

Análisis de la Transformación de Modelo CIM a PIM en el Marco de Desarrollo de la Arquitectura Dirigida por Modelos (MDA)

Martínez S.*; Lamoth L.*; Moreno R**; Jacho N.***

* Universidad de Holguín, Departamento de Informática .Holguín, Cuba
Email: {ssolerm; llamothb}@fcainf.uho.edu.cu

** Universidad Central “Martha Abreu” de Las Villas, Cuba,
e-mail: rosendo@uclv.edu.cu

*** Universidad de las Fuerzas Armadas ESPE, Departamento de Ciencias Computacionales, Sangolqui, Ecuador
e-mail: npjacho@espe.edu.ec

Resumen: La Ingeniería de Software es una disciplina vanguardista que más progreso exige en el proceso de desarrollo de software, en particular, los métodos, algoritmos y técnicas novedosas que enriquezcan y nutran el desarrollo de herramientas computacionales; herramientas que asistan al desarrollo de otros sistemas y apoyen el proceso de desarrollo de sistemas. Entre los marcos de desarrollo propuesto destaca la arquitectura dirigida por modelo (MDA) que pretende dar solución al problema de las plataformas cambiantes en las empresas que asumen el uso de sistemas informáticos, la interoperabilidad y la portabilidad de los mismos. Este trabajo aborda las etapas dentro del marco de desarrollo de software, la arquitectura dirigida por modelos, el uso de estándares y realiza un análisis de las tendencias y aportes principales teórico prácticos realizados en este sentido haciendo énfasis en la transformación del modelo independiente de la computación (CIM) al modelo independiente de la plataforma(PIM).

Palabras clave: MDA, CIM, PIM, Modelo de Plataforma Específico PSM, Transformación, Notación y Modelos de Procesos de Negocio BPMN

Abstract: Software engineering is one of the vanguards disciplines that more progress required on software development process. Methods, algorithms and new techniques to enrich the development of computational tools to assist the development of other systems. Among the frameworks developing highlighting Model Driven Architecture (MDA).This framework pretending to solve changing platforms problems inside of enterprises, the interoperability and portability. That's paper show the steps of MDA, the standard used and an analysis about new contributions and tendency in that way making emphasis in the transformation of Computationally-Independent Model CIM to Platform-Independent Models PIM.

Keywords: MDA, CIM, PIM, Platform-Specific Model PSM, Business Process Model and Notation BPMN, Transformation.

1. INTRODUCCION

En las últimas décadas se ha trabajado en el área de desarrollo de sistemas para encontrar técnicas que permitan incrementar la productividad y el control de calidad en cualquier proceso de elaboración de software. Dentro de esta área, la Ingeniería de software es una de las temáticas que más desarrollo demanda en los momentos actuales. Las investigaciones relacionadas se han dirigido hacia el desarrollo de métodos y algoritmos apropiados que conlleven a la creación y explotación de herramientas que asistan por medios computacionales al desarrollo de otros sistemas.

Entre las alternativas que han surgido figura el marco de trabajo MDA, que constituye un intento de proporcionar una

solución al problema de las plataformas cambiantes y a la portabilidad de los sistemas de información. MDA como arquitectura está destinada a aumentar la productividad en el desarrollo de software reduciendo el coste del mismo en términos de tiempo y de reutilización de componentes desarrollados para aplicaciones similares, dentro de las diferentes etapas del ciclo de desarrollo de software.

La base fundamental del marco de trabajo MDA, es la construcción y generación de mecanismos de especificación de modelos y sus transformaciones, hasta la obtención de un producto de software (modelo) dirigido a elevar la calidad del proceso de desarrollo de software, lo cual, constituye un problema de investigación que se pretende solucionar hoy día. Dentro de MDA, la transformación juega un papel

fundamental, al punto de socializarse como “el corazón y el alma de MDA” o “el eslabón perdido de MDA”. Para cristalizar los propósitos de una transformación, se deben utilizar lenguajes que permitan expresar qué quiere transformar y cómo quiere transformar.

La lógica declarativa para expresar el qué, se ha venido estandarizando a través de la maduración y acogida del lenguaje de restricciones de objetos (OCL) por sus siglas en inglés, pero la lógica imperativa para expresar el cómo, todavía no se encuentra suficientemente estandarizada [1].

