

Extendiendo la meta-arquitectura aportada por el enfoque MDA con conocimiento del dominio

Luis Roqué Fourcade, Liliana Arakaki, Daniel Riesco, Germán Montejano
Facultad de Ciencias Físico Matemáticas y Naturales - Universidad Nacional de San Luis
Ejército de Los Andes 950, San Luis, Argentina, +54 (0266) 4520300

email: araroq@unsl.edu.ar, liliana.arakaki@yahoo.com, driesco@unsl.edu.ar, gmonte@unsl.edu.ar

Resumen

Los métodos y técnicas provistos por la Ingeniería de Software hasta la actualidad, proveen un amplio soporte a la actividad de especificación de prácticamente todos los artefactos producidos por el proceso de Diseño de Software y, en contrapartida, un escaso soporte a la actividad característica y recurrente del proceso de Diseño de Software de síntesis y evaluación de alternativas para la toma de decisiones dejándola, paradójicamente para un proceso ingenieril, librada casi por completo al grado de experticia del desarrollador.

Presentamos aquí una línea de investigación que, a partir de resultados obtenidos en trabajos previos [1], se propone desarrollar una meta-arquitectura como una especialización de la aportada por el enfoque Model Driven Architecture (MDA) [2]. La línea propone realizar dicha especialización extendiendo esta meta-arquitectura con conocimiento de la definición de Diseño de Software aportada por trabajos previos [1]. La intención consiste en anticipar en ésta un conjunto de decisiones de diseño e incluir restricciones que acoten el universo de evaluación para otras, con el objetivo de soportar la actividad de síntesis y evaluación de alternativas para la toma de decisiones, mejorando sensiblemente el perfil ingenieril de la actividad de Diseño de Software.

Palabras clave: Ingeniería de software - Diseño – Meta-arquitectura - UML – Modelado - Transformación - MDA

Contexto

Esta línea de investigación, se desarrolla en el marco del proyecto 'Ingeniería de software: aspectos de alta sensibilidad en el ejercicio de la profesión de ingeniero de software', de la Facultad de Ciencias Físico-Matemáticas y Naturales de la Universidad Nacional de San Luis, bajo el número 22F222.

Introducción

Los avances incorporados por los métodos y técnicas actuales son realmente significativos. En el área de especificaciones, mediante lenguajes y estándares que soportan la especificación de todos los

artefactos de software involucrados en el proceso de desarrollo de software y, en el área de proceso, mediante la identificación y consideración de las actividades requeridas para construir dichas especificaciones. Sin embargo, en el área de proceso, se requieren avances aún más significativos. Esto es así porque estas especificaciones, describen solo aspectos estáticos de las actividades y adolecen de soportes más concretos a la ejecución propiamente dicha de la actividad. Esta situación se ve reflejada en múltiples aspectos de estos métodos y técnicas. En particular, la actividad característica y recurrente de síntesis y evaluación de alternativas para la toma de decisiones queda librada casi por completo al grado de experticia del desarrollador.

En la dirección de modelar y soportar el proceso de desarrollo de software, se destacan los esfuerzos y avances realizados por Object Management Group (OMG) [3] y, entre éstos, el enfoque MDA [2]. Son muchos y de gran importancia los aportes realizados por este enfoque, pero aquí estamos interesados en particular en el modelo del proceso de desarrollo de software que propone. Define una arquitectura para el proceso y un modelo de transformaciones constituyendo una meta-arquitectura a partir de la cual es posible obtener la arquitectura de múltiples ocurrencias del proceso de desarrollo de software. La gran importancia de este aporte es clara, sin embargo, adolece aún de un grado de abstracción que dificulta (o imposibilita) la derivación a partir de esta meta-arquitectura de soportes efectivos para actividades concretas de desarrollo de software como lo es por ejemplo la actividad de síntesis y evaluación de alternativas para la toma de decisiones que aquí nos ocupa.

Por este motivo y siguiendo con el enfoque propuesto en trabajos presentados previamente [1] [4] [5], proponemos aquí especializar esta meta-arquitectura extendiéndola con resultados de estos trabajos de manera tal que el proceso de instanciación de procesos de desarrollo de software sea capaz de derivar una arquitectura más especializada en el proceso de Diseño de Software,

en particular en la actividad de síntesis y evaluación de alternativas para la toma de decisiones. Para este fin, desarrollamos a continuación, los conceptos que fundamentan la propuesta y luego presentamos la hipótesis que fundamenta la línea de investigación sugerida.

