Scientia Et Technica
ISSN: 0122-1701
scientia@utp.edu.co
Universidad Tecnológica de Pereira
Colombia

Pascuas Rengifo, Yois Smith; Mendoza Suarez, Jefersson Andres; Córdoba Correa, Eliana Dirley

Desarrollo Dirigido por Modelos (MDD) en el Contexto Educativo

Scientia Et Technica, vol. 20, núm. 2, junio, 2015, pp. 172-181
Universidad Tecnológica de Pereira
Pereira, Colombia

Disponible en: http://www.redalyc.org/articulo.oa?id=84942286012

Desarrollo Dirigido por Modelos (MDD) en el Contexto Educativo

Model-Driven Development (MDD) in the Educational Context

Yois Smith Pascuas Rengifo[1], Jefersson Andres Mendoza Suarez[1], Eliana Dirley Córdoba Correa[1]
[1]*Ingeniería de Sistemas, Universidad de la Amazonia, Florencia – Caquetá, Colombia*
el.cordoba@udla.edu.co
y.pascuas@udla.edu.co
je.mendoza@udla.edu.co

Resumen **- El desarrollo dirigido por modelos (MDD por sus siglas en inglés) es un paradigma de desarrollo de software basado en modelos, aplicados en diferentes ámbitos; en este artículo se presenta una descripción actual del MDD, se tiene en cuenta sus ventajas, los puntos claves de la iniciativa, el ciclo de vida y las transformaciones que realiza. Se estudian proyectos que practican este paradigma especialmente en el contexto educativo y artículos que exponen el impacto que ocasiona el adoptarlo, de igual manera se exponen los beneficios de cada proyecto que ponen en práctica éste paradigma y herramientas que lo soportan.**

Palabras Claves: Código, Educación en ingeniería, Gestión de desarrollo de software, Metamodelado, Transformación.

Abstract:- The model-driven development (MDD) it's a software paradigm development based on models used in different fields; this article present a current description about the MDD, one considers its advantages, the key points of the initiative, life cycle and the transformations it makes. Are studied projects that practice this paradigm especially in the educational context and articles that expose the impact that causes adopt it, just as the benefits of each project that implement this paradigm and its tools that support it.

Keywords: Code, Education in engineering, Software development management, Metamodeling, Transformation.

I. INTRODUCCIÓN:

Los procesos de MDD utilizan como columna principal el trabajo con modelos para la ejecución del desarrollo de software, con el propósito de interponer un elemento que ayude a la interrelación entre la propuesta tradicional que se basa en lenguajes de programación y plataformas de objetos o componentes de software**.**

El principal propósito que cumple el MDD es tratar de minimizar los costes y tiempo de desarrollo de las aplicaciones software, en la búsqueda de mejorar la calidad de las aplicaciones, independiente de la plataforma donde el software se ejecuta, y garantice las inversiones en tecnología. Las transformaciones que realiza entre modelos y los lenguajes específicos del dominio, se resaltan sus ventajas, principales metas y algunos riesgos. Además, se analizan los antecedentes, productos que emplean el MDD especialmente en el contexto educativo y algunos otros dominios que lo aplican.

Para abordar la temática de este artículo en cuanto al MDD en el contexto educativo se consultaron diferentes bases de datos especializadas como la IEEE Xplore, ACM y Google Scholar. De los artículos encontrados se seleccionaron los artículos que tratan especialmente del tema, además se seleccionaron trabajos que exponen el impacto de la adopción del MDD y artículos de estudios de algunos dominios que emplean el MDD. Para abordar el tema de MDD en el contexto educativo, se requirió responder a las siguientes preguntas: ¿Qué es MDD y cuáles son sus principales características?, ¿Qué impacto ocasiona el MDD al ser adoptado en las industrias?, ¿Qué beneficios trae el MDD al ser adoptado en el contexto educativo?, ¿Qué dominios emplean el uso del MDD? Con base en estas preguntas se organizó el artículo de la siguiente manera en la sección II se encuentra la parte de la contextualización y significados de los términos concernidos con el tema del MDD, en la sección III se presentan las discusiones es decir un resumen más detallado de los trabajos seleccionados que ponen en práctica el paradigma MDD en particular el contexto educativo y por último en la sección IV las conclusiones.

II. CONTEXTUALIZACIÓN

A. Desarrollo Dirigido por modelos (MDD):

El MDD según [1], [2], [3], [4] y en [5] coinciden que es un paradigma emergente que resuelve numerosos problemas

Fecha de Recepción: 11 de Mayo de 2014
Fecha de Aceptación: 25 de Junio de 2015

asociados con la composición e integración de sistemas a gran escala basado en el uso de modelos, soportado por potentes herramientas que tienen como objetivo reducir el tiempo de desarrollo y mejorar la calidad de los productos, separando el diseño de la arquitectura. Con el objetivo principal de permitir aumentar la productividad y reducir los costes del desarrollo.

Los modelos según [6] los autores definen que estos se van generando desde la parte más abstracta a lo más concreto, a través de transformaciones y/o refinamientos. Los puntos claves de la iniciativa del MDD según [7] y en [8] son: La abstracción, automatización y estándares, trayendo consigo beneficios como la adaptación de los cambios tecnológicos, requisitos, re-uso y mejora la comunicación tanto para los usuarios como para los desarrolladores.

Pons, Giandini y Pérez en [1], presentan que las propuestas concretas más conocidas y utilizadas en el ámbito de MDD son, por un lado MDA desarrollada por el OMG (Object Management Group) y por otro lado el modelado específico del domino (DSM) acompañado por los lenguajes específicos del dominio (DSL), ambas iniciativas guardan naturalmente una fuerte conexión con los conceptos básicos de MDD, es decir los principios fundamentales del MDD no constituyen realmente nuevas ideas sino que son reformulaciones y asociaciones de ideas anteriores.

En [1] y en [9] los autores definen que la Arquitectura Dirigida por Modelos (MDA por sus siglas en inglés), es una arquitectura que proporciona un conjunto de guías para estructurar especificaciones expresadas como modelos, siguiendo el proceso MDD, es decir MDA se refiere a las actividades que llevan a cabo los desarrolladores, mientras que el MDD hace referencia a su definición formal según [10] y [11].

El OMG mantiene la marca registrada sobre MDA, así como sobre varios términos similares incluyendo MDD. El acrónimo principal que aún no ha sido registrado por el OMG hasta el presente es MDE, que significa Model-Driven software Engineering (Ingeniería de software dirigida por modelos), consecuencia de esto, el acrónimo MDE es usado actualmente por la comunidad investigadora internacional cuando se refieren a ideas relacionadas con la ingeniería de modelos sin centrarse exclusivamente en los estándares del OMG, es decir que el MDE utiliza conceptos del MDD. En la Figura 1 se representa la relación entre el MDD, MDA y MDE.

