

Revisión de literatura transformación de los modelos en MDD

Diana Carolina Cardona¹, Sergio Alexander Sierra²

¹Estudiante Especialización en Ingeniería de Software Universidad de Medellín, *carolinacardona1987@gmail.com*

²Estudiante Especialización en Ingeniería de Software Universidad de Medellín, *sergioalexandersierra@gmail.com*

Resumen

MDD (Model-Driven Development) es una metodología enfocada en la creación y transformación de modelos, teniendo como objetivo fundamental aumentar la productividad maximizando la compatibilidad entre diferentes sistemas, simplificando el proceso de diseño y donde se busca acabar con el paradigma que pretende hacer de los modelos solo documentación en muchos casos innecesaria y que aportan poco valor al producto final. El enfoque de desarrollo dirigido por modelos tiene dos componentes principales: los modelos y las transformaciones. Un modelo es una especificación formal de la función, estructura o comportamiento de una aplicación o sistema [1]. MDD Plantea una nueva forma de entender el desarrollo y mantenimiento de sistemas de software con el uso de modelos como principales artefactos del proceso de desarrollo. Los modelos se van generando desde los más abstractos a los más concretos a través de pasos de transformación y/o refinamientos, hasta llegar al código fuente.

Una transformación de modelos es entonces la generación automática de un modelo destino a partir de un modelo origen, de acuerdo a una especificación basada en un conjunto de reglas de transformación. Estas reglas describen cómo una o más construcciones del modelo origen son transformadas en una o más construcciones del modelo destino, y son ejecutadas por un motor de transformaciones.

Palabras clave: MDD (Model-Driven Development), Metodología, Desarrollo, Software, Documentación, Modelos, Origen, Destino, Reglas, Transformación, Código.

1 Introducción

El desarrollo de software basado en modelos (Model-Driven Development, MDD) es un nuevo enfoque de ingeniería de software donde los modelos cobran gran importancia y se convierten en la principal actividad de desarrollo. Su éxito depende fuertemente de la disponibilidad de lenguajes y adecuadas herramientas para la transformación de estos modelos y finalmente la generación automática de código fuente. Es aquí donde se hacen fuerte conceptos tales como PIM (Modelos independientes de la plataforma) y PSM (Modelos específicos de la plataforma).

Una transformación entre modelos puede orientarse a la definición básica de un sistema. "...un sistema es una unión de partes que funcionan de manera conjunta o independiente para lograr ciertos resultados o productos, basados en necesidades" [2], Donde el insumo inicial y el producto final es un modelo.

En este proceso de transformación de modelos, podemos encontrar en el nivel de abstracción más alto, el concepto de PIM. Este modelo define los conceptos sin incluir ningún detalle específico de la tecnología que se usará para implementar el sistema. Como siguiente paso, un PIM se transforma en uno o más PSMs, donde cada PSM representa la proyección del PIM de una plataforma específica donde estos son convertidos a código fuente. Debido a que los PSMs están orientados a un dominio tecnológico específico, esta transformación es bastante directa.

A diferencia del proceso de software tradicional, donde las transformaciones de modelo a modelo, o de modelo a código son realizadas con la intervención humana, en MDD se busca que estas transformaciones sean ejecutadas en lo posible por herramientas donde la participación de personas es mínima.

Este documento está basado en la búsqueda y estudio de material en donde se desarrollen temas similares al propuesto en este trabajo, con el fin de dar a conocer un punto de vista de lo que es MDD y como la transformación de modelos es esencial en todo este proceso. De igual forma se busca contextualizar un poco acerca de cómo los lenguajes en el cual los modeladores construyen el proyecto tienen a menudo carencias de definición y dificultades para realizar estas transformaciones.

El artículo está estructurado de la siguiente manera: En la sección 2 se presenta el método de trabajo donde se incluye la pregunta de investigación y los criterios de inclusión y exclusión. En la sección 3 se realiza la evaluación de los trabajos seleccionados. En la sección 4 se plantea la discusión de los resultados de trabajos encontrados con los propios. En la sección 5 se desarrollan las conclusiones obtenidas en el estudio de esta problemática. En la sección 6 se incluyen las referencias que sirvieron como apoyo para el desarrollo de este artículo.

2 Método

2.1 Preguntas de investigación

¿Cómo afecta la transformación de modelos a la productividad, mantenimiento y documentación de un sistema?

¿Existen herramientas que realmente soporten estas transformaciones?

¿Qué tan flexibles son estas transformaciones a los cambios tecnológicos?

¿Existe algún proceso que facilite la intervención de las transformaciones de los modelos en el desarrollo de software, reduciendo errores y automatizando el proceso, facilitando así la transformación de los modelos de PIM a PSM?

