Este punto de vista está centrado en el domino del sistema así como en los requerimientos, detalles de la estructura y procesamiento del sistema. Esta forma de ver el sistema genera un Modelo Independiente de Computación (CIM).

  Los CIM se asocian a los modelos de dominio y es recomendable que sean especificados a partir de un vocabulario común para todos los elementos (computacionales o humanos) implicados en el desarrollo del sistema.

- **Punto de vista independiente de plataforma**. Es la encargada de mostrar la especificación del sistema tomando en cuenta no solo las especificaciones de funcionamiento propias del sistema - especificadas en el CIM- sino también las especificaciones para la implementación en un medio computacional. Esta forma de ver el sistema genera El Modelo Independiente de Plataforma (PIM).

  El PIM representa los aspectos que no cambiarán de una plataforma a otra, de acuerdo a una tecnología o método de implantación escogido para la representación computacional.

- **Punto de vista específico de plataforma.** Esta vista combina el punto de vista independiente de plataforma con los detalles y características propias del uso de una plataforma de desarrollo. Esta forma de ver el sistema genera el Modelo Específico de Plataforma (PSM).

> En el PSM se puede observar la manera en la que un sistema usa las herramientas de la plataforma para el cumplimiento de los objetivos trazados en la etapa de especificación inicial.

Es importante identificar un elemento importante que surge del PSM que es la Implementación Específica de la Plataforma (PSI): es un código del sistema a desarrollar. El código -en su totalidad o en parte- será generado automáticamente mediante algoritmos de transformación.

Aplicando MDA

Con el fin de desarrollar un proyecto utilizando el enfoque MDA es necesario definir primero las abstracciones que se utilizarán para representar el dominio del problema y de las tecnologías de plataforma que se emplearán para el desarrollo del sistema. Estas abstracciones se describirán formalmente el uso de metamodelos o perfiles. En algunos casos estos metamodelos o perfiles pueden estar realizados por una especificación de OMG existente (Object Computing Inc. [OCI], 2015).

Cuando se decide desarrollar un proyecto basado en MDA, los arquitectos o diseñadores del MDA, deberán definir de forma cuidadosa los requerimientos del negocio, así como los modelos y herramientas que se utilizarán en el transcurso del desarrollo; por ejemplo, el arquitecto decide utilizar herramientas como UML para representar el diseño inicial del proyecto, o sea definir CIM basado en modelos de ingeniería de software, además de que es el responsable de definir las reglas para la entrada de datos en los modelos, también el arquitecto define restricciones para cuando se están transformando los distintos modelos del diseño de MDA.

El arquitecto definirá las traducciones entre cada uno de los modelos, además de estar asesorando y acompañando a los responsables de estas transformaciones, para que no se pierdan datos y sea coherente cada transformación del proyecto, cuando se pasa a la transformación de los modelos al código, es decir de PSM a Código, previamente ya se habrá definido en qué lenguaje o en qué plataforma se desarrollará el proyecto, como se utilizan herramientas especializadas en el manejo de MDA, y algunas de estas ya generan parte del código del software, dan la oportunidad a los desarrolladores de hacer parte del diseño del software final deseado, claro está que a las herramientas actuales les falta un nivel más alto de madurez a la hora de generar el código.

"Nuestra creencia es que ninguna herramienta única va a satisfacer las necesidades de todo el proyecto y que las herramientas existentes aún tienen que llegar a un nivel necesario de madurez y grado de interoperabilidad" (OCI, 2015, p. 4).

En la Figura 2 podemos apreciar un esquema del desarrollo de un proyecto siguiendo el modelo MDA.