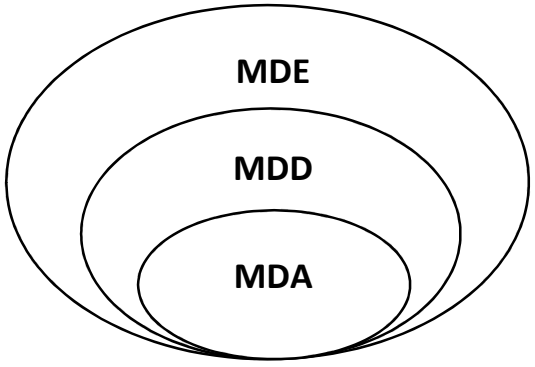

Fig. 1. Relación entre MDD con MDA y MDE [12].

B. Ciclo de Vida:

El ciclo de vida muestra los tipos de artefactos que se crean durante el proceso de desarrollo, estos son modelos formales, es decir, modelos que pueden ser comprendidos por computadores [12]. Según [6] y [13] los modelos son una representación conceptual o física a escala de un proceso o sistema, con el fin de analizar su naturaleza, desarrollar o comprobar hipótesis y permitir una mejor comprensión del fenómeno real al cual el modelo representa, perfeccionando los diseños, antes de iniciar la construcción de las obras u objetos reales [1].

Las cualidades que los modelos poseen según [14] son: comprensibilidad, precisión, consistencia, completitud, flexibilidad y re-usabilidad.

El MDD identifica distintos tipos de modelos: modelos con alto nivel de abstracción, según [15] son modelos que elevan el nivel de desarrollo del software y reducen las diferencias entre los dominios de la tecnología; es decir modelos independientes de cualquier metodología computacional, llamados CIM (Computational Independent Model), los modelos independientes de cualquier tecnología de implementación llamados PIM (Platform Independent Model), modelos que especifican el sistema en términos de construcciones de implementación disponibles en alguna tecnología específica, conocidos como PSM (Platform Specific Model), y los Modelos que representan el código fuente.

Según [1], [16] y [17] lo novedoso que propone MDD es que las transformaciones entre modelos, es decir, de un PIM a PSM sean automatizadas. Los tres pasos principales en el proceso de desarrollo de MDD se ilustran en la figura 2.

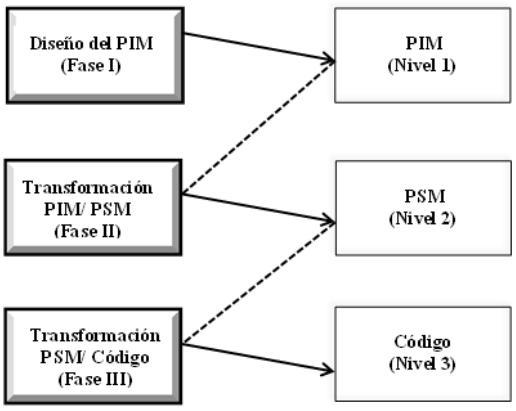

Fig. 2 Pasos en el proceso del desarrollo del MDD.

C. Metamodelo

El metamodelo es un mecanismo que permite definir formalmente lenguajes de modelado sin ambigüedades para que una herramienta de transformación pueda leer, escribir y

entender los modelos, el metamodelo en sí es un modelo, y se utiliza para precisar cómo se traducen los pensamientos en palabras. El metamodelo describe la sintaxis abstracta del lenguaje. Esta sintaxis es la base para el procesamiento automatizado de los modelos [1].

D. Transformaciones

Las transformaciones según [8] y [18] son el proceso en donde se toma un modelo de entrada y se produce otro modelo como salida. Las transformaciones son esenciales en el proceso de MDD.

En este contexto, la transformación de metamodelos es una actividad central para el manejo de modelos, donde su objetivo es especificar la manera de producir una serie de modelos de destino en función de un conjunto de modelos de origen. La transformación de un modelo tiene que definir la forma de generar un modelo destino (Mb), conforme a un metamodelo destino (MMb), de un modelo origen (Ma), acorde con un metamodelo origen (MMa), en la figura 3 se muestra de manera general los procesos de transformaciones de metamodelos [19].

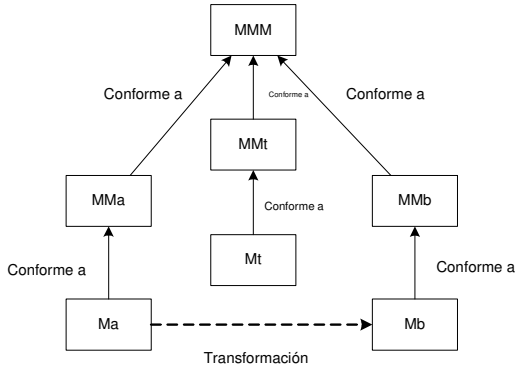

Fig. 3 Procesos de transformaciones de metamodelos [19]

E. Arquitectura de 4 capas de modelado del OMG:

La Arquitectura de cuatro capas de modelado es la propuesta del OMG orientada a estandarizar conceptos relacionados al modelado, desde los más abstractos a los más concretos. Los cuatro niveles definidos en esta arquitectura se denominan: M3-Meta-Metamodelo, M2-Metamodelo, M1- modelo del sistema e instancias, como se muestra en la figura 4.

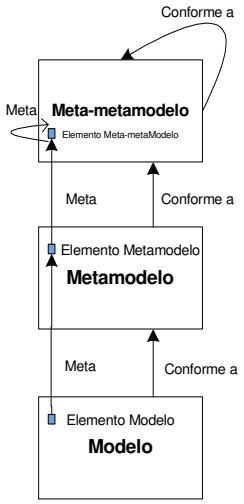

Fig.4 Modelo de la Arquitectura de 3 capas del Modelado [19]

F. Herramientas para la creación de modelos de software y para la generación de código.

Actualmente existen numerosas herramientas para trabajar con MDA y en la mayoría de casos permiten la generación de código para diferentes plataformas, las herramientas más utilizadas son: Enterprise Architect (EA) soporta transformaciones de MDA usando plantillas con transformaciones fáciles de editar y desarrollar [20], y Eclipse según [21] y en [22] lo definen como una plataforma abierta de libre distribución, creada y soportada por una gran comunidad de desarrolladores que han incorporado herramientas para dar apoyo en cada una de la fases del ciclo de vida del enfoque de MDD [22]. En cuanto a las herramientas para la generación de código se encuentran: JET, Acceleo, Xpand, Apache Velocity Project, Modeling SDK for Visual Studio y MOFScript, siendo esta última la herramienta más utilizada.