2.2 Criterios de inclusión y exclusión

Para la presentación de este documento de estado del arte se revisaran artículos comprendidos entre el año 2000 y 2013 donde nos enfocaremos en la revisión de título, resumen, introducción y conclusiones que nos permitan identificar información relevante acerca de:

- Relación entre modelos PIM y PSM
- Herramientas de modelado y/o interacción entre PIM y PSM
- Contextualización acerca de lo que es MDD y la transformación ente modelos.
- Transformación entre modelos (CIM, PIM, PSM, Código fuente).

Excluiremos aquellos artículos que consideremos no nos aportan la suficiente información para el estudio de la problemática planteada, también descartaremos los documentos que no cumplan con los criterios mencionados anteriormente.

2.3 Trabajos seleccionados

Tabla 1. Trabajos seleccionados

ID	Título	Autores	Fecha	Fuente
1	Behavior Model Mapping and Transformation using Model-Driven Architecture	Abdalla, Mohammed Mukhtar, Osman Abdullah, Azween Downe, Alan Giffin	2011	http://www.waset.org/journals/waset/v50/v50-89.pdf
2	Transformation From CIM to PIM Using Patterns and	<u>Kherraf, S.</u> Ecole de Technol. Super., Montreal <u>Lefebvre, E. ; Suryn, W.</u>	2008	http://ieeexplore.ieee.org/xpl/abstractAuthors.jsp?tp=&arnumber=4483222&url=http%3A

	Archetypes			%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4483222
3	Ingeniería dirigida por modelos	Trejo, J., & Robles, A	2010	http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/risi/2010n2/v7n2/a02v7n2.pdf
4	Proceso de Desarrollo de Software Mediante Herramientas MDA	López, E.D. González, M. López, M. Iduñate, E.L.	2007	http://www.iiisci.org/journal/CV\$/risi/pdfs/C476A1.pdf
5	Tools for Mapping Technique between PIM and PSM	Xiuhua Zhang.	2002	http://publications.nr.no/note_omni0302.pdf
6	Reflexiones acerca de la adopción de enfoques centrados en modelos en el desarrollo desoftware	<i>Juan Bernardo Quintero</i> <i>Jhon Freddy Duitama Muñoz</i>	2011	http://www.scielo.org.co/scielo.php?pid=S0123-21262011000100012&script=sci_arttext

3 Evaluación de los trabajos seleccionados

3.1 Criterios de evaluación

A continuación se enumeran los criterios de evaluación que se revisarán en los trabajos seleccionados de la sección 2.3:

1. Cómo impactan las reglas de transformación en la intervención de transformación de modelos de un PIM a PSM.
2. Se identifican las posibles dificultades de la transformación de modelos.
3. Se da una definición de como un modelo se transforma en otro modelo más específico.
4. Define el proceso y características de transformación de PIM a PSM.

5. Se definen algunos de los modelos que se identifican en MDD.

ESCALA DE EVALUACIÓN

A continuación se explicará la escala de evaluación de los artículos seleccionados en la sección 2.3, la calificación será cualitativa describiendo a continuación los calificativos de la evaluación:

1. Excelente: El artículo cumple completamente con el criterio de evaluación.
2. Bueno: El artículo cumple en gran parte con el criterio de evaluación.
3. Regular: El artículo cumple parcialmente con el criterio de evaluación.
4. Deficiente: El artículo no cumple con el criterio de evaluación.

3.2 Resultado de la evaluación

Tabla 2. Resumen de la evaluación

		Criterios de evaluación				
		Criterio 1	Criterio 2	Criterio 3	Criterio 4	Criterio 5
Trabajos	ID1	Bueno	Bueno	Excelente	Bueno	Excelente
	ID2	Regular	Regular	Excelente	Deficiente	Excelente
	ID3	Excelente	Regular	Bueno	Deficiente	Excelente
	ID4	Regular	Bueno	Bueno	Bueno	Excelente
	ID5	Regular	Deficiente	Regular	Bueno	Excelente
	ID6	Bueno	Regular	Deficiente	Regular	Regular

ID1: Behavior Model Mapping and Transformation using Model-Driven Architecture

Este artículo nos brinda un contenido poco descriptivo de la Arquitectura dirigida por modelos, se centra principalmente en la definición de los procesos de transformación de modelos y la creación del modelo de mapeo o modelo de asignación. Describe la importancia del mapeo en la transformación de modelos de PIM a PSM. Detalla minuciosamente las dos formas de realizar Mapping (vertical, horizontal) y el formalismo que usualmente se usa para transformar un modelo en otro modelo.

