

**Revisión de literatura sobre la inclusión de requisitos no
funcionales en MDSD para la generación de aplicaciones
con altos volúmenes de información**

Autores

Carlos Andrés Arroyave

Victor Manuel Henao Hernandez

Título otorgado

Especialista en Ingeniería de software

Asesor del trabajo

Jesus Andrés Hincapie Londoño

Especialización en Ingeniería de software

Facultad de ingenierías

Medellín

2014

Revisión de literatura sobre la inclusión de requisitos no funcionales en MDSD para la generación de aplicaciones con altos volúmenes de información

Carlos Andrés Arroyave¹, Victor Manuel Henao Hernandez²

¹Grupo Bancolombia, *carlosandresarroyave@gmail.com*

²Grupo Exito, *victorhenao@gmail.com*

Resumen

En las últimas décadas los desarrollos de software han tomado una gran importancia en las organizaciones; el tener aplicativos con buenos tiempos de respuesta ha desencadenado la necesidad de construir modelos de software que permitan gestionar grandes volúmenes de información y de concurrencia de usuarios. El acelerado ritmo de desarrollo ha contribuido al nacimiento de frameworks de trabajo como MDSD, que permitan a las organizaciones aumentar las tasas de éxito de sus proyectos. Este artículo estará orientado al análisis de las aplicaciones generadas con enfoques MDSD cuando se enfrentan al manejo de altos volúmenes de información y el cumplimiento de la necesidad de requerimientos no funcionales (RNFs) como el rendimiento, la escalabilidad y la adaptabilidad; se identificará en qué parte del proceso MDSD (model driven software development) debe ser incluida el manejo de los RNFs.

Palabras clave: MDSD, Rendimiento, Transaccionalidad, Requerimiento No funcionales, Información

1 Introducción

En los últimos años un gran número de equipos multidisciplinarios conformados por ingenieros de desarrollo y de calidad, físicos, matemáticos y demás carreras afines a la tecnología, han enfocado sus esfuerzos en investigaciones que logren la integración de los procesos de desarrollo de software con las necesidades no

funcionales de la industria. Los métodos tradicionales de construcción han desplazado los conceptos de rendimiento, adaptabilidad y escalabilidad para las etapas posteriores al desarrollo, trasladando esta tarea a las fases de certificación o estabilización del aplicativo, incrementando las posibilidades de fracasos y ocasionando que un alto porcentaje de los proyectos de software se vean obligados a implementar cambios que pueden ir desde la simple modificación de una interfaz hasta el replanteamiento general de la arquitectura de la solución (Goldschmidt, Reussner, & Winzen, 2008).

MDS (Model Driven Software Development) es una evolución de la aplicación de conceptos provenientes de modelos de desarrollo **ModelDrivenEngineering** (MDE) y **ModelDrivenArchitecture** (MDA) (Roser, Lautenbacher, & Bauer, 2008), los cuales tienen como objeto separar el diseño de la arquitectura de la etapa de construcción, facilitando así que el desarrollo y la arquitectura puedan ser alterados en forma independiente sin generar dependencia uno del otro. MDA centra su mayor utilidad en albergar todo lo correspondiente al levantamiento de requisitos No Funcionales, mientras que MDE se ocupa de todo lo relacionado a los requisitos funcionales (Cortellessa et al., 2006).

Los RNFs por lo general no son fácilmente identificables en un proyecto de software; con MDS se busca que este tipo de situaciones sean trabajadas con el modelamiento, detectando y definiendo claramente los requisitos funcionales y los no funcionales, para posteriormente llevar dichos requisitos al modelamiento de la aplicación. Basándonos en el estado actual de las aproximaciones realizadas por algunos investigadores sobre MDS (Ameller, Franch, & Cabot, 2010), se requiere de un especial cuidado para no dejar de lado los RNF, como ocurre igualmente en otros marcos de trabajo.