Model Driven Architecture (MDA) [2]

Object Management Group (OMG) [3], es un consorcio internacional de estándares de tecnología, fundado en 1989, sin fines de lucro y de membresía abierta. Los estándares impulsados por OMG, que incluyen Unified Modeling Language (UML) [6] y Model-Driven Architecture (MDA) [2], hacen posible entornos poderosos de diseño visual, ejecución y mantenimiento de procesos de software y otras clases de procesos [3].

El enfoque MDA permite derivar valor de modelos y arquitectura para soportar el ciclo de vida completo de sistemas físicos, organizacionales y de tecnología informática, con el objetivo de ayudar a manejar complejidad e interdependencia en sistemas complejos. Apoyándose en la definición de conceptos como sistema, modelo, metamodelo, arquitectura y capa arquitectural, y en el hecho de que todos son modelos, estructura y ordena el proceso de desarrollo definiendo dos arquitecturas basadas en niveles de abstracción, desde puntos de vista diferentes pero ortogonales entre sí [7].

Arquitectura de Meta-modelado

El “meta-modelado” es clave en MDA. El enfoque MDA define para esta actividad una arquitectura basada en niveles de abstracción donde cada nivel define un nivel de meta-modelado más concreto que el superior y más abstracto que el inferior.

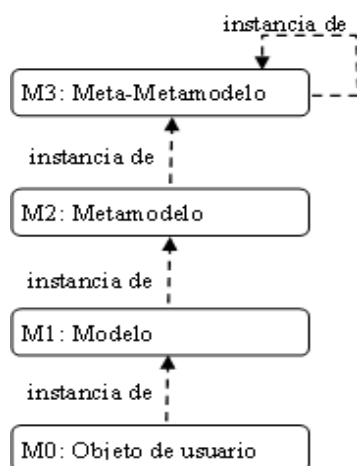


Figura 1. Arquitectura de Meta-modelado de MDA

En la *Figura 1. Arquitectura de Meta-modelado de MDA*, se puede ver esta arquitectura con los niveles de abstracción denominados M0, M1, M2 y M3. La estructura, términos, notaciones, sintaxis, semántica y reglas de integridad que son usados

para expresar un Modelo (M1), constituyen el lenguaje de modelado el cual, en su representación como modelo, es llamado Metamodelo (M2). Por esta razón un Modelo (M1) es considerado una instancia de un Metamodelo (M2). Dado que un Metamodelo (M2) es también un modelo, requiere también un lenguaje, un “Lenguaje de Meta-modelado” el cual, en su representación como modelo, recibe la denominación de Meta-Metamodelo (M3). Por esta razón un Metamodelo (M2) es considerado una instancia de un Meta-Metamodelo (M3). En el otro extremo, un Modelo (M1) también expresa un lenguaje de modelado, en este caso utilizado para expresar abstracciones de una realidad denominadas Objeto de Usuario (M0), razón por la cual una abstracción de una realidad, un Objeto de Usuario (M0), es considerada una instancia de un modelo (M1).

El lenguaje, en el nivel M3 (Meta-Metamodelo) utilizado para definir metamodelos, como por ejemplo el metamodelo UML, es MetaObject Facility (MOF) [8], cuya representación como modelo es un Meta-Metamodelo. A pesar de que es posible extender esta arquitectura con niveles más abstractos aún, para OMG no hay nuevo meta-nivel encima de MOF. Es decir que, en este punto, OMG decidió que el lenguaje utilizado para definir construcciones que especifiquen las reglas para la creación de ocurrencias en MOF, sea el propio MOF, el cual está definido como una extensión del metamodelo UML Core [8].