En tal sentido este trabajo tiene como objetivo, analizar los aportes realizados en el ambiente de desarrollo de Arquitectura Dirigida por Modelos MDA, que permita establecer las bases para formalizar reglas que faciliten la transformación del Modelo Independiente de la Computación CIM al Modelo Independiente de la Plataforma PIM, a partir de la utilización de métodos y técnicas de investigación científica como el análisis y síntesis, el histórico lógico, la modelación, la observación científica y la revisión de documentos.

2. EVOLUCIÓN DE LA ARQUITECTURA DIRIGIDA POR MODELOS

En [2] se anuncia el surgimiento de la arquitectura dirigida por modelos MDA como respuesta a las necesidades emergentes en el proceso de desarrollo de software, haciendo alusión a los problemas de integración e interoperabilidad entre los sistemas. Trata de unificar lo construido con lo que se está construyendo. Esgrime una respuesta ante el paso acelerado de la tecnología, que requiere de la existencia de una arquitectura que haga posible la interoperabilidad de la infraestructura con que se cuenta.

De este modo, se anuncia MDA como un enfoque de desarrollo de sistemas que proporciona un medio a través de los modelos para dirigir el proceso de entendimiento, diseño, construcción, desarrollo, operación, mantenimiento y modificación a la altura del año 2002.

En [3] se especifican los modelos esenciales que se conciben en MDA para la obtención de un sistema informático. Se estructura en el modelo independiente de la plataforma (PIM) y en modelos de plataformas específicas (PSMs). Donde la funcionalidad del PIM es realizada y especificada de algún modo en una plataforma específica PSM, que se deriva a partir de alguna transformación.

Propone un proceso de desarrollo basado en la realización y transformación de modelos, centrando su atención en los modelos como elemento básico, situando a estos como estrategia clave para entender y especificar una solución de software.

El mencionado artículo evidencia su aporte en la propuesta de la concepción y transformación de modelos, tomándolos como punto de partida en el desarrollo de software, lo cual constituye uno de los aspectos más importantes [3, 4]. Sin

embargo en MDA Guide Versión 1.0.1 [5] se enuncia entre los conceptos básicos de MDA, el modelo independiente de la computación (CIM) como modelo que forma parte de los elementos a considerar en esta propuesta. El CIM no muestra detalles de la estructura del sistema, suele ser llamado modelo del dominio y el vocabulario empleado es muy cercano a los expertos del dominio en cuestión. Esta propuesta es actualizada por los propios autores de MDA en 2005 donde refuerza los conceptos fundamentales especificados para guiar el desarrollo basado en MDA.

Con este enfoque se favorece el desarrollo de aplicaciones empresariales completamente funcionales a partir de modelos con UML y otros estándares que garanticen la independencia de las diversas plataformas, abiertas o propietarias, como servicios web, .Net, Corba, J2EE, u otras.

2.1. Los modelos en MDA y su contribución en el ciclo de vida de un producto informático

MDA propone la definición y uso de modelos en el proceso de desarrollo de software a diferente nivel de abstracción, así como la posibilidad de la generación automática de código a partir de los modelos definidos y de las reglas de transformación entre dichos modelos. [6, 30] Los modelos a los que se hace referencia son:

- CIM: Es el modelo independiente de la computación, surge en la fase inicial del proceso de desarrollo comprendiendo la modelación del negocio en su totalidad.
- PIM: Representa los modelos que describen una solución de software que no contiene detalles de la plataforma concreta donde será implementada la solución, de ahí su nombre de modelos independientes de la plataforma. Estos modelos surgen como resultado del análisis y diseño.
- PSM: Los modelos específicos de la plataforma. Surgen del PIM y se crean entre las fases del diseño y la codificación de la solución.

Luego, el código se genera después de la codificación y las pruebas.