MOFScript es una de las herramientas para la transformación de modelo a texto, que permite apoyar la generación de código de una aplicación o la documentación de los modelos. MOFScript es un lenguaje de transformación presentado por la OMG [23].

III. DISCUSIÓN

Se revisaron trabajos y proyectos que aplican el desarrollo de Software dirigido por modelos en el contexto educativo, donde se identifican las características, finalidad principal de cada proyecto y los beneficios que se obtienen al utilizar MDD en el desarrollo del producto. El análisis se realiza teniendo en cuenta en primer lugar los antecedentes del paradigma MDD, en segundo lugar los trabajos que aplican el MDD especialmente en el contexto educativo, en tercer lugar trabajos que muestran el impacto que ocasiona la adopción de este paradigma, se sigue con trabajos que se pueden implementar en un futuro en el entorno de la educación y finalmente se hablan de dominios que implementan el MDD.

A. Antecedentes:

Durante la historia, el proceso de desarrollo de software ha resultado costoso, riesgoso, incierto y demasiado lento para las condiciones de negocio actual.

El problema que siempre ha permanecido es la complejidad, la no reutilización y poca expresividad en los lenguajes de programación. Después de muchos años de innovación y de intentos, aparece un proceso de producción de software soportado por modelos conceptuales y dirigidos por transformaciones.

A lo largo de estos años se ha visto surgir el Desarrollo de Software Dirigido por Modelos (MDD), como una nueva área dentro del campo de la ingeniería de software. 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.

Según Pons, Giandini y Pérez en [1], el MDD es una disciplina que está generando expectativas como alternativa sobresaliente a los métodos convencionales de producción de software, más orientado al espacio de la solución que al espacio del problema.

B. MDD en el contexto educativo:

Dentro de los trabajos analizados en el contexto educativo se encuentra [24] en donde los autores presentan un estudio con base en los laboratorios virtuales (LV); como anteriormente en las prácticas de laboratorios, se trabajaba bajo sistemas análogos, existían pocas partes electrónicas con software, talento humano y espacio restringido; debido a la cantidad de estudiantes y problemas de costos, en [25] se plantea que los LV surgen básicamente, por la necesidad de crear herramientas de apoyo al estudiante para sus prácticas de laboratorio, con el objetivo de optimizar diferentes aspectos relacionados con el aprendizaje y la investigación.
Pascuas, Bocanegra, Ortiz y Pérez en [24] proponen un metamodelo origen denominado Experimento, que muestra la estructura y funcionalidades de los LV y un metamodelo destino llamado Laboratorio Virtual, que permite que la aplicación sea desarrollada en cualquier plataforma. Posteriormente se presenta una serie de transformaciones de esos modelos a código para que sean interoperables con el estándar SCORM, como lo son: Transformaciones modelo a modelo utilizando ATL y transformaciones modelo a texto con MOFScript para generar código y empaquetarlo con SCORM.

En [26], Texier, Giusti, Oviedo, Villarreal y Lira definen que los repositorios institucionales (RI) son estructuras web interoperables que permiten depositar recursos académicos y científicos descritos por medio de un conjunto de datos específicos (metadatos); estos RI tienen como propósito recopilar, catalogar, gestionar, acceder, difundir y preservar. Los beneficios que proporcionan el MDD en los RI es que pueden ser utilizados en diferentes procesos presentes en este, proporcionando a los desarrolladores tanto los conceptos del dominio como lenguajes de modelado sin ambigüedades, para poder realizar transformaciones de modelos en un lenguaje fuente a un modelo en un lenguaje destino.

Montenegro, Cueva, Sanjuán, y Gaona en [27], dan a conocer cómo los sistemas de gestión de aprendizaje (LMS) presentan el problema de interoperabilidad entre plataformas, si este problema no existiera se podría migrar un curso que está desplegado sobre la plataforma LMS-A a la plataforma LMS-B manteniendo todos sus contenidos y funcionalidades, pero como esto no es posible, ya que cada LMS está construido bajo un estándar diferente o si están bajo el mismo estándar cada una hace su propia interpretación; para dar solución a este problema se presenta la creación de *KiwiDSM:* herramienta de lenguaje de dominio específico (DSL), que apoyada en la ingeniería dirigida por modelos (MDE), permite modelar módulos que conforman un sistema de gestión del aprendizaje (LMS) en el área de comunicaciones para así mismo proporcionar beneficios ya que al trabajar con MDD se reduce el tiempo y esfuerzo en la creación y despliegue de los módulos y modelados creando el metamodelo compatible con los requerimientos de dicho LMS.

De Souza, Castro-Filo y Andrade en [28], presentan estudios relacionados con la adaptación de la tecnología digital y los recursos educativos, proponen procesos automáticos de personalización, tomando en consideración el perfil de los estudiantes, generalmente determinado por las evaluaciones. Utilizan los objetos de aprendizaje (LO) como recursos digitales desarrollados para ayudar a los maestros a poder presentar conceptos pedagógicos para los estudiantes. Sin embargo, la forma en que se producen estos recursos permitendo a los maestros la adaptación a la realidad de sus estudiantes. Por lo tanto, este trabajo propone un proceso de desarrollo de objetos de aprendizaje personalizables utilizando una estrategia basada en modelos. Sobre la base de este enfoque, los maestros harán ajustes a LO y así lograr una mayor autonomía en el uso de estos recursos.

Los Mecanismos gráficos de consultas para el almacén de datos históricos (Data Warehouse) dentro del MDD, como la especificación de las consultas sobre una estructura de base de datos representa un gran reto para los usuarios no técnicos, la forma tradicional de acceder a una base de datos ha sido por medio de SQL. Debido a la complejidad de la formulación de búsquedas no triviales, el objetivo es hacer más accesibles a una gama más amplia, estableciendo el uso de lenguajes gráficos para que facilite al usuario final la especificación de consultas [29]. Las consultas se hacen aún más complicadas cuando los datos son almacenados en estructuras complejas tales como base de datos históricos (DW). Una manera de abordar esta complejidad es a través del uso del MDD. El objetivo que plantean Neil, Irazábal, Vincenzi y Pons en [29] es proporcionar una simplificación en el mecanismo de búsqueda de información en la DW, donde los usuarios pueden llevar a cabo las consultas por medio del uso de un entorno gráfico, sin tener en cuenta cualquier detalle de implementación.