MDS tiene como filosofía el desacoplamiento de sus herramientas, dividiendo en tres conceptos la definición de la arquitectura a utilizar, estos son la transformación, lenguaje específico de dominio y la plataforma de destino (Quintero, Rucinke, Anaya, & Piedrahita, 2012), esta división ha ocasionado la aparición de una gran cantidad de artefactos, que al tratar de generar un proceso

de control y de calidad pueden surgir problemas de adaptabilidad debido al alto número de componentes. El desarrollo de este artículo tiene como objetivo la revisión de literatura confiable y seria, publicada en bibliotecas virtuales tales como la Association of Computing Machinery (ACM), IEEE, EBSCOhost y que tengan como foco la integración de requerimientos no funcionales mediante el enfoque de modelos, más específicamente MDSD, logrando determinar en qué parte de la etapa del modelo debe ser incluido el análisis del rendimiento, escalabilidad y adaptabilidad del sistema que se desea construir.

Este artículo contendrá la siguiente estructura: En la sección I se realizará una contextualización teórica sobre los enfoques de modelos y los aportes generados desde la comunidad científica. En la sección II se realizará una evaluación teórica, de los artículos electrónicos publicados en las bibliotecas virtuales sobre la integración de RFNs en la construcción de los modelos de desarrollo por medio de MDSD. En la sección III se presentará el resultado de la evaluación realizada en la sección anterior. La sección IV planteará la discusión sobre la integración de RFNs con los modelos MDSD. Finalmente, la última sección presentará las conclusiones generales y específicas del artículo.

2 Método

Basado en la literatura almacenada en las bibliotecas virtuales como Association of Computing Machinery (ACM), IEEE, EBSCOhost y portales de búsqueda como googleSchool y CiteSeerX, se realiza la extracción de información de artículos electrónicos, conferencias y casos de éxito en la industria de software relacionados con la integración de requisitos No Funcionales con modelos de desarrollo de software.

2.1 Preguntas de investigación

Planteamiento de las preguntas de investigación.

¿Cómo se integran en una solución los RNFs por medio de enfoque dirigido por modelos?

¿Qué casos de éxito sobresalen en implementaciones de RNFs mediante MDSD?

¿Cómo diferenciar RF de RNFs en modelos MDSD?

¿Cómo se modelan los RNFs mediante UML para MDSD?

2.2 Criterios de inclusión y exclusión

Inclusión

Para esta revisión literaria serán seleccionados todos aquellos artículos que cumplan las siguientes condiciones:

- La fecha de publicados en la bibliotecas virtuales ACM, IEEE y EBSCOhost debe ser a partir del año 2006 hasta la fecha.
- Información almacenada en paper, conferencia y libros que tengan como foco principal la integración de requisitos No funcionales con enfoques dirigidos por modelos.
- Artículos que incluyan casos exitosos y fallidos de procesos de modelado con UML orientados a la inclusión de RFNs en los 5 últimos años.
- Referencia de herramientas construidas con enfoques dirigidos con modelos y cuya integración incluya el manejo de RFNs.
- Solo se tendrá en cuenta información disponible en forma digital.

Exclusión

Para esta revisión literaria serán excluidos todos aquellos artículos que cumplan las siguientes condiciones:

- Artículos cuya fecha de publicación sea inferior al año 2006
- Papers que no hayan sido publicados en las bibliotecas virtuales ACM, IEEE y EBSCOhost
- Literatura enfocada al manejo de requisitos Funcionales.
- Explicación funcional de herramientas que hayan sido construidas con enfoques dirigidos por modelos.
- Material disponible en medio físico (Libro, revistas) no será utilizado dentro de la elaboración de este artículo.

De los términos utilizados dentro de la introducción y el resumen y haciendo uso de los conectores lógicos AND y OR se generaron las siguientes cadenas de búsqueda.

Tabla 1. Cadenas Generales de Búsqueda

Cadenas Generales de Búsqueda	
	"model driven software development" AND (RFNs (performance OR adaptability))
	"model driven software development" OR "model driven Architectures" OR "Model Driven Engineering"
	"model driven software development" AND (case (success OR failure))

2.3 Trabajos seleccionados

La tabla 1 muestra la selección de referencias literarias seleccionadas para la elaboración de este artículo.