El proceso de desarrollo de software basado en MDA presupone un cambio en la secuencia de actividades para la obtención de un producto informático. A pesar de que el ciclo de vida del producto informático mantiene sus etapas desde la captura de requisitos hasta el despliegue de la aplicación; el uso de este marco de trabajo deviene la adaptación del ciclo de desarrollo de software. Una vez que se avanza en las etapas dentro del ciclo de desarrollo de software se va centrando el trabajo en la construcción de los modelos, su refinamiento y la posterior transformación.

En el proceso de desarrollo tradicional, se genera una gran cantidad de documentos y diagramas para especificar requisitos, clases, colaboraciones, etc. La mayoría de este material pierde su valor en cuanto comienza la fase de codificación, y gradualmente se va perdiendo la relación

entre los diagramas y su correspondencia con el producto final. Sin embargo con el uso de MDA, los cambios y la documentación derivada se pueden mantener sin costes adicionales. Esto se debe a que; una nueva entrada en los elementos a considerar se asume en el modelo CIM o PIM y con las transformaciones exactas se obtiene el resto de los elementos. Estos modelos desempeñan una labor clave con la gran diferencia de que estos no se abandonan tras la codificación.

De acuerdo con lo expresado en [7, 29] todos estos modelos se apoyan en los diversos estándares y tecnologías Unified Modeling Language (UML), Meta Object Facility (MOF), XML Metadata Interchange (XMI), Common Warehouse Metamodel (CWM) y Object Constraint Language(OCL), a través de su uso es posible enfrentar la rápida evolución tecnológica y la diversificación de plataformas y tecnologías, problemas por los cuales surge el marco de desarrollo de software basado en MDA.

2.2. Transformación de Modelos

La transformación es el paso que: partiendo desde un primer modelo, interpretado por determinado lenguaje pueda obtenerse un destino. En [8] se ofrece una taxonomía a través de un modelo de aspectos que establece una comparación entre los diferentes enfoques realizados alrededor de la transformación de modelos teniendo en cuenta los criterios y los aportes de diversos autores.

La transformación de modelos se considera el proceso central de MDA. Para especificar estas transformaciones se apoya el uso del estándar Queries/Views/ Transformations (QVT). Este estándar se expresa en MOF y queda establecido un lenguaje para la transformación de modelos, la consulta y la generación de vistas. QVT brinda la posibilidad de explorar los modelos desde disímiles puntos de vistas. Considera en el mismo la transformación, el mapeo y el marcado, elementos de interés dentro de este proceso.

En [7] se ilustra que los modelos tanto de origen como el destino, tienen una transformación, se especifican conforme a los metamodelos correspondientes, definiendo las transformaciones y sus ejecuciones se reflejan en la capa de la instancia de estos metamodelos.

Por otra parte, si se tiene en cuenta la definición de mapeo, asumida por los autores es posible realizar mezclas de diferentes estrategias en su ejecución.

Los autores asumen la definición para el marcado expresados en [7, 9] como la identificación de cada uno de los elementos del modelo origen, que permiten seleccionar qué reglas y qué tipo de mapeo corresponden para la transformación. Las marcas para un modelo suelen provenir de diferentes fuentes.

3. ANÁLISIS DEL ESTADO ACTUAL DE MDA

Con los referentes y definiciones que sustentan MDA, resulta muy compleja la deserción y delimitación de las fronteras

entre un modelo y otro. Esto está provocando la falta de homogenización en la definición de un modelo y los parámetros que este contendrá según se observa en [10].

Si se analizan los elementos existentes alrededor de la definición de cada uno de los modelos que forman MDA y sus transformaciones se aprecia que para el caso de las transformaciones del PIM a PSM y de PSM a código, existen varias herramientas que la realizan. Sin embargo la definición del CIM y su transformación al PIM es la menos explorada.

Esta situación se debe precisamente a la naturaleza de los modelos involucrados en las primeras etapas del proceso de desarrollo (CIM y PIM) cuyos detalles se describen a en la Tabla 1.

Si se tienen en cuenta los elementos descritos hasta el momento se puede concluir que existen varios referentes teóricos que abordan el enfoque MDA donde prima la variedad en las interpretaciones de cada uno de los modelos en los que subyace este marco de desarrollo. Por tales razones y en unión al criterio de diversos autores se afirma la definición y delimitación de cada uno de los modelos de MDA. Todo esto con el objetivo de obtener representaciones homogéneas en la definición de la información a plasmar en CIM y/o PIM, como también las la transformación entre estos dos modelos.