En [30], Teruaki K., Takao F., Seiko A. y Shin K presentan el desarrollo de un programa de educación utilizando MDD para los principiantes, donde el objetivo es enseñar el modelado con MDD, ya que permite a los estudiantes adquirir habilidades de modelado en un corto período de tiempo. El primer concepto utilizado por este programa es "la educación de modelado de software utilizando MDD", que permite a los estudiantes repetir el modelo de refinamiento en un corto período de tiempo. El segundo concepto utilizado por este programa es "la educación en espiral en las técnicas fundamentales y la experiencia del desarrollo", que permite a los estudiantes aprender técnicas fundamentales y comprender los métodos que utilizan estas técnicas fundamentales.

Francia R., Bierman J. y Cheng B. en [31], proponen el repositorio del MDD (ReMoDD), proyecto que tiene que ver con el desarrollo de un repositorio que contenga artefactos que apoyan la investigación y la educación en el desarrollo dirigido por modelos (MDD). La plataforma ReMoDD proporciona interfaces y el intercambio de mecanismos que permitan una variedad de herramientas para recuperar artefactos del repositorio y presentar artefactos candidatos al repositorio. ReMoDD incluirán casos de estudios, documentados MDD, ejemplos de modelos que reflejan las prácticas de modelado lo bueno y malo, modelos de referencia (incluyendo metamodelos) que se pueden utilizar como base para la comparación y evaluación de técnicas de MDD, modelos genéricos y transformación que reflejan la experiencia de modelado reutilizable, descripciones de modelar técnicas, prácticas y experiencias, y modelando ejercicios y problemas que se pueden utilizar para desarrollar las tareas y proyectos de aula.

En [32], los autores sostienen que el área de investigación del desarrollo basado en modelos es probable que conduzca en poco tiempo a las herramientas que permitirá automatizar la mayor parte de la actividad de la construcción en procesos de desarrollo de software. Por lo tanto, analizaron el impacto que dicha evolución tendría en el plan de estudios de ingeniería de software, tanto en términos del cuerpo de conocimientos y las estructuras de los cursos, recurriendo a analogías con el impacto durante la década de 1960 del desarrollo de los lenguajes de programación orientados a problemas y así poder compiladores para ellos; con el objetivo de implementar los componentes de los sistemas del desarrollo de software en la parte del análisis y diseño para reutilizar dichos elementos en otros desarrollos, estas señales se originan en el trabajo de investigación sobre el desarrollo dirigido por modelos, que tiene como objetivo automatizar el proceso de diseños que es transformado en el código del programa.

En [33], Hamou-Lhadj, Gherbi, Nandigam presentan que la idea básica de MDA es un desplazamiento del enfoque de desarrollo a partir del código al modelo. Por otra parte, el OMG apoya la MDA con un conjunto de normas. Estos incluyen el Meta Object Facility (MOF), el Lenguaje Unificado de Modelado (UML), la Object Constraint Language (OCL), el XML Metadata Interchange (XMI), etc. Estos estándares introducen un importante cuerpo de conocimientos que debe integrarse en un curso de ingeniería de software. Además, la MDA introduce un cambio de código; tales cambios tienen un impacto en enseñanza de la ingeniería del software en general. En este trabajo se han introducido de manera sucinta el cuerpo de conocimiento implícito en la MDA y sus asociados normas y discutieron el impacto de la MDA en ingeniería de software en general basada en una experiencia en la enseñanza de un curso de posgrado sobre el tema, para presentar y discutir los conceptos y permitiendo tecnologías del enfoque basado en modelos para ingeniería de software (basado en normas MDA) que deberían estar cubiertos en un curso sobre el tema. Por lo tanto, también se discute las dificultades encontradas por los estudiantes cuando el aprendizaje de estos conceptos, así como sus preocupaciones con respecto a cuánto deben conocer la pendiente de la curva de aprendizaje. Dado que las empresas confían en las soluciones de software para preservar su posición en un mercado altamente competitivo, la necesidad para los sistemas de software confiable y robusto es vital. Últimamente, ha habido un gran interés en la construcción software utilizando modelos como sus principales artefactos. A diferencia de técnicas tradicionales de desarrollo que tienden a ser, modelo impulsado por los enfoques centrados en código, como el MDA.

C. Impacto de la adopción del MDD

Con respecto al impacto que ocasiona la adopción del paradigma del MDD a continuación se citan los siguientes trabajos que estudian este caso en particular.

En [43] y en [44], los autores presentan el impacto que ocasiona el desarrollo dirigido por modelos en la Arquitectura de Procesos de Software, ya que el MDD siendo un enfoque de desarrollo de software cada vez más popular aun en la práctica industrial, no está claro. El objetivo del trabajo es entender cómo la adopción de herramientas y técnicas MDD afecta el proceso arquitectónico, el papel y las responsabilidades del arquitecto de software. Cuando se aplica el MDD, el proceso arquitectónico según en [45], es más estructurado y riguroso, el papel del arquitecto en MDD es más amplio y más exigente, es decir debe convertirse en un experto en el manejo de esta herramienta. El arquitecto de software debe evangelizar MDD y continuamente comunicar el valor añadido de MDD para cada una de las funciones de los miembros de los equipos involucrados. Con la adopción de herramientas y técnicas MDD se exige un cambio en los límites entre los roles tradicionales en el proceso de desarrollo de software.

El trabajo [46] y [47] los autores dan a conocer los desafíos a la hora de implementar y adoptar el desarrollo dirigido por modelos, puesto que se necesita un apoyo eficaz para este proceso en el desarrollo del modelo y las relaciones entre los modelos; es por eso que se hace un análisis sobre el uso del MDD en un contexto empresarial. Los resultados confirman que se presentan "desafíos" relacionados con el uso del MDD cuando este es implementado. Aun con las grandes ventajas que éste proporciona según en [40], el apoyo al proceso MDD todavía no está bien establecido, hay información dispersa sobre el existente modelo de procesos, y se encuentra muy poco material publicado sobre su uso y las empresas se enfrentan a retos cuando adoptan este nuevo paradigma de MDD.

Solberg, Simmonds, Reddy, Ghosh y France en [48] y Fuentes y Sánchez en [49] dan a conocer una técnica orientada a aspectos para apoyar la separación de las preocupaciones en el desarrollo dirigido por modelos; como la gestión de relaciones y la especificación de transformaciones entre modelos en los diversos niveles de abstracción son tareas complejas; los modelos de sistemas enredados con inquietudes como la seguridad hacen difícil el desarrollo de sistemas complejos. El MDD que utiliza técnicas orientadas a aspectos, facilita la separación de las preocupaciones, ya que el uso del marco

simplificará tanto la tarea de desarrollo del modelo y la tarea de especificar transformaciones.

Un mecanismo para la separación de las preocupaciones en el mismo nivel de abstracción es modelar el sistema con vistas. En donde una vista del sistema describe una cierta faceta del sistema. La separación de las preocupaciones es reconocida como un principio clave para hacer frente a la complejidad en el desarrollo de software.