Además presentan dos propuestas para realizar el proceso del Mapping de manera directa e indirecta dado un formalismo a través de un caso explícito haciendo uso de UML. También hace referencia a algunos lenguajes utilizados para realizar las transformaciones.

Para finalizar identifica y hace énfasis en como el paradigma del desarrollador debe cambiar enfocándose en el análisis para la generación de modelos y no en el código fuente como tal.

ID2: Transformation From CIM to PIM Using Patterns and Archetypes

Este artículo nos brinda un contexto muy básico y poco descriptivo de la Arquitectura dirigida por modelos, se centra mayormente en la descripción de los procesos de transformación de modelos y la creación del modelo de asignación. Describe la importancia de la transformación de modelos de CIM a PIM. Detalla minuciosamente las formas de realizar transformaciones y el formalismo que usualmente se usa para transformar un modelo en otro modelo y su relación con los procesos de metamodelos. De manera poco detalla se describe el papel de UML para diagramar los procesos de negocio y las tareas de los usuarios, dando prioridad al diagrama de actividades detallado para especificar los requisitos del sistema. Identifica cuales son los roles de las personas que realizan el proceso de transformación del CIM a PIM.

Para finalizar se describe un estudio de un caso donde se ilustra el enfoque planteado, como se reutilizan los componentes, y como se mejora la eficiencia con el modelado.

ID3: Ingeniería dirigida por modelos

Este artículo contextualiza en gran medida la Ingeniería dirigida por modelos (IDM) y su estado hoy en día, las problemáticas que existen con la forma del desarrollo de software tradicional y una descripción poco profunda sobre los procesos de transformación. Su énfasis se da más que todo en la definición de la ingeniería dirigida por modelos y el uso de “Lenguajes de modelamiento de dominio específico” como soporte para la IDM y su necesidad de expresar conceptos de dominio efectiva, es decir, expresar de manera detallada los conceptos del mundo real y sus posibles variaciones para verse reflejadas enteramente en un modelo conceptual. Omite enteramente cualquier información referente al proceso de mapeo o modelo de asignación, reglas de transformación o automatización de este proceso.

ID4: Proceso de Desarrollo de Software Mediante Herramientas MDA

Este artículo hace referencia al desarrollo de software mediante herramientas MDA. Se identifica con precisión la deficiencia de los diferentes tipos de modelos y como cada modelo es utilizado para transformarse en otro modelo más específico o en su defecto en una transformación que resulta en la generación de código fuente. Seguidamente se realiza un estudio sobre las herramientas MDA, mostrando características, limitaciones y su aplicabilidad. Mostrando un panorama de lo que falta mejorar e implementar en ellas respecto a las transformaciones.

ID5: Tools for Mapping Technique between PIM and PSM

Este artículo es un documento muy bien estructurado que da a conocer cualquier elemento o proceso relacionado con la Arquitectura dirigida por modelos (MDA), su contextualización hoy en día y conceptos propios del MDA. También habla del proceso del MDA y detalla muy bien cada uno de sus componentes y procesos que intervienen, como lo son: Definición de CIM, PIM y PSM, Modelo de asignación o mapeo, reglas de transformación, lenguajes para realizar mapeo, ilustración de una transformación haciendo uso del UML, etc. Define y contextualiza un poco sobre las herramientas utilizadas en el proceso.

No trata minuciosamente las dificultades que existen en el modelo de asignación o mapping y tampoco propone mejoras para evitar dificultades en la creación de este modelo. Su enfoque se da más en la definición y detalle de todo lo relacionado con el MDA

Trabajo 6: Reflexiones acerca de la adopción de enfoques centrados en modelos en el desarrollo desoftware.

Este artículo nos deja ver claramente la definición de los mecanismos para utilizar modelos en la automatización de tareas del desarrollo de software. Se plantean diferentes definiciones sobre lo que es MDA, el planteamiento que se realiza sobre la transformación de modelos, las ventajas y desventajas en la ingeniería dirigida por modelos; las problemáticas en las transformaciones de los enfoques centrados en modelos. Los procesos de transformación que se realizan, dando inicio en los requisitos de usuario hasta convertirse en código fuente de una manera automatizada, pasando por PIM, PSM, código fuente y llegar a una solución satisfactoria para el usuario.

En toda esta definición del proceso de transformación también se hace un análisis sobre las herramientas y lenguajes que intervienen en MDA; se evalúan las características, ventajas y desventajas.