3.1. Modelo Independiente de la Computación

En MDA el modelo de origen es el CIM, que permite modelar los requisitos del sistema. Por tanto es de especial utilidad cuando se desea crear un vocabulario común para el desarrollo de software que se empleará en las etapas sucesivas en su desarrollo. Describe el ambiente en que se desarrollará el sistema.

Utiliza un lenguaje para modelar procesos de negocios que no tiene que ser precisamente UML, aunque éste puede ser derivado perfectamente utilizando MOF. El CIM se manifiesta a través de los procesos de negocio y deja establecidas las relaciones entre trabajadores humanos o no, sus interacciones, responsabilidades y roles.

De este modo se facilita la comprensión de la lógica del negocio evitando todo tipo de conocimiento especializado o de sistema. Una muestra de ello es lo especificado en [11] que representa los requisitos mediante diagramas de caso uso, de actividad y de secuencia. Del mismo modo en que se representa la distribución de los procesos y la información a tratar. El CIM constituye una conexión entre los expertos del negocio que aportan los requisitos del sistema, en la misma medida que relaciona a los diseñadores y desarrolladores responsables de la implementación [12].

Tomando como base esta afirmación y los sustentos teóricos consultados se puede considerar entre las formas de representar el Modelo Independiente de la Computación es el Modelo y Notación de Procesos de Negocio (BPMN), este es

un estándar de Object Management Group (OMG) cuyo objetivo es proveer de una notación gráfica estandarizada en formato de flujo de trabajo, fácil de entender y de comunicar entre usuarios de todo nivel, eliminando así la brecha de comunicación que permite el modelado de procesos de negocio [25].

BPMN es ampliamente adoptada por las empresas para los propósitos de modelado de los procesos básicos, análisis detallado de performance, especificación de requerimientos y diseño ejecutable. La BPMN 2.0 añade nuevos elementos de diseño orientado a la ejecución, ofreciendo una definición teórica de lo que hace un modelo ejecutable.

Tabla 1. Elementos para la especificación de CIM y PIM y los métodos de transformación