D. Trabajo futuro en el contexto educativo

A continuación se citan trabajos futuros que se pueden implementar en el contexto educativo:

Se desarrolla un trabajo orientado a proponer un enfoque MDD para la integración de tiempo real del sistema operativo específico (RTOS) en el modelo de diseño en tiempo real válido [34], como la transición desde el modelo de diseño para el modelo de implementación es una fase crítica en el proceso de desarrollo de sistemas embebidos en tiempo real, se propone una solución que va más hacia un aumento en el nivel de abstracción utilizando el modelo de desarrollo impulsado por modelos (MDD) con el fin de superar la creciente complejidad de tiempo real de sistemas embebidos (Rte), MDD introduce modelos intermedios a partir del modelo de la especificación y saliendo con el modelo de la aplicación, mientras que pasa a través del modelo de diseño; este trabajo, propone un enfoque de dos pasos sobre la base de una descripción explícita de dos tipos de plataforma: la plataforma abstracta que se utiliza a nivel de diseño para validar las diferentes opciones de diseño, y la plataforma de ejecución concreta, la primera consiste en las pruebas de viabilidad, cuya función es ayudar al diseñador detectar los posibles problemas de refinamiento y la segunda consiste en el mapeo que asegura el cumplimiento del modelo de aplicación.
Con el fin de reducir los costes de desarrollo, un principal objetivo del enfoque MDD, éste se aplica en el desafío de automatizar una correcta transición desde el diseño a los modelos de implementación, que conserve las propiedades de tiempo, realizando una validación de temporización en donde a nivel de especificación se puede aplicar un análisis de los presupuestos de tiempo, a nivel de diseño, análisis de programación y el análisis de rendimiento. Para hacer frente a este problema, se propone añadir un paso de pruebas de viabilidad y un paso de correspondencia entre el diseño y las fases de ejecución.
La metodología óptima introduce temporización de validación del nivel de especificación con el fin de orientar el diseño del modelo de concurrencia que satisface las limitaciones de tiempo.

Como los modelos son un artefacto clave para la generación de aplicaciones de software según [35], a estos se les debe evaluar su calidad y para ello la detección de defectos es considerado un enfoque adecuado, ya que proporciona un alto nivel de validez empírica, que es proporcionada por la variedad de modelos que se observan, y aun la herramienta más utilizada para el modelado que es UML no está permitiendo una especificación completa de la funcionalidad de las aplicaciones y ha presentado ciertas falencias a la hora de implementar modelos que se aplican en el MDD [36] y para evitar esto, el uso de una sola técnica para encontrar defectos no garantiza que todos los defectos sean encontrados, es recomendable el uso de varias técnicas de verificación con el fin de encontrar el mayor número posible de defectos. Para esto en [37], los autores presentan un caso de estudio para evaluar la utilidad de un procedimiento de medición de tamaño funcional (FSM) para detectar defectos en los modelos de un entorno MDD; un procedimiento FSM se puede utilizar para identificar defectos en los modelos conceptuales de un enfoque MDD porque analiza todas las construcciones conceptuales que participan en la funcionalidad del sistema.

El caso de estudio que presentan Marín, Giachetti, Pastor, Vos y Abran en [37], corresponde a un estudio exploratorio sobre los defectos y los tipos de defectos que se encuentran en los modelos OO-Method (método orientado a objetos que pone la tecnología MDD en práctica) por un equipo de inspección y por el procedimiento OOmCFP (OO-Método puntos función cósmica), el objetivo de este trabajo es comparar estos defectos y tipos de defectos con el fin de evaluar la utilidad de OOmCFP para detectar defectos.
El OOmCFP es un sistema de medición automatizado que implementa un estándar ISO para la medida del tamaño funcional, así como un procedimiento FSM definido para aplicar esta norma con el modelo conceptual de un enfoque MDD. En el caso de estudio que se analizó dio como resultado que el FSM es útil ya que encontró todos los defectos relacionados con un tipo de defecto y también encontró diferentes tipos de defectos que un equipo de inspección no identificó. Sin embargo, un equipo de inspección también es necesario para encontrar defectos que el FSM no puede encontrar. Así, la combinación de estas dos técnicas es un enfoque interesante para evaluar la calidad de los modelos conceptuales.

Streitferdt, Wendt, Nenninger, Nyßen y Lichter en [38], presentan el dominio de la automatización, el cual es un enfoque nuevo, que trata de la utilización de desarrollo dirigido por modelos, el cual propone el uso de éstos a diferentes niveles de abstracción y realiza transformaciones entre ellos, con el fin de derivar una implementación de la aplicación, utilizándolos para mejorar la comunicación y la interacción de diferentes expertos en el dominio ya que poseen una semántica definida y reducen los malentendidos comunes. Además, el MDD promete los siguientes beneficios: ahorro de tiempo de generación de código y mejora en la calidad.

En el proyecto [39] los autores plantean un enfoque innovador para la realización de un marco de desarrollo dirigido por modelos que se basa en el modelado de las actividades de aprendizaje adaptativo conscientes dentro de los entornos de aprendizaje sensibles al contexto. Su elemento central consiste en un lenguaje de modelado visual de dominio específico llamado CAAML (lenguaje de modelado de actividades de adaptación consciente del contexto); el objetivo

es apoyar a los diseñadores pedagógicos para modelar las actividades de aprendizaje adaptativo sensibles al contexto y transformarlos en modelos ejecutables representados en IMS Learning Design (IMS-LD), que es un lenguaje de modelado educativo que tiene como objetivo definir formalmente una estructura semántica para anotar los procesos de enseñanza y aprendizaje, y así convertirlos en entidades reutilizables entre diferentes cursos y aplicaciones.

Por esto se implementa CONTACT-Me (contexto y actividad modelador de adaptación para ambientes de aprendizaje flexible) la cual es una herramienta de creación basada en el lenguaje CAAML [40], que es un lenguaje de modelado visual de dominio específico y la cual es creada para alcanzar el objetivo establecido. Con el fin de garantizar la interoperabilidad de las actividades diseñadas a través de diferentes plataformas de aprendizaje, esta herramienta transforma los modelos representados en el lenguaje CAAML en ejecutable modelos representados en IMS-LD. De este modo la complejidad de IMS-LD está oculta por el uso de los conceptos relacionados con contextos sensibles. CONTACT-Me está basada en un enfoque MDD, ya que este permite reutilización y mejoras en el sistema.