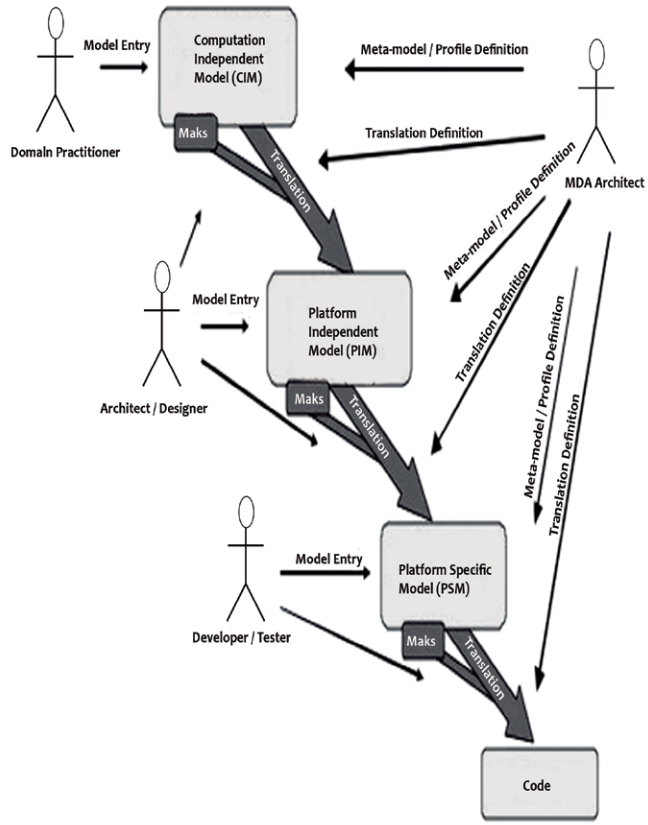


Figura 2. Esquema del desarrollo basado en MDA.
Fuente: OCI (2015).

Los modelos bien definidos son fuente de nueva información, con estos se puede llegar por ejemplo, a conocer las debilidades o fortalezas de un área de la organización (empresa), además los modelos son una buena fuente de documentación tanto para analistas, desarrolladores y el propio arquitecto del MDA.

Revista ***UNIMAR*** *- 33 (2)- rev. UNIMAR.-* pp. 271-279.
ISSN: 0120-4327, ISSN Electrónico: 2216-0116,
Universidad Mariana, San Juan de Pasto, Nariño, Colombia, 2015.

Como se puede apreciar, aplicar este este tipo de enfoque para la realización de los proyectos de software puede generar muchas ventajas, ya que los modelos aplicados para un tipo de proyecto pueden ser fácilmente adaptados para cualquier tipo de propuesta de negocio, sí desde un principio se hacen los modelos detallados y bien documentados; además los modelos se vuelven una marcar o memoria corporativa para cualquier organización, desde la captura de requisitos, diseño e implementación, se vuelven distinción del trabajo de la organización.

Algunos ejemplos de aplicación del modelo MDA

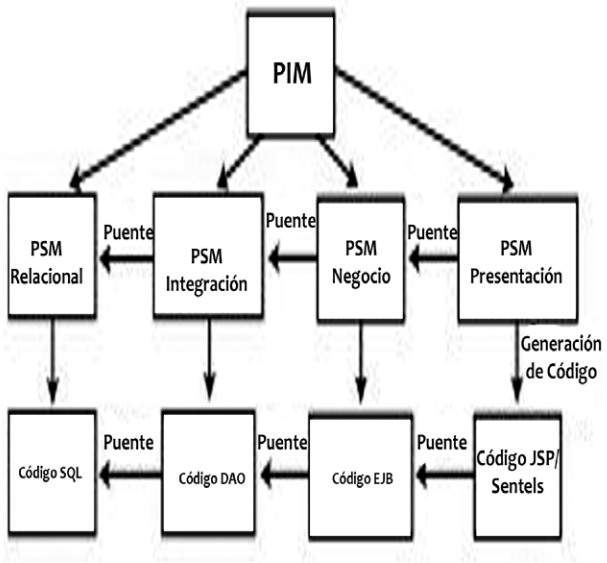
Con el fin de continuar con el estudio de las aplicaciones basadas en el enfoque MDA y sus fronteras (CIM, PIM y PSM), se estudiarán algunos proyectos de software que han sido desarrollados con esta metodología basada en modelos.

Técnicas de Ingeniería del Software aplicada a los sistemas agrícolas (recurso electrónico): un enfoque orientado a objetos y UML (Papajorgi y Pardalos, 2014)

Este artículo se basa en técnicas aplicadas al diseño software dedicado al campo agrícola, algunas de las técnicas que se buscan es usar diagramas UML, diagramas de secuencia y de colaboración, todas estas técnicas enfocadas a unificarlas con el estándar MDA, utilizando herramientas que se encuentran en el mercado; con esto se busca que el software que se produzca con estas herramientas sea un software de calidad para crear mejores sistemas agrícolas.