Arquitectura del Proceso de Desarrollo

Modelar el proceso de desarrollo, claramente constituye uno de los principales requisitos del proceso de maduración de la Ingeniería de Software. Los distintos modelos de ciclo de vida de desarrollo definidos a lo largo del proceso evolutivo han coincidido en algunas o todas y en el significado de un conjunto de categorías de actividades que han incorporado de diferentes maneras a sus respectivos modelos. Estas categorías de actividades son: modelado del problema, análisis del problema, diseño de la solución y construcción de la solución. La estrategia del enfoque MDA se apoya en considerar a los artefactos, resultantes de la ejecución de esas actividades del proceso de desarrollo, como modelos y en ordenar estos artefactos (modelos) en una arquitectura (dirigida por modelos) basada en niveles de abstracción.

En primer lugar, esta estrategia categoriza los modelos según su orientación en [7]:

- **Modelos de negocio o de dominio:** tienen por objetivo construir modelos del dominio asociados a la recolección de requisitos y son

históricamente denominados en OMG como Computational Independent Model (CIM).

- **Modelos de sistemas lógicos:** tienen por objetivo construir modelos lógicos orientados tanto al modelado del problema como del diseño de la solución y son categorizados como:
 - Modelos Independientes de la plataforma (PIM): orientados al modelado lógico del problema sin incluir decisiones vinculadas a la plataforma.
 - Modelos Dependientes de la plataforma (PSM): orientados al modelado del diseño de la solución en términos de la plataforma en la que se realizará el sistema.
- **Modelos de implementación:** tienen por objetivo modelar la forma en que un sistema o subsistema particular es implementado. Por estar ligados a una plataforma particular, frecuentemente son considerados como PSM. Sin embargo, algunos los consideran en una categoría específica, Modelos Específicos de Implementación (ISM).

Transformaciones de modelos en MDA

El enfoque MDA completa el modelo del proceso de desarrollo integrando los artefactos (modelos) y las actividades de desarrollo que los producen mediante el recurso de transformación. Estas transformaciones son definidas entre metamodelos para ser aplicadas a la transformación de uno o más modelos de entrada en otro de salida sin importar los niveles de abstracción definidos para la arquitectura del proceso de desarrollo. Dado que ambas arquitecturas son ortogonales, una transformación puede ser utilizada para producir una representación a partir de otra, o para cruzar capas arquitecturales.

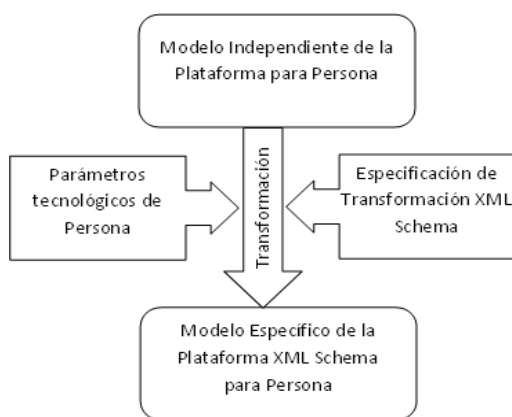


Figura 2. Ejemplo de Transformación Parametrizada [7]

Asumiendo las definiciones PIM y PSM introducidas en *Arquitectura del Proceso de Desarrollo*, la Figura 2. *Ejemplo de Transformación Parametrizada* presenta un ejemplo de un patrón de transformación, tomado de la guía “MDA Guide Version 2.0” [7], ilustra este concepto a través de los siguientes componentes:

- **Modelo independiente de la plataforma para Persona (PIM):** el modelo de información lógica representando información de persona.
- **Especificación de transformación XML Schema:** un componente reusable ejecutado por la transformación. Este componente es específico de XML Schema pero permite algunas opciones.
- **Parámetros tecnológicos de Persona:** una pequeña especificación que selecciona XML Schema como la tecnología deseada y proporciona cualquier parámetro requerido por el componente de transformación.
- **Transformación:** la transformación real que produce el PSM XML Schema para Persona.
- **Modelo específico de la plataforma XML Schema para Persona:** una representación XML Schema para Persona como está definida por el PIM.