4 Discusión

El objetivo principal de este trabajo es identificar las posibles carencias que existen entre las transformaciones de modelos en MDD, principalmente en la transformación de PIM a PSM. Para este estudio se seleccionaron una serie de artículos los cuales fueron estudiados para encontrar información que nos apoyara en la revisión de esta problemática.

En la revisión del artículo Behavior Model Mapping and Transformation using Model-Driven Architecture encontramos información que nos da alguna claridad acerca de las transformaciones y refinamientos utilizados en estas.

La transformación de modelos es el proceso de convertir un modelo en otro para el mismo sistema. Se utiliza un lenguaje para describir dicha transformación [1]. Una transformación establece un conjunto de reglas que describen como un modelo expresado en un lenguaje origen puede ser transformado en un modelo en un lenguaje destino. Las categorías principales de transformación son los refinamientos (mappings) verticales y horizontales. Los refinamientos verticales relacionan modelos del sistema situados en distintos niveles de abstracción, como transformaciones de PIM a PSM ó de PSM a PIM. Los refinamientos horizontales, relacionan o integran modelos que cubren distintos aspectos en un mismo nivel de abstracción, mantienen la consistencia entre distintos niveles y garantizan que la información modelada sobre una entidad del sistema es consistente con lo que se dice sobre ella en cualquier otra especificación situada en el mismo nivel de abstracción [3].

Para realizar las transformaciones, OMG ha definido un estándar para la definición de transformaciones llamado QVT (Query/View/Transformation) [4].

El artículo Transformation From CIM to PIM Using Patterns and Archetypes en comparación con el estudiado anteriormente se centra más en las transformaciones de modelos CIM a PIM y el papel UML2 en este proceso. La

información que nos brinda este artículo es de poca ayuda con respecto al caso de estudio planteado por lo que no profundizaremos en su contenido.

Continuando con nuestro estudio, MDD se basa en estándares como UML(Unified Modeling Lenguaje), XMI (XML Metadata Interchange), MOF (Meta Object Facility) y CWM(Common Warehouse Metamodel) con el fin de obtener algunos beneficios fundamentales tales como Productividad, Portabilidad, Interoperabilidad Mantenimiento y documentación tal y como se describe en el artículo Proceso de Desarrollo de Software Mediante Herramientas MDA.

Para ayudar con esto MDE (Model-Driven Engineering) dice que la calidad de software puede ser comprobada y asegura con tres técnicas diferentes: validación de modelos, comprobación de modelos y pruebas basadas en el modelo [5].

En el artículo “Proceso de Desarrollo de Software Mediante Herramientas MDA.” Y complementando el artículo inmediatamente anterior se verifica como en el desarrollo de software dirigido por modelos las transformaciones de modelos son consideradas como activos importantes que deben ser manejadas con principios sólidos de ingeniería de software: estas transformaciones deben ser analizadas, diseñadas, implementadas, probadas, mantenidas y sujetas a la administración de configuración. Existe una considerable cantidad de trabajos que se están desarrollando alrededor de este enfoque y diversas herramientas que lo soportan, tanto comerciales como de open source. Estas herramientas realizan transformaciones de modelo a modelo o de modelo a código, pero ninguna de ellas toman en cuenta los mismos conceptos o características, además no consideran el proceso completo de MDA, es decir, sólo toman los modelos PIM y PSM pero no CIM.

Uno de los fundamentos que presenta el autor, es la separación que utilizan en el mapeo de PIM a PSM, refiriéndose a que PSM aborda tanto cuestiones tecnológicas como de negocio, por lo que se debe separar y a esto se le llama separación de aspectos, para que la parte del negocio no se disperse en la tecnología, sino, que el modelo PIM del negocio se enriquezca de la tecnología.

Refiriéndonos a esta transformación, en el artículo *Tools for Mapping Technique between PIM and PSM* nos presenta una relación muy similar del autor anterior en cuanto a la definición de cada modelo independiente, pero en la transformación de PIM a PSM nos ofrece la siguiente descripción. La forma en que un PIM podría ir directamente al código es por una asignación automática de un PIM a los estereotipos de un PSM y luego un generador de código que puede generar el código del PSM generado automáticamente. Puede haber algunas asignaciones predeterminadas definidas para PIM a PSM Corba o algunas de las otras plataformas, pero aún no para XML. PIM basado en UML ofrece el punto de partida para la transformación de modelos en el desarrollo de sistemas basados en MDA. Habrá cuatro tipos de transformación de modelos, a partir de los PIM a PSM y viceversa, de un PIM a otro, y de un PSM a otro.

Se verifica que cada elemento PIM se transforma en un PSM a través de unas reglas definidas y técnicas de mapeo, como lo cita [7], se utiliza el mapeo objeto-relacional (ORM) técnica de hibernate para hacer este tipo de objetos persistentes en la transformación.