El trabajo de desarrollo de software semiautomático basado en MDD para Robots heterogéneos multi-articulares en [41], los autores lo proponen debido a que puede resultar difícil de reutilizar software embebido, como el modelo, diseño y código, para los robots de múltiples conjuntos heterogéneos, en este trabajo se propone el mecanismo de desarrollo de robot de múltiples articulaciones y el desarrollo de software semi-automático basado en MDD para desarrollar los sistemas heterogéneos, además presentan un ejemplo de modelado de robots múltiples conjuntos heterogéneos y se mencionan el desarrollo de software semi-automático con el ejemplo.
Para resolver este problema de desarrollo de software embebido, se implementan las herramientas automáticas para la transformación de modelos (tales como TIM (Target independent model) y TSM (Target specific model)) y la generación de código. En [41], Seung, Kim, y Chul adoptan un tipo de mecanismos del proceso de software, que es MDA, en el proceso de desarrollo de software embebido, sobre la base de este proceso, se realizan los mecanismos automáticos para robots de múltiples conjuntos heterogéneos que se desarrollan fácilmente a través de la conversión automática de un modelo a otro modelo basado en cada objetivo particular de hardware. Con este método es posible mejorar la productividad de código asociado a la reutilización de modelos.
El enfoque que presentan Seung, Kim y Chul en [41], es un modelo que se centra en el desarrollo de software embebido donde se eligen varios diagramas, tales diagramas eran modelos estáticos y modelos dinámicos, el primer modelo utiliza diagrama de clases para representar el aspecto invariable de un sistema y el segundo modelo hace referencia a los diagramas de mensajes y de estados concurrentes que representan el comportamiento dinámico del sistema.
El proceso de transformación MDD establecido constaba de dos pasos de transformación T1, T2; el T1 es la transformación de un TIM a TSM asociado con el sistema operativo y el procesador y el T2 es la transformación de la TSM a una TDC transformar en realidad el código ejecutable de TSM a través de transformaciones T1. En el T2 se genera un lenguaje con el modelo de plantilla en el básico de diagrama de clases, diagrama de mensajes concurrentes, y el diagrama de estado concurrentes de TSM, sus posibles lenguajes de código generables son C, C++, y Java. El método de generación de código se obtiene automáticamente utilizando modelos y como resultado, esto conduce a reducir el ciclo de vida del desarrollo de software embebido.

En [42], los autores proponen el DSL RPG: un estudio de caso de la ingeniería lingüística utilizando MDD para generar juegos de rol para los teléfonos móviles, como en el ámbito de los juegos digitales se tienen muchos problemas de desarrollo debido a su creciente complejidad; dificultades como: poca reutilización de código con el fin de desarrollar un juego multi-plataforma y la realización de la verificación del juego a través de pruebas extensas y costosas, lo cual da como resultado una baja productividad en el desarrollo de soluciones de juego, para dar solución a estos problemas se propone un lenguaje específico de dominio (DSL) para un juego de rol (RPG) líneas de productos, el cual es construido por completo con una técnica de desarrollo de software impulsada por abstracciones de alto nivel también llamado MDD, dándose a conocer los beneficios de la aplicación de varios MDD en términos de creación rápida de prototipos de juegos de plataforma.

E. Dominios que implementan MDD

Se analizaron algunos dominios que emplean desarrollo de software dirigido por modelos y que han tenido éxito como lo son: Los Almacenes de Datos (Data Warehouse) [29] gracias al metamodelo CWM (Common Warehouse Metamodel), que se utiliza para un intercambio de información de metadatos del almacén de datos [50].

Según [51] el modelado de aplicaciones web, por ejemplo en [52], Meliá, Martínez, Pérez y Gómez presentan una herramienta que da soporte a la aproximación denominada OOH4RIA-Tool permitiendo representar los modelos y transformaciones para acelerar la obtención de una RIA (Rich Internet Applications) implementada con el framework GWT; en [53], los autores presentan ADM (Ariadne Development Method): Método de diseño para la generación de prototipos web rápidos a partir de modelos, el método ADM sigue un enfoque MDD para afrontar el modelado de las aplicaciones hipermedia y web que permite representar todas las características del sistema, utilizando abstracciones pertenecientes al dominio de la hipermedia, interrelacionándolas entre sí y consiguiendo una especificación independiente de plataforma y por último en [54] y [55] los autores presentan el desarrollo de una aplicación de hipermedia móvil usando derivación semiautomática a partir de los modelos específicos.

Integración de sistemas gestión de aprendizaje, conocidos como LMS (Learning Managment Systems), [39], que

permiten principalmente la interoperabilidad de los recursos existentes en mínimo dos plataformas [56].

Desarrollo de aplicaciones usando estándares de metamodelos tales como: Business Process Modeling Notation (BPMN) es una notación gráfica que describe la lógica de los pasos de un proceso de negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes en las diferentes actividades. [57]

Ontology Definition Metamodel (ODM) es una definición de la OMG especificación para hacer los conceptos de arquitectura dirigida por modelos aplicables a la ingeniería de ontologías [25].

IV. RESULTADOS

La tabla 1, resume las características que se han tenido en cuenta para cada una de las propuestas orientadas al contexto educativo. Se presentan como columnas las características y como filas las propuestas en donde el símbolo ☑ significa que lo contiene y el símbolo ☒ que no lo contiene.

Item	Propuestas	Características						
		MDD	Orientada a la Educación	Modelo	PIM	PSM	Transformación	Código
1	Pascuas, en [24]	☑	☑	☑	☑	☑	☑	☑
2	Texier, en [26]	☑	☑	☑	☒	☒	☑	☑
3	Montenegro, en [27]	☑	☑	☑	☑	☑	☑	☑
4	De Souza, en [28]	☑	☑	☑	☑	☑	☑	☒
5	Neil, en [29]	☑	☑	☑	☒	☒	☑	☒
6	Teruaki en [30]	☑	☑	☑	☑	☑	☑	☑
7	Francia, en [31]	☑	☑	☑	☑	☑	☑	☑
8	Cowling en [32]	☑	☑	☒	☒	☒	☒	☒
9	Hamou-Lhadj, en [33]	☑	☑	☑	☑	☑	☑	☑

Tabla 1. Resumen de propuestas MDD en el contexto educativo.

V. CONCLUSIONES

Los modelos son la herramienta fundamental en el paradigma del MDD, ya que estos son una conceptualización del dominio del problema y de su solución, focalizándose sobre el mundo real: donde se identifica, clasifica y abstrae los elementos que constituyen el problema los organizan en una estructura formal, en donde la abstracción es una de las principales técnicas con la que la mente humana se enfrenta a la complejidad. Ocultando lo que es irrelevante, un sistema complejo se puede reducir a algo comprensible y manejable; cuando se trata de software, es sumamente útil abstraerse de los detalles tecnológicos de implementación y tratar con los conceptos del dominio de la forma más directa posible. De esta forma, el modelo de un sistema provee un medio de comunicación y negociación entre usuarios, analistas y desarrolladores que oculta o minimiza los aspectos relacionados con la tecnología de implementación.