Tabla 2. Trabajos seleccionados

ID	Título	Autores	Fecha	Fuente
Art.1	Approach for generating performance models from UML models of SOA systems	Mohammad Alhaj, Dorina C. Petriu	2010	ACM
Art.2	Representing and Using Nonfunctional Requirements: A Process-Oriented Approach	John Mylopoulos, Lawrence Chung, and Brian Nixon	2008	IEEE
Art.3	Dealing with Non-Functional Requirements in Model-Driven Development	David Ameller, Xavier Franch, Jordi Cabot	2010	IEEE
Art.4	Integrating Non-Functional Requirement Modeling into Model Driven Development Method	Yi Liu, Zhiyi Ma*, Weizhong Shao	2010	IEEE
Art.5	Software Performance	Vittorio	2006	ACM

	ModelDrivenArchitecture	Cortellessa, Antinisca Di Marco, Paola Inverardi		
Art.6	Model-Driven Performance Engineering of Self-Adaptive Systems: A Survey	Matthias Becker, Markus Luckey, Steffen Becker	2012	ACM

3 Evaluación de los trabajos seleccionados

En el presente numeral se realiza la evaluación cualitativa de los artículos seleccionados de las bibliotecas virtuales.

3.1 Criterios de evaluación

Los criterios de evaluación que se tienen en cuenta en este artículo son los siguientes:

- **Criterio 01:** Casos de éxito donde sobresalgan herramientas construidas bajo enfoques MDSD
- **Criterio 02:** Ilustración de desarrollos elaborados con modelos MDSD que integren la solución de RFNs
- **Criterio 03:** Se tendrán en cuenta aquellos artículos que realicen una contextualización teórica detallando la diferencia entre requisitos No funcionales y requisitos funcionales.
- **Criterio 04:** Implementaciones realizadas con metodologías UML que involucren RFNs.

Los anteriores criterios han sido enumerados en el orden de importancia que tendrán en el desarrollo de este artículo.

3.2 Resultado de la evaluación

Para la evaluación de los artículos se tiene en cuenta la siguiente escala cualitativa la cual está identificada con un carácter como se describe a continuación.

RC: Responde completamente a las preguntas de investigación.

RS: Responde satisfactoriamente a las preguntas de investigación.

RP: Responde parcialmente a las preguntas de investigación.

NR: No responde a las preguntas de investigación.

La tabla resumen debe tener el siguiente formato:

Tabla 2. Resumen de la evaluación

		Criterios de evaluación			
		Criterio 1	Criterio 2	Criterio 3	Criterio 4
Trabajos	Art.1	RP	RS	RP	RC
	Art.2	RC	RS	RP	RC
	Art.3	NR	RS	RC	RS
	Art.4	RC	RS	RP	RC
	Art.5	NR	RP	RP	RS
	Art.6	RS	RC	NR	NR

El Artículo identificado con el id **Art.1** enfoca su contenido en la implementación de proyectos bajo arquitecturas SOA (Service Oriented Architecture), acompañado de modelamiento UML en cada una de las versiones estandarizadas en la década pasada por la OMG (UML 1.X y UML 2.X) y basado en MARTE (Modeling and Analysis of Real Time and Embedded systems). Se propone un trabajo futuro que se considera importante para evaluar un caso de éxito, interesa mucho conocer el uso y aceptación de las extensiones propuestas para Eclipse. Adicionalmente es importante una introducción más clara a los RNF.

El trabajo con identificador **Art.2** propone un marco general donde representa el uso de requisitos no funcionales durante el proceso de desarrollo. El marco consta de componentes básicos que prevén la representación de los RNFs en términos de objetivos relacionados entre sí. El artículo muestra casos de éxito partiendo desde la complejidad general del sistema pasando por las necesidades globales del desarrollo, tales como desempeño, confiabilidad, robustez y

portabilidad y finalizando con la interacción entre el modelo y los RFNs. Este paper cumple con el criterio de inclusión de referencia de herramientas construidas con enfoques dirigidos por modelos donde incluyan la integración de requisitos no funcionales.