Año	Fuente	Representación		Transformación de CIM a PIM
		CIM	PIM	
2004	[2]	Mediante ViewCIM derivados del proceso de desarrollo de software orientado a aspectos. Sustenta el modelado de la lógica del negocio con múltiples vistas	Mediante ViewPIM y se derivan de la representación del CIM empleada. Son transformados en diagramas de clases que colaboran para realizar dichos casos de uso	No dispone de un método de transformación de CIM a PIM
2007	[3]	Modelo del Negocio del Nivel CIM (Modelo de Valor)	Modelado del Sistema de Información del nivel PIM (Modelo de Casos de Uso)	Mediante guías propuestas por SOD-M, mapeo de requisitos a nivel CIM en elementos de modelos de nivel PIM
2007	[4]	Se representa por los Requerimientos del Usuario, enfocados hacia el desarrollo multidimensional de almacenes de datos.	Es una derivación del CIM denominado Modelo Conceptual multidimensional con independencia de plataformas y que contiene los elementos para generar el almacén de datos.	Mediante la descripción de un perfil de UML para construir un PIM para modelado multidimensional en almacenes de datos. Se consideran las fuentes de datos operacionales para completar el PIM inicial según las formas normales multidimensionales
2007	[5]	Secure Business Process (SBP) descrito con BPMN o UML que constituyen la materialización de un conjunto de requisitos de seguridad proporcionados por los expertos del negocio.	Conjunto de Clases de Análisis	El PIM se especifica con el Diagrama de Actividades de UML 2.0-AD extendido con BPSec. Ambos elementos constituyen aportes de los autores para representar los aspectos relacionados con la seguridad en los procesos de negocio. La transformación es realizada con QVT
2008	[6]	Especificación de Secure Business Process (SBP)	Clases de Análisis y los Casos de Uso	Se describe mediante QVT, reglas de refinamiento y listas de comprobación
2008	[7]	Modelo de transacciones de negocio (BTM) que refleja las interacciones entre organizaciones y se modela con BPMN	Diagrama de Colaboraciones UML2	Utiliza el lenguaje transformación atlas ATL y se describe un híbrido para transformar el modelo de transacciones de negocio (BTM) elaborado con BPMN y obtener un modelo de Colaboraciones UML2
2009	[8]	El CIM tiene correspondencia con los niveles de refinamiento de Requisitos y Análisis	El PIM tiene correspondencia con el nivel de Refinamiento de Diseño	Se realiza mediante un conjunto de actividades especificadas por medio del planificador de transformaciones.
	[9]	BPMN para representar un CIM	Se obtiene a partir de las transformaciones	A partir de uso de UML y transformaciones a partir de QVT
*	[10]	Se representa a partir de los artefactos tomados del modelo WRSPM asociado a las siglas en inglés de mundo, requisito, especificación, plataforma y máquina, sobre la base de la ingeniería de requisitos para definir los elementos que conforman el CIM	Modelo de Requerimientos.	La transformación del CIM al PIM se realiza mediante los elementos que componen WRSPM.
*	[11]	Combinación del Modelo del Negocio y el Modelo de Requerimientos	No	No
*	[12]	Modelo de Requisitos. Esta representación se obtiene empleando método de diseño web A-OOH relativo a Hipermedia orientada a objetos adaptados, proveniente de los principios de la ingeniería web dirigida por modelos.	Modelos conceptuales de dominio y navegación. Esta representación se obtiene mediante el mismo método que se especificó el CIM.	Modeladas utilizando QVT. La trazabilidad de requisitos entre modelos conceptuales se define (mediante la aplicación de modelos weaving que permite identificar las reglas aplicadas en las transformaciones realizadas)
2014	[27]	Compuesto por un modelo de componentes formal expresado mediante una ontología y un modelo de componentes UML.	Se representa mediante una ontología incluyendo en el metamodelo los conceptos de componentes, servicios y funcionalidades	Considera la transformación automática implementando un plugin para el framework. Posibilita la obtención de un modelo de componentes UML en XMI
2011	[28]	Se pronuncia sobre la reutilización de ontologías como base para el modelado dada que estas expresan el conocimiento sobre un dominio específico generado por sus propios expertos sobre la base de la lógica descriptiva que se considera.	No	A través de mecanismos de refinamiento y validación, la ontología empleada que constituye una analogía con los mecanismos de representación en varios niveles propios de MDA.

* El año de publicación del artículo no aparece en la fuente consultada.

3.2. Modelo Independiente de la Plataforma

El PIM es un modelo del sistema de alto nivel que representa la estructura, funcionalidad y restricciones del sistema sin aludir a una plataforma determinada. Este modelo servirá de base para todo el proceso de desarrollo y es el único que debe ser creado íntegramente por el desarrollador. Suele ser el punto de entrada de todas las herramientas para MDA.

Al no incluir detalles específicos de una tecnología determinada, este modelo es útil porque es fácilmente comprensible por los usuarios del sistema, y por tanto les resultará más sencillo validar la corrección del sistema. Además, facilita la creación de diferentes implementaciones del sistema en diferentes plataformas, dejando intacta su estructura y su funcionalidad básica.

Si se ha creado un modelo abstracto, donde se describe el comportamiento del sistema, aspectos funcionales y no funcionales independientes del entorno de computación y tecnologías de implementación, que pueden ser reutilizados en múltiples plataformas. Si se garantiza además que los requisitos del negocio se especifican utilizando diagramas UML y el sistema es modelado desde el punto de vista que mejor soporte los requisitos del usuario final, independiente de la implementación de la plataforma/tecnología [13], se han creado las condiciones que garanticen también la transformación de PIM a PSM.

3.3. Transformaciones

Dadas las condiciones a cumplir, para la formalización de reglas de transformación entre modelos y las condiciones para su ejecución, para realizar la transformación del CIM a PIM se afirma el uso del estándar QVT-Relations en el contexto de la arquitectura de metamodelos MOF[26] [29]. El metamodelo permite el intercambio, interoperabilidad y ejecución de modelos.