En los trabajos relacionados en el contexto educativo que ponen en práctica el MDD se evidencia que los modelos son una herramienta que proporciona ventajas en este contexto, porque permite estructurar y facilitar la reutilización de modelos para la creación de nuevos sistemas.

La utilidad del modelo y el metamodelo en MDD está centrada en definir lenguajes de modelado sin ambigüedades, contando con herramientas de transformación para leer y entender los modelos, en tener reglas de transformación claras que describan cómo un modelo en un lenguaje fuente va a ser transformado a un modelo en un lenguaje destino y en el uso de definiciones formales obtenidas por la sintaxis de los lenguajes, facilitando su automatización.

En la última década, ha habido un creciente interés en el MDD, en la industria y el mundo académico, el MDD no es una idea nueva, es un prometedor paradigma para abordar la composición del sistema y la integración; centrado en el modelo de desarrollo y generación de código a partir de una mayor abstracción, trayendo beneficios de productividad; sin embargo, hay pocos estudios empíricos disponibles para estudiar el impacto de la adopción de MDD en los procesos de desarrollo de software industrial, ocasionando grandes impactos a la hora de la adopción de este nuevo paradigma.

VI. REFERENCIAS

[1] Pons, C., Giandini, R., Pérez, G., 2010, Desarrollo de Software Dirigido Por Modelos, Conceptos Teóricos y su aplicación práctica; Universidad Nacional de La Plata.

[2] Introduction to the MDD Maturity Model, ESI, European Software Institute; Buenos Aires –INTI –June 2007

[3] Bran S., The Pragmatics of Model-Driven Development; IBM Rational Software. 2003

[4] Jordi C., MDD - Desarrollo de software dirigido por modelos que funciona.

[5] Pons C., El Proceso de Desarrollo MDD; Universidad Abierta Interamericana, Buenos Aires, 2008.

[6] Rojas Escobar K., Un acercamiento al desarrollo dirigido por modelos, Facultad 5. Departamento de Ciencias Básicas. Universidad de las Ciencias Informáticas, 2011.

[7] Booch, G. "An MDA Manifesto". InFrankel, D. and Parodi J. (eds) The MDA Journal: Model Driven Architecture Straight from the Masters, 2004.

[8] Durán M. F., Troya C. J., Vallecillo A., Desarrollo de Software Dirigido por modelos; UOC, Universitat Oberta de Catalunya.

[9] López L. E., González G. M., López S. M., Iduñate R E., Proceso de Desarrollo de Software mediante herramientas MDA; Departamento de Ciencias Computacionales, Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), Cuernavaca, Morelos, C.P. 62490, México.

[10] A. Navarro, J. Cristóbal, C. FernándezChamizo, and A. FernándezValmayor,-- "Architecture of a multiplatform virtual campus," Software: Practice and Experience, Sep. 2011.

[11] Pelechano V., Vallecillo A., Muñoz J., Fons J., Acta del I taller sobre MDD y MDA y aplicaciones (DSDM'04), Málaga, España, 2004.

[12] Texier J., Giusti M., Oviedo N., Villareal G., Lira A., Los Beneficios del Desarrollo de Software Dirigido por Modelos en los Repositorios Institucionales; Universidad Nacional Experimental de Táchira, Venezuela.

[13] Higuera Igarza I., Grass Boada D.H., Páez Valdés A., Una introducción al desarrollo de software dirigido por modelos, Departamento de Programación e Ingeniería de Software. Facultad 2. Universidad de las Ciencias Informáticas, 2012.

[14] Vallecillo A., Desarrollo Dirigido Por Modelos, Dpto. Lenguajes y Ciencias de la Computación; Universidad de Málaga, 2010.

[15] Balasubramanian K., Gokhale A., Karsai G., Sztipanovits J., Neema, Developing S., Applications Using Model-Driven Design Environments, Vanderbilt University, Published by the IEEE Computer Society, February 2006.

[16] Desarrollo Basados En Modelos; UGR, Universidad de Granada.

[17] Giandini R., Aplicando MDD al desarrollo de un Sistema; Master en Tecnología Inform ática, UAI, Seminario de Actualización Tecnológica, 2008.

[18] Jiménez A., Vara J. M., Bollati V. A., Gestión de la trazabilidad en el desarrollo dirigido por modelos de Transformaciones de modelos. Universidad Rey Juan Carlos, Madrid (España).

[19] ATL User Manual -version 0.7, 2007, ATLAS group LINA & INRIA.

[20] Enterprise Architecture, 2012, "Enterprise Architecture", Visitado el: 19 de Noviembre de 2012,
Disponible en:
http://www.sparxsystems.es/New/products/ea.html

[21] Chicote C. V., Caceres D. A., Herramienta eclipse para el desarrollo de software dirigido por modelos, Universidad Politécnica de Cartagena. 2007.

[22] Vicente C. & Alonso D., 2009, "Herramientas Eclipse para Desarrollo de Software Dirigido por Modelos". Visitado el 19 de Noviembre 2012. Disponible en:
http://www.mondragon.edu/jisbd2009/Documentos/TutorialJISBD_DSDM.pdf

[23] Curso DSDM. Sesión 4 - Genereración de código (MOFScript) Grupo Modelum Universidad de Murcia 10 de marzo de 2009. Disponible en: http://www.modelum.es/cursos/centic2009/formacion/sesion4_mofscript.pdf Visitado el 9 de Julio 2011

[24] Pascuas R. Y., Bocanegra G. J., Ortiz L. E., Pérez C. N. Desarrollo Dirigido Por Modelos Para La Creación de Lavoratorios Virtuales, Ingeniería de Sistemas, Universidad de Amazonia, Florencia, Colombia, 2012.

[25] Pascuas Rengifo Y. S. Laboratorios Virtuales integrados con tecnología grid para la universidad de la Amazonia. Florencia, Caquetá, 2010.

[26] Texier J., De Giusti M., Oviedo N., Villarreal G., Lira A., Los Beneficios del Desarrollo Dirigido por Modelos en los Repositorios Institucionales; Universidad Nacional Experimental del Táchira (UNET), Venezuela.

[27] Montenegro, C. E., Cueva J. M., Sanjuán O., y Gaona P. A. Desarrollo de un lenguaje de dominio específico para sistemas de gestión de aprendizaje y su herramienta de implementación "KiwiDSM" mediante ingeniería dirigida por modelos. En: Ingeniería Vol. 15, No. 2. 67-81. 2010.