Diseño de Software

Desde 2012 hasta la fecha, en el marco del proyecto, se desarrollan diferentes actividades de investigación orientadas a proveer más y mejores soportes al proceso de Diseño de Software. En particular, actividades desarrolladas en el marco de la línea de investigación presentada en la WICC 2015 [1], arrojaron como resultado una definición de Diseño de Software que heredó representatividad, estructura y organización, de la propuesta de definición del concepto de diseño formulada por Wand y Ralph [9]; y heredó especificidad en el dominio de Ingeniería de Software, de la propuesta de definición formulada por Buschmann et al. [10]. Además, extiende el concepto tradicional de Diseño de Software agregando más estructura y organización, incorporando conocimiento del enfoque de Patrones de Diseño de Software en la Arquitectura de Software resultante y proyectando luego esa estructura y organización en el proceso de Diseño de Software.

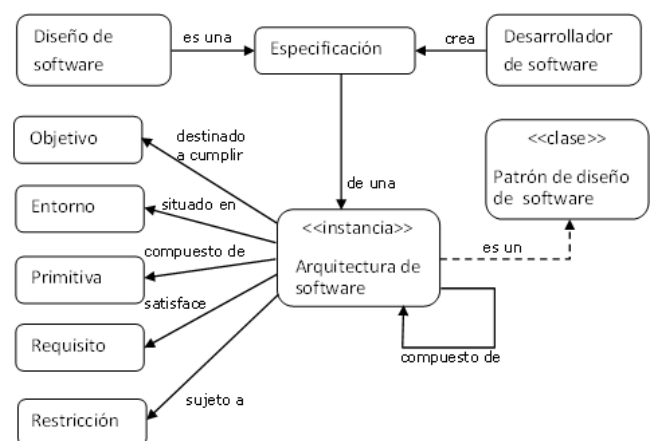


Figura 3. Modelo conceptual de diseño de software extendido con patrones [1] [9]

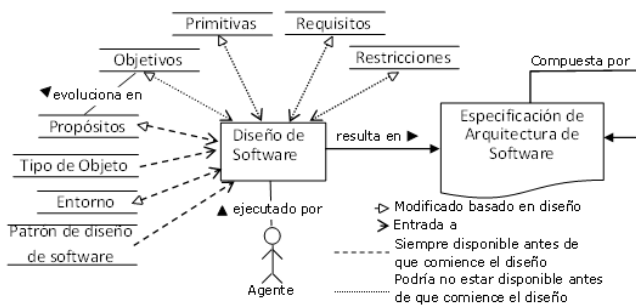


Figura 4. Nivel contextual del modelo conceptual de diseño extendido con patrones [1] [9]

Las dos figuras precedentes, presentan la definición utilizando un modelo conceptual y uno contextual de esta definición de Diseño de Software.

Como se observa en la *Figura 3. Modelo conceptual de diseño de software extendido con patrones*, la definición reconoce al objeto de diseño como Arquitectura de Software, pero además:

- La definición idealmente supone que cada parte que integra la Arquitectura de Software de una aplicación, puede ser obtenida como una instancia de un Patrón de Diseño de Software. Si bien no es ésta la realidad del estado actual, ni hay certeza de que lo vaya a ser en un estado futuro, sí hay importante evidencia de su efectividad en un buen número de casos. Por lo tanto, asumir sólo este tipo de casos, simplifica el análisis y no perjudica las conclusiones del trabajo, por el contrario, favorece su desarrollo despejando estos casos de interés y permitiendo la concentración en ellos sin comprometer la validez de los resultados. La pretensión aquí precisamente, consiste en distinguir el fuerte grado de dependencia (ser instancia de) del proceso de Diseño de Software respecto de Patrones de Diseño de Software, en comparación con el resto de las dependencias.
- La estructura que la definición reconoce en el objeto de diseño como Arquitectura de Software, es reforzada con la asociación 'compuesto de', la cual indica la capacidad que las Arquitecturas de Software tienen para componerse entre sí en la arquitectura del sistema. Un aspecto notable es la coincidencia de esta capacidad con la capacidad equivalente de composición entre patrones.

En la *Figura 4. Nivel contextual del modelo conceptual de diseño extendido con patrones*, este conocimiento de Patrones de Diseño de Software es inyectado como parámetro en el proceso influenciando la toma de decisiones requerida para producir la especificación de la Arquitectura de Software.