Enfoque MDA para el diseño de un data warehouse difuso (Zambrano, Varas y Urrutia, 2012)

La idea principal de esta investigación es la elaboración de un almacén de datos enfocado en un modelo MDA, las ventajas de utilizar este enfoque en el diseño de un data warehouse (DW) es que entre los distintos modelos del MDA hay un nivel de abstracción necesario de la aplicación y diseño en cada etapa del desarrollo; además que con este enfoque se logra una separación de cada etapa de trabajo de los proyectos DW. Por otro lado, en el ciclo de desarrollo de DW se han utilizado herramientas y diversas metodologías con modelos que no tienen un nivel de integración de las diferentes etapas del desarrollo de un DW.



Figuara 3. Enfoque MDA para DW.
Fuente: Zambrano et al. (2012).

El desarrollo de software dirigido por modelos en los repositorios institucionales

Los Repositorios Digitales están teniendo un gran auge en las comunidades universitarias, de manera especial, por su capacidad para difundir y preservar las actividades académicas y científicas, y normalmente, se registran en los llamados "directorios de repositorios" (Texier, De Giusti y Gordillo, 2013). El enfoque en que se desarrolló este articulo es ver la necesidad de desarrollar software con la metodología de desarrollo dirigido por modelos, ya que por la complejidad de los repositorios institucionales se requiere que las diferentes partes interesadas: desarrolladores, dueños del negocio y expertos del dominio, se pongan de acuerdo sobre qué debe llevar el sistema, qué ventanas, cómo sera el entorno gráfico, etc.; motivo por el cual se propone utilizar MDA para el desarrollo de este tipo de aplicaciones, ya que como hemos visto este estándar facilita y separa todos estos componentes de una forma muy ordenada y clara para el trabajo en equipo.

Estado de la complejidad arbitraria y Arquitectura Dirigida Por Modelos en el desarrollo de software en Cuba (Silega, Macias, Febles y Noguera, 2014)

Este documento se centra en hacer una evaluación y ver el estado de arte del desarrollo de software en Cuba, haciendo relevancia en que en

muchas ocasiones, los ingenieros de sistemas y desarrolladores de software se preocupan más por mejorar otras áreas de investigación, que mejorar su propia área; a esto se refieren caundo dicen que tenemos diferentes métodos para desarrollar software de calidad.

Diseño de servicios web semánticos utilizando el desarrollo de Software Dirigido por Modelos

Los servicios web semánticos ofrecen beneficios que ayudan a la evolución de la Web, como el descubrimiento, invocación y composición dinámica y automática de recursos, habilitan efectivamente la interoperabilidad entre sistemas, permitiendo una amplia gama de nuevos servicios y oportunidades de negocios en la Internet. La estructura necesaria para proveer estos beneficios, hace que su desarrollo sea un proceso complejo, requiriendo establecer formas más fáciles y dinámicas que garanticen reutilización, calidad y rapidez. El desarrollo dirigido por modelos realiza una contribución eficiente en estos aspectos, dado que trabaja de manera intrínseca conceptos como separación de conceptos, reusabilidad e interoperabilidad entre componentes (Vega y Umaña, 2014).

3. Transformaciones

Entre los diferentes modelos construidos en MDA existe una estrecha relación. Los modelos más abstractos son la base para la construcción de los modelos específicos, así como los modelos son los que soportan un nivel de abstracción mayor, esta relación es representada en esta arquitectura por las Transformaciones entre Modelos, las cuales constituyen una de las características fundamentales de esta propuesta.

Los procesos de transformación se denominan mappings y están conformados por una serie de reglas de transformación que son operativas en un determinado dominio, y que permiten pasar de un modelo a otro. Uno de los factores claves para la acogida de MDA, por parte de la comunidad informática, es su énfasis en los modelos.

Por ejemplo, en los modelos de MDA se transforma un modelo independiente de la plataforma en uno específico, añadiendo información que permita enlazar ambos modelos. La transformación de PIM a PSM puede llevarse de la siguiente manera: de forma totalmente automática, generando un PSM completo a partir del PIM. Para este método se utilizan herramientas especializadas que implementan diferentes algoritmos de transformación para pasar de un tipo de modelo a otro, y forman parte de uno de los pilares de MDA.