El estado de arte identificado como **Art.3** centra su trabajo en detallar teóricamente las diferencias existentes entre los enfoques dirigidos por modelos partiendo de la inclusión de requisitos no funcionales (NRFs) y requisitos funcionales, además de ello hace análisis desde el punto de vista matemático que son utilizados cuando se integran RFNs con modelos dirigidos con enfoques MDD. Dado el enfoque de este artículo, los autores no se concentran en tratar casos de éxito de aplicativos construidos bajo enfoques MDSD. Por el contrario si cubren de manera clara el manejo de RFNs en MDD, es positivo en este artículo el análisis de las diferentes estrategias MDD que se pueden abordar, incluyendo frameworks de trabajo y las cargas de responsabilidades que están a cargo de los desarrolladores.

El propósito del artículo identificado con el id **Art.4** propone un enfoque basado en modelos UML y los impactos que se tienen en la identificación de requisitos no funcionales (RFNs) en la implementación de modelos MDD. El autor toma modelos vigentes representados en UML basados en el diseño de caso de uso, diagramas de secuencia y de clases con el objeto de identificar en etapas tempranas los RFNs. Este artículo abarca de manera clara su enfoque, pero no profundiza demasiado en los criterios de evaluación descritos anteriormente; es muy amplio en el manejo de UML, aborda algunos aspectos sobre los impactos de los RFNs sobre los modelos, los elementos estructurales, elementos de comportamiento, las restricciones sobre un modelo y la forma de unir un modelo no funcional con un modelo funcional. Es interesante analizar el trabajo a futuro y evaluar las ventajas que este enfoque propone para lograr casos de éxito.

Objetivo del **Art.5** es integrar la validación del rendimiento en el marco de MDA. El documento propone una extensión del marco de MDA por medio SPMDA (ModelDriven Performance Software Architecture), además de los típicos modelos

MDA, incrusta nuevos modelos y transformaciones que tienen en cuenta las actividades de validación de rendimiento. Para este objetivo, el autor define nuevos tipos de transformaciones de modelos, que son diferentes a MDA, cuyo objetivo es convertir los modelos menos detallados en procesos más detallados, hasta la generación de código automatizado. Este artículo describe un framework y metodología para trabajar con modelos orientados al desempeño de las aplicaciones, pero no profundizan en casos de éxito logrados con dicho enfoque. Adicionalmente se abarcan de forma clara las dificultades que se tienen en etapas tempranas del modelamiento, cuando es requerido realizar las estimaciones sin todavía tener mucha información respecto a las necesidades de desempeño de la aplicación.

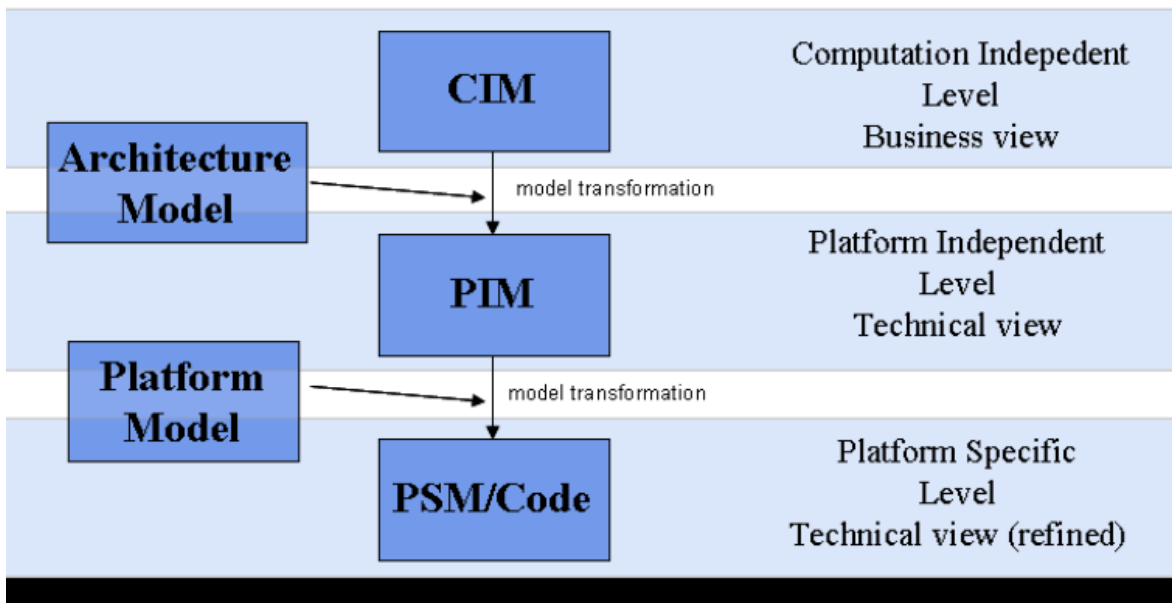
El **Art.6** brinda al lector una contextualización teórica de cómo las aplicaciones desarrolladas con enfoques dirigidos por modelos deben ser completamente auto-adaptables a los cambios continuos para la integración de requisitos no funcionales, este artículo fue seleccionado dado que se centra en identificar y comprender los enfoques de la ingeniería desde la perspectiva del análisis y el rendimiento de los sistemas auto-adaptables y construidos por modelos.