QVT-Relations es el lenguaje declarativo, es una especificación de las relaciones entre modelos MOF. Además es descriptivo porque define el qué y se basa en un conjunto de reglas que deben cumplirse simultáneamente para determinar si existe consistencia entre dos conjuntos de modelos.

También BPMN 2.0 incorpora características nuevas para la construcción de modelos ejecutables, que permite la construcción de diagramas, que puedan ser ejecutados sobre un motor de procesos. Incorpora un metamodelo basado en MOF que define la sintaxis y semántica abstracta de los constructores de modelado. El metamodelo refina y formaliza la semántica de ejecución de BPMN.

Un elemento que permite aseverar tal planteamiento lo constituye el hecho de que, tanto los metamodelos de BPMN 2.0 como el de UML 2 se representan como instancias del metamodelo MOF. Esto facilita la formalización de reglas de transformación entre el diagrama de procesos definidos con

la notación de BPMN 2.0 y un diagrama de clases UML, lo cual define la estructura del PIM, en este sentido van encaminados los trabajos futuros.

4. CONCLUSIONES

A partir de los aspectos analizados se ha arribado a las siguientes conclusiones:

En el marco de desarrollo de software basado en MDA existen varios referentes teóricos que lo describen, pero en algunos de estos excluyen al CIM del análisis, en la propuesta de desarrollo de MDA. Cada uno de ellos, tienen formas diferentes de interpretar cada modelo, en su mayoría no describen el proceso de la transformación del CIM al PIM. Con el análisis realizado se pudo comprobar que la definición teórica de los modelos MDA es clara, sin embargo la información de la delimitación del proceso de transformación de los modelos de CIM a PIM, y de PIM a PSM es escasa y heterogénea.

La definición del modelo CIM con BPMN2.0 constituye una propuesta a tomar en consideración debido a las características de esta notación, además de ser fácil de representar es una instancia de MOF, uno de los elementos que conforman el núcleo de MDA, lo cual favorece su aceptación en este sentido.

El modelo al ser parte central de MDA y al complementarse con ontologías, contribuye a mantener la uniformidad de cada modelo CIM, PIM, PSM, que aunque sean realizados por distintas personas, se disminuya la subjetividad de los programadores y, al mantener la estandarización permite validar los modelos para ser reutilizados y reducir el tiempo de desarrollo de software.

REFERENCIAS

- [1] Quintero, J.B. and R. Anaya de Páez, *Marco de Referencia para la Evaluación de Herramientas Basadas en MDA*. Grupo de Investigación en Ingeniería de Software, Universidad EAFIT., 2007.
- [2] Soley, R. and O.S.S. Group, *Model Driven Architecture*. 2000, Object Management Group. p. 12.
- [3] *Model Driven Architecture (MDA)*. 2001. p. 31.
- [4] Muñoz, J. and V. Pelechano, *MDA a Debate*. 2004, Universidad Politécnica de Valencia.
- [5] OMG, *MDA Guide Version 1.0.1*. 2003.
- [6] Cabello Espinosa, M.E., *Master: Análisis y diseño de un generador automático de sistemas de diagnóstico basado en líneas de producto*. 2007, Universidad Politécnica de Valencia: Valencia, España.
- [7] Quintero, J.B. and R. Anaya, *MDA y el papel de los modelos en el proceso de desarrollo de software*. Revista EIA, 2007: p. 131-146.
- [8] Czarnecki, K. and S. Helsen, *Classification of Model Transformation Approaches*, University of Waterloo: Canada.
- [9] Corredera de Colsa, L.E., *Arquitectura dirigida por modelos para J2ME*. 2005.
- [10] López L, E.D., et al., *Proceso de Desarrollo de Software Mediante Herramientas MDA*. Departamento de Ciencias Computacionales Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), 2006.
- [11] Gómez, J., Sáenz, H., *Propuesta de inclusión de MDA en Ingeniería de sistemas de la Universidad Tecnológica de Pereira*. Scientia et Technica Universidad Tecnológica de Pereira, 2011