Como se ha mencionado anteriormente, para llevar acabo las transformaciones que sirven para pasar de un modelo a otro en un MDA, se deben tener claro las restricciones del sistema, así como las reglas para realizar dichas transformaciones; en su mayoría, las transformaciones son orientadas a objetos para representar y manipular los elementos del sistema, una transformación indica una generación automática de un modelo a partir de otro. Las transformaciones pueden ser:

- **Endógenas**: si están expresadas en el mismo lenguaje del modelo (Orozco, Giraldo y Trefftz, 2013).
- **Exógenas**: no están expresadas en el mismo lenguaje que los modelos (Orozco et al., 2013).
- **Verticales:** los lenguajes destino son diferentes entre sí (Orozco et al., 2013).
- **Horizontales**: los lenguajes destino no son diferentes entre sí (Orozco et al., 2013).

Además de esto, las transformaciones tienen un enfoque distinto en cada fase del esquema MDA; a continuación se explicará cada uno de los enfoques que se pueden tener.

Enfoque de modelo a Código: este enfoque se ve aplicado entre las fases PSM y código; existen diversas herramientas en el mercado que pueden generar un código básico del modelo representado, por ejemplo, herramientas UML generan un código básico en algunas plataformas de programación. Algunas de estas herramientas son Enterprise Architect.

Enfoque modelo a modelo: este enfoque se presenta cuando se transforma PIM a PSM o CIM a PIM, algunas características serán compartidas entre los modelos, las cuales pueden llegar a ser instancias de las otras, creando metamodelos de destino

Revista ***UNIMAR*** *- 33 (2)- rev. UNIMAR.-* pp. 271-279.
ISSN: 0120-4327, ISSN Electrónico: 2216-0116,
Universidad Mariana, San Juan de Pasto, Nariño, Colombia, 2015.

iguales o diferentes, todas estas transformaciones darán vida a nuevas variables y patrones de diseño, que al final del proyecto ayudaran a hacer el proceso más estándar para cada uno de los proyectos de software, las transformaciones modelo a modelo tienen el fin de llegar a una plataforma destino de programación, la cual se verá implementada por los desarrolladores especializados, que tendrán a su disposición un código básico generado por la interpretación de los modelos, diseñados por los interesados del proyecto.