En el artículo *Reflexiones acerca de la adopción de enfoques centrados en modelos en el desarrollo de software*, se enfatiza en la reutilización de modelos como estrategia en la agilidad de desarrollo, al punto de involucrar de forma exhaustiva el concepto de modelo y prometer que para construir una aplicación de software basta con construir modelos y transformarlos de forma semiautomática y asistida en el código de un sistema de información. Sin embargo, expresa todos los retos que se deben afrontar en la acogida de enfoques centrados en modelos, debido que las técnicas, lenguajes y herramientas no están unificados y esto dificulta a los equipos de desarrollo comenzar un proyecto con la certeza de estar utilizando la estrategia de transformación correcta.

El autor, apoya su trabajo en describir las falencias y problemas en la adopción de MDD, tales como, los modelos se vuelven obsoletos, los modelos no se pueden intercambiar fácilmente entre herramientas. Esto afecta en gran medida el ámbito

de las herramientas que apoyan a los enfoques centrados en modelo. De los artículos seleccionados, este es quizá el que describe de manera más descriptiva las ventajas y desventajas de MDD y como desde la definición hasta la transformación con las herramientas se logra un enfoque que nos ayude a identificar si esto es beneficioso para los proyectos que queremos afrontar.

En el artículo ingeniería dirigidas por modelo, aunque presenta un enfoque orientado a servicios (SOA), nos deja ver claramente la descripción de la metodología de MDD y las transformaciones que se realizan, hacen una descripción histórica del desarrollo dirigido por modelos, realizan una definición y los objetivos que se quieren lograr con esta metodología. Se identifica con el artículo citado anteriormente, en que describen los beneficios de la adopción de MDD en los proyectos, siendo la característica principal aumentar la productividad al máximo, la compatibilidad entre sistemas y simplificar el proceso.

5 Conclusiones

Esta sección tiene dos propósitos. El primero es resumir la labor que el trabajo descrito asumió, así como los resultados. La segunda es describir las implicaciones más relevantes de dichos resultados.

Durante este estudio nos encontramos con la dificultad de recopilar información que en verdad nos aportara con nuestra investigación, la mayoría de trabajos encontrados describían el tema a grandes rasgos sin profundizar en un determinado problema.

En la recopilación de esta información, encontramos que el modelado de sistemas es una poderosa técnica para la comprensión y diseño de solución de un problema determinado. La utilidad de esta metodología es la capacidad de llevar un problema complejo a una representación abstracta del mismo, descomponiéndolo en subsistemas más sencillos de afrontar, por esto es que el desarrollo dirigido por modelos plantea una nueva forma de entender el desarrollo y mantenimiento de sistemas de software con el uso de modelos como principales artefactos del

proceso de desarrollo. Por ellos y para lo que resaltamos en cada uno de los artículos, identificamos rasgos comunes.

Al realizar una descripción exhaustiva de los diferentes planteamientos de la ingeniería de modelos, resaltando los pro y contra de la esta adopción, donde se identifica como esta ingeniería ayuda a las empresas a obtener el control en la documentación y otros artefactos existentes, mejora la productividad y se mejora la mantenibilidad del desarrollo.

Se logra una mejor adopción de MDD según el nivel de madurez con que se adopte en los proyectos, se describen tres niveles de madurez y los beneficios adoptados en cada una. Desarrollo apoyado por modelos, desarrollo asistido por modelos y desarrollo dirigido por modelos.

6 Referencias

[1] OMG, "Mda guide version 1.0.1", Object Management Group, Tech. Rep., 2003

[2] O Johansen. Introducción a la teoría general de sistemas, 1982

[3] Moreno N., R.J., Romero R., Vallecillo A., Desarrollo de Software dirigido por modelos, in Fábricas de Software: experiencias, tecnologías y organización, Ra-Ma, 2007.

[4] OMG, Meta Object Facility (MOF) 2.0 Query/View/Transformation, v1.0, OMG, 2008

[5] Den Haan Joan (2009). <http://www.theenterprisearchitected.eu>: "MDE - Model Driven Engineering - reference guide"

[6] Aguilar, J... (2011). Una Introducción al Desarrollo de Software Dirigido por Modelos (DSDM). University of Alicante, Spain. Retrieved from <http://info.maz.uasnet.mx/dsdm/materiales/Bloque I/Una Introducción al Desarrollo de Software Dirigido por Modelos.pdf>

[7]http://mermaid.econ.kuleuven.be/publications%5C2006_3M4MDA_event_driven_mda_transformations.pdf