4 Discusión

En la conferencia Internacional de la IEEE sobre ingeniería de requisitos (Ameller et al., 2010), presentada en el año 2010, los expositores se enfocaron en identificar los problemas existentes en la integración efectiva de los RNFs con el modelo de desarrollo MDD, para ello los autores realizan un ejercicio académico en el cual plantean el desarrollo de un portal web para una agencia de viajes y donde muestran dos escenarios, el primero de ellos es una agencia de viajes que ofrece paquete vacacional en hoteles 5 estrellas, pero dicho portafolio cuenta con un número muy pequeño de asociados; el segundo escenario también se encuentra enfocado a una compañía mundial que ofrece paquetes de viajes, pero con la marcada diferencia que esta compañía si cuenta con un alto volumen de clientes. Los anteriores escenarios deberán ser construidos bajo modelo de desarrollo basado en enfoques MDSD. Los autores realizan un paralelo entre los

diferentes tipos de modelos existentes, el primero de ellos es cuando la plataforma es independiente del modelo (PIM), el segundo plataforma específica del modelo (PSM) y la tercera corresponde a la computación independiente de los modelos (CIM), los catedráticos muestran los conceptos necesarios a la hora de integrar los requisitos no funcionales (RNFs) dentro de los modelos MDD. La figura 1 muestra el orden jerárquico en el cual fue abordado el ejercicio académico del diseño de una aplicación web.

Figura1. Modelo de arquitectura propuesto en el ejercicio.



Luego de implementación web realizada para ambas compañías se logra determinar que es viable diseñar este tipo de aplicaciones con enfoques dirigidos por modelos, sin embargo, MDD está lejos de ser implementado en la industria y solo servirá como referencia para proyectos académicos, esto debido a su difícil integración con los requisitos no funcionales que exige el mercado actual.

Los desarrollos elaborados con MDSD no solo tiene aplicabilidad en portales web, tal es el caso de la implementación descrita por los docentes de la universidad Augsburg (Roser et al., 2008) donde muestran la implementación de un ERP (Enterprise Resource Planning) para PYMES utilizando enfoques MDSD; haciendo

uso de arquitecturas SOA (Services Oriented Architectures) la cual es flexible para soportar diferentes componentes de integración, además hacen uso de una plataforma de servicios web como WSBPEL la cual puede ser combinada con JavaEE.

Sin embargo, la implementación de software en enfoques dirigidos por modelos está estrechamente ligada al tamaño y la complejidad del mismo, es decir, a mayor tamaño del proyecto la implementación de modelos tales como MDSD se vuelve más complejo y costoso para su creación, esto repercute que cada vez que se integre un nuevo componente no funcional sea necesario realizar una modificación al código fuente, que en la gran mayoría de las veces afecta directamente el rendimiento y escalabilidad del aplicativo.

En la conferencia de ingeniería del software celebrada en el 2010 en Asia, los Ph.D Yi Liu y Zhiyi Ma, ambos profesores de la universidad de Pekin realizan un acercamiento al manejo de requisitos no funcionales de los desarrollos dirigidos por modelos desde la óptica del levantamiento de requisitos, ellos proponen en su conferencia tomar en primer lugar los modelamientos UML desde las propiedades funcionales de la aplicación para luego pasar al proceso de refinamiento de los requisitos no funcionales, sin embargo, esta propuesta de integrar los meta-modelos en función de los requisitos funcionales y no funcionales en la etapa del análisis, permite que en la fase de diseño se tenga una visión más clara del modelo de desarrollo, pero no garantiza que la implementación de estos sea tan adaptable a las solución del problema; el desacoplar la generación de código con la integración de los requisitos no funcionales no es un proceso tan transparente al momento de construir una aplicación, dado que los RNFs son necesidades tan puntuales de la organización, que en un alto porcentaje obligan que el desarrollo sea modificado en su código fuente para lograr ser adaptado a las solicitudes de los usuarios finales.