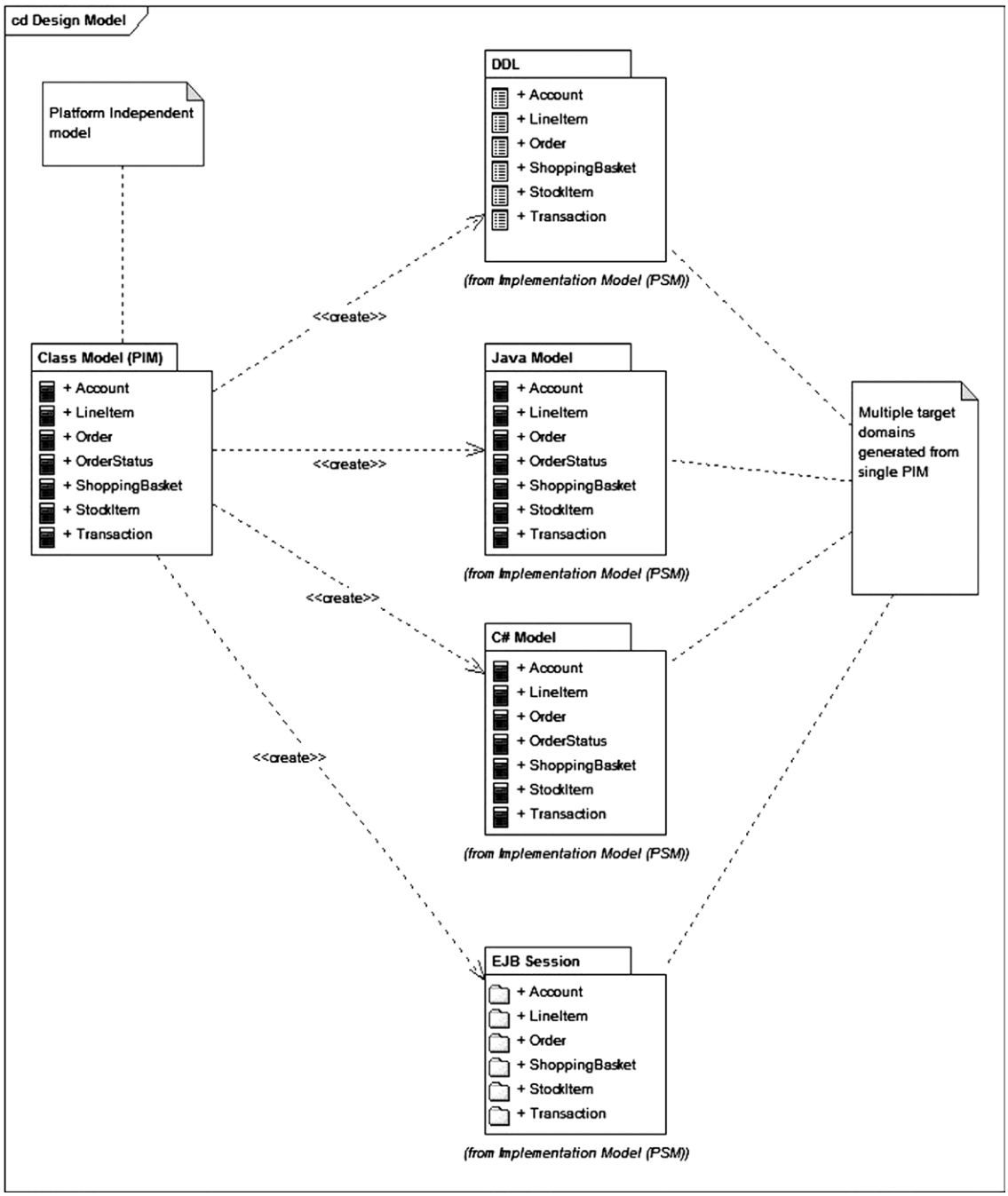


Figura 4. Representación de transformaciones en MDA.
Fuente: S.A., Sparx Systems Argentina - SOLUS (s.f.).

4. Herramientas con enfoque MDA

En el mercado existen diversos software especializados para modelar y generar código a partir de estos modelos, algunas herramientas son muy completas y poderosas, mientras otras no son tan amigables con el usuario y requieren de una escalabilidad de conocimiento para aprender a desarrollar en ellas; cada una de estas herramientas tiene ventajas una sobre otras, pero lo que se debe tener claro a la hora de utilizarlas es identificar qué se va hacer, con qué propósito y qué necesidades tiene la organización, para así adquirir la herramienta adecuada. Como se mencionó anteriormente a estas herramientas les falta un nivel de madures en lo que se refiere a código fuente generado, ya que es muy básico lo que estas entregan. A continuación se expondrán algunas herramientas que pueden ser utilizadas para desarrollos enfocados en MDA, pero no se hará una discusión de cuál herramienta es mejor, pues no es el objetivo de este artículo.

Eclipse: es un entorno de desarrollo para java, y se ha convertido en una herramienta que permite la creación de metadatos, además se pueden generar clases basadas en modelo de datos.

AndroMDA: es un software de código abierto, que puede tomar cualquier número de modelos y transformarlos en un "idioma" de cualquier lenguaje de programación, a través de procesos llamados paquetes o cartuchos, que permiten traducir modelos a código java, php, .net, entre otros (andromda, s.f.).

Enterprise Architect: es una herramienta basada principalmente en UML que puede hacer transformaciones entre modelo y modelo, además de modelo a código.

Model Driven Architecture (MDA): es una buena forma de "manejar la complejidad, alcanzar altos niveles de reutilización y reducir significativamente el esfuerzo de desarrollo necesario en los proyectos de desarrollo de software. Con el apoyo de la MDA incorporado, Enterprise Architect ayuda a cerrar la brecha entre el análisis y la aplicación" (S.A., Sparx Systems Argentina - SOLUS, s.f.).

5. Discusión

Al investigar y documentar este artículo se encontraron autores que proponen los esquemas MDA como una ventaja para una organización, ya que este esquema promueve en cada realización de un proyecto hacerlo de una forma separada pero coordinada con cada parte de su ciclo de vida; esto hace que un desarrollo común se vuelva en un desarrollo base para la realización de futuros proyectos, ya que desde un principio se están creando esquemas de trabajo que promueven buenas prácticas de ingeniería de software.

Ventajas:

- El MDA, permite que el modelo sea una representación de una parte de un procedimiento, estructura o del comportamiento de un sistema.
- El MDA en el proceso de elicitación, análisis e implementación de la variabilidad (y su trazabilidad) para un contexto de desarrollo de líneas de producto software.
- Más herramientas disponibles para su desarrollo (OptimalJ, Arcstyler, etc.) e integrando varias áreas de conocimiento en Ingeniería de Software.
- Es promovido por el OMG, que es un consorcio de las más importantes empresas de desarrollo de software.
- Proporciona un enfoque nuevo para las organizaciones dedicadas al desarrollo de software, brindándoles la posibilidad de estandarizar la mayoría de sus trabajos de desarrollo.