- [12] de Castro, M.V., *Aproximación MDA para el desarrollo orientado a servicios de sistemas de información Web: del Modelo del Negocio al Modelo de Composición de Servicios Web*. 2007, Universidad Rey Juan Carlos: Escuela Técnica Superior de Ingeniería de Telecomunicación Departamento de Lenguajes y Sistemas Informáticos.
- [13] de Castro, V., et al., *Obteniendo Modelos Sistemas de Información a partir de Modelos de Negocios de Alto Nivel: Un Enfoque Dirigido por Modelos*. 2007.
- [14] Amaya, P., C. González, and J.M. Murillo, *Separación de Aspectos en MDA: Una aproximación basada en múltiples vistas*. Actas del I Taller sobre Desarrollo Dirigido por Modelos, MDA y Aplicaciones (DSDM'04), 2004.
- [15] Mazón, J.-N. and J. Trujillo, *Desarrollo de modelos multidimensionales de almacenes de datos basado en MDA: del análisis de requisitos al modelo lógico*. DSDM 2007: IV Taller sobre Desarrollo Dirigido por Modelos, MDA y Aplicaciones, 2007. **Vol. 1**(No. 6).
- [16] García- Rodríguez de Guzmán, I., et al., *Implementación de Heurísticas en QVT para la obtención de Clases de Análisis a partir de Modelos de Proceso de Negocio Seguros*. DSDM 2007: IV Taller sobre Desarrollo de Software Dirigido por Modelos, MDA y Aplicaciones, 2007. **Vol. 1**(No. 6).
- [17] Rodríguez, A., E. Fernández-Medina, and M. Piattini, *Elicitación de Requisitos de Seguridad en Procesos de Negocio*. 2008
- [18] Bocanegra, J., J. Peña, and A. Ruíz-Cortés, *Una Aproximación MDA para Modelar Transacciones de Negocio a Nivel CIM*. SISTEDES, 2008.
- [19] Muñoz, J. and V. Pelechano, *MDA vs Factorías de Software*. Dept. de Sistemas Informáticos y Computadores Universidad Politécnica de Valencia, 2009.
- [20] Tabares Betancur, M.S., *Un Patrón de Trazabilidad para Controlar la Evolución de los Intereses en un Espacio Multidimensional*. 2009, Facultad de Minas de la Universidad Nacional de Colombia.
- [21] Martínez, I., et al., *Integración de modelos BPMN en ambientes MDA*. Laboratorio de Innovaciones en Sistemas de Información, LINSI. Facultad Regional La Plata, Universidad Tecnológica Nacional. La Plata, Buenos Aires, Argentina.
- [22] Iqueira, F.L. and P.S. Muñoz Silva, *Analyzing CIM to PIM Transformations Using the WRSPM model*. INFOCOMP, v. 11, no. 1, p. 41-50.
- [23] Reza Sharifi, H. and M. Mohsenzadeh, *A New Method for Generating CIM Using Business and Requirement Models*. World of Computer Science and Information Technology Journal (WCSIT). **Vol. 2, No. 1**.
- [24] Aguilar, J.A., I. Garrigas, and J.-N. Mazón, *Modelos de weaving para trazabilidad de requisitos Web en A-OOH*. Actas de los Talleres de las Jornadas de Ingenierías de Software y Bases de Datos. Vol.2: p. 146-155.
- [25] Freund, J., B. Rucker, and B. Hitpass, *BPMN 2.0 Manual de Referencia y Guía Práctica*
- [26] García, M., *Formalization of QVT-Relations: OCL-based Static Semantics and Alloy-based Validation*. Institute for Software Systems (STS) Hamburg University of Technology (TUHH), 21073 Hamburg, 2008
- [27] Silega-Martínez, N., et al., *Framework basado en MDA y ontologías para la representación y validación de modelos de componentes* Framework based on MDA and ontology for the representation and validation of components model. Revista Cubana de Ciencias Informáticas, 2014. Vol.8(No.2): p. 16.
- [28] Lopez, G., et al. *Ontologías en arquitecturas dirigidas por modelos*. in XIII Workshop de Investigadores en Ciencias de la Computación. 2011
- [29] Pons, C., Giandini, R. S., & Pérez, G. *Desarrollo de software dirigido por modelos*. 2010
- [30] García, J., et al., *Desarrollo de software dirigido por modelos: Conceptos métodos y herramientas*, Ra-ma Editorial, 2013