La implementación de requisitos no funcionales en aplicación construidas con enfoques dirigidos por modelos debe tener presente el dominio sobre el cual se desean aplicar los RFNs, es decir, requisitos como la eficiencia, usabilidad, operabilidad pueden ser incluidos desde la etapa de análisis y en los meta-modelos, a diferencia de requisitos como la exactitud, escalabilidad, interoperabilidad aumentan la probabilidad que las aplicaciones construidas con enfoques dirigidos por modelos deban ser modificados en su código fuente.

5 Conclusiones

La principal contribución de este artículo fue presentar una visión general sobre la viabilidad de implementar requisitos no funcionales en soluciones construidas con enfoques dirigidos por modelos, para ello se han tomado como referencias casos de éxito presentados en artículos o conferencias que centran la atención en este tema. De la revisión literaria realizada dentro de este artículo se observa que MDSD es un modelo que en la actualidad tiene su mayor participación en la academia dado que su aplicabilidad en la industria local no es muy acogido aun; sumado a ello la implementación de los RFNs hace que la complejidad y el costo de diseñar aplicaciones construidas bajo modelos MDSD, MDE y MDA sea cada vez más alta, motivo por el cual las empresas todavía no aceptan mucho el concepto de MDSD.

La viabilidad de integrar requisitos no funcionales en aplicaciones construidas con enfoques dirigidos por modelos es bastante alta, sin embargo, el alto costo en dinero y tiempo hace que no existan en el mercado aplicaciones construidas bajo la filosofía de MDSD, sumado a esto la poca masificación del concepto a nivel de la industria del software, hace que el número de aplicaciones desarrolladas por modelos sea mínima.

Para finalizar, MDSD desde el punto de vista teórico sería de gran utilidad y rentabilidad para las organizaciones, pero su poca visibilidad ha hecho que la implementación sea nula.

6 Referencias

- Alhaj, M., & Petriu, D. C. (2010). Approach for generating performance models from UML models of SOA systems 1 Introduction 2 Related Work, 268–282.
- Ameller, D., Franch, X., & Cabot, J. (2010). Dealing with Non-Functional Requirements in Model-Driven Development. *2010 18th IEEE International Requirements Engineering Conference*, 189–198. doi:10.1109/RE.2010.32
- Cortellessa, V., Informatica, D., Aquila, U., Marco, A. Di, Inverardi, P., & Aquila, U. (2006). Software Performance Model-Driven Architecture, (1-59593-108-2), 1218–1223.
- Goldschmidt, T., Reussner, R., & Winzen, J. (2008). A case study evaluation of maintainability and performance of persistency techniques. *Proceedings of the 13th international conference on Software engineering - ICSE '08*, 401. doi:10.1145/1368088.1368143
- Quintero, J., Rucisque, P., Anaya, R., & Piedrahita, G. (2012). How face the top MDE adoption problems. *2012 XXXVIII Conferencia Latinoamericana En Informatica (CLEI)*, 1–10. doi:10.1109/CLEI.2012.6427225
- Roser, S., Lautenbacher, F., & Bauer, B. (2008). MDSD light for ERP. *Proceedings of the 2008 ACM symposium on Applied computing - SAC '08*, 1042. doi:10.1145/1363686.1363930
- Mylopoulos, J., Chung, L., & Nixon, B. (2008). Representing and Using Nonfunctional Requirements : A Process-Oriented Approach, *IEEE 18(6)*, 483–497.
- Becker, M., Luckey, M., & Becker, S. (2012). Model-driven performance engineering of self-adaptive systems. *Proceedings of the 8th international ACM SIGSOFT conference on Quality of Software Architectures - QoSA '12*, 117. doi:10.1145/2304696.2304716