Desventajas:

- Está técnica está en un tiempo de crecimiento y desarrollo. Por eso la carencia en transformación completa y exacta de los modelos para desarrollo de software.
- Problemas relevantes para la reutilización y transformación de software en el contexto de MDA.
- Las herramientas que soportan MDA no son muchas, para realizar un análisis comparativo de la eficiencia de las mismas.
- Falta de evaluación de los cambios y el esfuerzo de adaptación requerido para hacerlo coincidir con los requisitos del sistema.

6. Conclusiones

En este artículo se logró tener una visión general de los principios, estándares y del mecanismo en los que está soportada la propuesta MDA. Una de las ideas que se consideran más relevantes en el enfoque MDA, es lograr una separación del modelado del negocio con la lógica del desarrollo del sistema, ya que esto permite que al momento de modelar algún sistema sea más sencillo para un ingeniero pensar de una forma más abstracta del mundo que lo rodea.

Con la ventaja de separar la lógica de desarrollo del negocio es más fácil reutilizar y adaptar diferentes negocios a un mismo modelo de desarrollo; esto agrega un valor muy importante a la hora del desarrollo de un software, viéndolo como una ventaja para cualquier organización dedicada al diseño de software de calidad. Por otra parte, es evidente que en la actualidad los gobiernos se están preocupando más por el campo de las TIC, esto refiere que a muchas empresas se les esté exigiendo una buena documentación del trabajo realizado a la hora de crear un software con el enfoque MDA, por ello, en cada paso de un modelo a otro, el arquitecto está asegurando que se debe tener buena documentación en cada una de las transiciones de los modelos.

7. Conflicto de intereses

Los autores de este artículo declaran no tener ningún tipo de conflicto de intereses del trabajo presentado.

Referencias

andromda. (s.f.). *andromda*. Recuperado de http://www.andromda.org/whatisit.html

Arando, F. (2006). Universidad Nacional de Colombia. En: *UN-Método para la elicitación de Requisitos de Software.* (5ta. ed.).

Group., O. M. (2015). Object Management Group. Recuperado de http://www.omg.org/.

Kleppe, A. (2014). MDA Explained. *The Model Driven Architecture™: Practice and Promise*. Addision-Wesley.

Object Computing Inc. (OCI). (2015). *OMG.* Recuperado de http://www.omg.org/mda/mda_files/OCI2004.pdf

Orozco, D., Giraldo, W. y Trefftz, H. (2013). MDE; MDA; Transformaciones y DSLs. Una breve introducción. Recuperado de https://repository.eafit.edu.co/bitstream/handle/10784/5107/Articulo8CCC.pdf?sequence=4&isAllowed=y

Papajorgi, P. y Pardalos, P. (2014). *Técnicas de ingeniería del software aplicada a los sistemas agrícolas [recurso electrónico]: un enfoque orientado a objetos y UML*. Viena-Austria.

Pressman, R. (1999.). *Ingeniería del Software un enfoque práctico* (5ta. ed.). McGraw-Hill.

S.A., Sparx Systems Argentina - SOLUS. (s.f.). *sparxsystems*. Recuperado de http://www.sparxsystems.com.ar/resources/mda/

Silega, N., Macias, D., Febles, J. y Noguera, M. (2014). Estado de la complejidad arbitraria y Arquitectura Dirigida por Modelos en el desarrollo de software en Cuba. *Revista Cubana de Ciencias Informaticas*, *8*(1), 156-171.

Texier, J., De Giusti, M. y Gordillo, S. (2013). *El desarrollo de software dirigido por modelos en los repositorios institucionales*. Medellín, Colombia: DYNA.

Vega, W. y Umaña, H. (2014). Diseño de Servicios Web Semanticos utilizando el desarrollo de software dirigido por modelos. *Ventana Informática,* (30), 97-108.

Zambrano, C., Varas, M. y Urrutia, A. (2012). Enfoque MDA para el diseño de un data warehouse difuso. *Ingeniare. Revista Chilena de Ingeniería*, *20*(1), 99-113.