[28] C. de Souza M. de Fatima, Castro-Filo J.A., Andrade R. MC., Model-Driven Development in the Production of Customizable Learning Objects, International Conference on Advanced Learning Technologies, 2010.

[29] Neil C., Irazábal J., De Vincenzi M., Pons C., Graphical Query Mechanism for Historical Data Warehouse within MDD, Universidad Abierta Interamericana (UAI),International Conference of the Chilean IEEE Computer Science Society, 2010.

[30] Teruaki K., Takao F., Seiko A. y Shin K., Development of a modeling education program for novices using model-driven development, Proceedings of the Workshop on Embedded and Cyber-Physical Systems Education, ACM New York, NY, USA, 2013.

[31] Francia R.,Bierman J.,HC Cheng B., Repository for model driven development (ReMoDD), Department of Computer Science and Engineering, Michigan State University, East Lansing, Michigan, Proceedings of the 2006 international conference on Models in software engineering, Springer-Verlag Berlin, Heidelberg ©2006.

[32] Cowling T., Model-Driven Development and the Future of Software Engineering Education; Department of Computer Science, University of Sheffield, 2013.

[33] Hamou-Lhadj A., Gherbi A., Nandigam J., The Impact of the Model-Driven Approach to Software Engineering on Software Engineering Education, Sixth International Conference on Information Technology: New Generations, 2009.

[34] Mzid R., Mraidha C., Babau J., Abid M.,A MDD Approach for RTOS Integration on Valid Real-Time Design Model, Euromicro Conference on Software Engineering and Advanced Applications, 2012.

[35] PAZ A., ESGUERRA A., ROJAS D., GARCÍA F., ARBOLEDA H., Manejo de variabilidad positiva y negativa en modelos de decisión; Grupo de Investigación en Informática y Telecomunicaciones (i2t), Universidad Icesi, Cali, Colombia, 2011.

[36] France R, Ghosh S., Dinh-Trong T., Model-Driven Development Using UML 2.0: Promises and Pitfalls, Colorado State University, Published by the IEEE Computer Society, 2006.

[37] Marín B., Giachetti G., Pastor O., Vos T. E., Abran A., Evaluating the Usefulness of a Functional Size Measurement Procedure to Detect Defects in MDD Models, Centro de Investigación en Métodos de Producción de Software, Universidad Politécnica de Valencia, Spain

[38] Streitferdt D., Wendt G., Nenninger P, Nyßen A., Lichter H.,Model Driven Development Challenges in the Automation Domain, Research Group Software Construction, RWTH Aachen University, IEEE International Computer Software and Applications Conference, 2008.

[39]Firstauthor J. M., Coauthor M., Secondcoauthor A., Ghezala H., Model-Driven Development of Context-aware Adaptive Learning Systems, Université des Sciences et Technologies, Lille – France, IEEE International Conference on Advanced Learning Technologies, 2010.

[40] Afonso M., Vogel R., Teixeira J., From Code Centric to Model Centric Software Engineering: Practical case study of MDD infusion in a Systems Integration Company, Enabler, Published by the IEEE Computer Society, Portugal, 2006.

[41]Seung H., Kim W., Chul Kim Y.,Semi-Automatic Software Development Based on MDD for Heterogeneous Multi-Joint Robots, Second International Conference on Future Generation Communication and Networking Symposia, 2008.

[42]Marques E., Balegas V., Barroca B, Bariši´c A., Amaral V.,The RPG DSL: a case study of language engineering using MDD for Generating RPG Games for Mobile Phones, Universidad e Nova de Lisboa, Portugal.

[43]Heijstek W., Chaudron M.,The Impact of Model Driven Development on the Software Architecture Process,Leiden University, Niels Bohrweg, The Netherlands,EUROMICRO Conference on Software Engineering and Advanced Applications, 2010.

[44] Quintero J., Duitama M. J., Reflexiones acerca de la adopción de enfoques centrados en modelos en el desarrollo de software Ingeniería y Universidad, vol. 15, núm. 1,pp. 219-243, Pontificia Universidad Javeriana, Colombia, 2011.

[45] Pandolfo P., Imparato C., Martiñera C., Triay F., Proyecto de investigación, modelado de procesos de negocio en el contexto de MDD, Escuela de Sistemas, Universidad J. 2011.

[46] Teppola S., Parviainen P., Takalo J., Challenges in the Deployment of Model Driven Development, Fourth International Conference on Software Engineering Advances, 2009.

[47] Quintero J. B., Duitama J. F., Reflexiones acerca de la adopción de enfoques centrados en modelos en el desarrollo de software, Vol 15, pp. 219-243, Pontifica Universidad Javeriana, Colombia, 2011.

[48] Solberg A., Simmonds D, Reddy R., Ghosh S., France R.,Using Aspect Oriented Techniques to Support Separation of Concerns in Model Driven Development, IEEE, 2005.

[49] Fuentes L. y Sánchez P., Desarrollo de software con aspectos dirigido por modelos. Universidad de Málaga, Málaga, España.

[50] J.-N. Mazón and J. Trujillo, "An MDA approach for the development of data warehouses," *Decision Support Systems*, vol. 45, no. 1, pp. 41–58, Apr. 2008.

[51] Meliá S. Un método de desarrollo dirigido por modelos de arquitectura para aplicaciones web. Universidad de Alicante.

[52] Meliá S., Martínez J. J., Pérez A y Gómez J., OOH4RIA Tool: Una Herramienta basada en el Desarrollo Dirigido por Modelos para las RIAs; Universidad de Alicante, IWAD, Campus de San Vicente del Raspeig, Apartado 99 03080.

[53] Montero S., Díaz P., Aedo I., Motenlls L., ADM: Método de diseño para la generación de prototipos web rápidos a partir de modelos, Laboratorio DEI. Dpto. de Informática, Escuela Politécnica Superior, Universidad Carlos III de Madrid, 2006.

[54] C. Challiol, "Desarrollo dirigido por modelos de aplicaciones de hipermedia móvil," Tesis, Facultad de Informática, 2011.

[55] Challiol C. Desarrollo Dirigido por modelos de aplicaciones de hipermedia móvil. Facultad de Informáticas, 2011.

[56] Romero A., Ceballos F.,Líneas de Productos de Software Dirigidas por Modelos (MD-SPL): Oportunidades y Retos; Universidad de los Andes, Bogotá, Colombia.

[57] BizAgi Process Modeler, BPMN Bussines Process Modeling Notation, Disponible en:
http://www.bizagi.com/eng/downloads/BPMNbyExample.pdf.

[58] "OMG Formal Specifications." [Online]. Available: http://www.omg.org/spec/. [Accessed: 18-Sep-2012].