Línea de Investigación y Desarrollo

Como anticipamos en la *Introducción*, OMG a través del **Model Driven Architecture (MDA)**, aporta una meta-arquitectura para el proceso de desarrollo que cubre virtualmente la mayoría de las instancias de procesos de desarrollo posibles pero que adolece aún de un grado de abstracción que dificulta (o imposibilita) la derivación de soportes efectivos para actividades concretas de desarrollo de software como lo es la de síntesis y evaluación de alternativas para la toma de decisiones en el proceso de *Diseño de Software*.

Por este motivo presentamos aquí una línea de investigación la cual se propone especializar esta meta-arquitectura extendiéndola con resultados obtenidos en trabajos previos [1] [4] [5], de manera tal que el proceso de instanciación sea capaz de derivar arquitecturas más especializadas en el proceso de *Diseño de Software*.

La línea de investigación se fundamenta en la hipótesis de que si disponemos de una definición de *Diseño de Software* que incorpore la influencia de una representación estándar de conocimiento del dominio en la actividad de síntesis y evaluación de alternativas para la toma de decisiones y además, el tipo del resultado del proceso, define para éste una estructura compatible con esa definición, podemos extender la meta-arquitectura con este conocimiento, de manera tal que la instanciación de procesos de desarrollo de software sea capaz de derivar arquitecturas más especializadas, en particular en la actividad de síntesis y evaluación de alternativas para la toma de decisiones.

Presentamos a continuación, una versión extendida del recurso de *Transformaciones* aportado por el **Model Driven Architecture (MDA)** el cual expresa la hipótesis de trabajo de la línea de investigación propuesta.

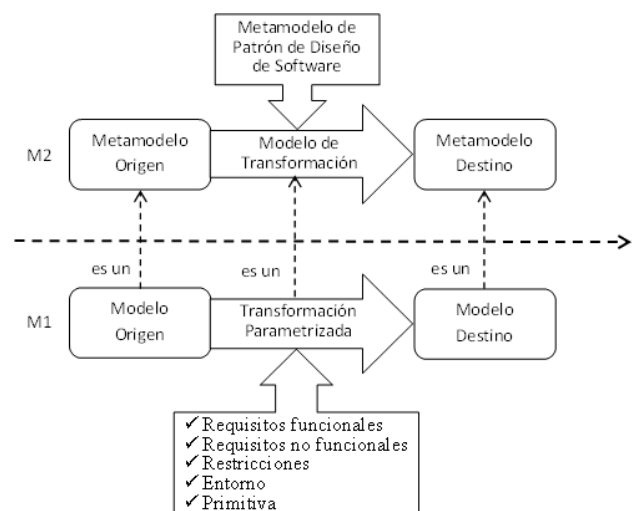


Figura 5. Modelo extendido de transformación

Resultados y Objetivos

En la *Figura 5. Modelo extendido de transformación*, se ve el proceso de transformación claramente influenciado por el conocimiento de la definición de *Diseño de Software* que hemos referido.

Son variados e interesantes los trabajos que se desarrollan en el marco de la línea de investigación y desarrollo presentada, muchos de los cuales se encuentran en curso en forma de trabajos de tesis de grado, trabajos de tesis de maestría, trabajos de publicación de resultados y trabajos de laboratorio en las carreras de Licenciatura en Ciencias de la Computación y de Ingeniería de Sistemas de la Universidad Nacional de San Luis. Entre estos trabajos podemos destacar:

- 1) Finalización de una tesis de la Maestría en Ingeniería de Software de la Universidad Nacional de San Luis, a ser presentada en los próximos meses, que presenta una propuesta integral de una meta-arquitectura que sintetiza la mayoría de los resultados hasta aquí obtenidos
- 2) En el presente año se inician trabajos de tesis en las carreras de Licenciatura en Ciencias de la Computación y de Ingeniería de Sistemas que se ocupan del desarrollo de mecanismos de instanciación Arquitecturas de Software a partir de Patrones de Diseño de Software basados en técnicas Orientadas a Objetos en un caso y en técnicas de XML Schemas en otro.
- 3) Este año se realizarán publicaciones de resultados de trabajos que se ocupan de la investigación de propiedades acerca de la relación de instancia entre Patrones de Diseño de Software y las Arquitecturas de Software, como resultado de efectuar extrapolaciones a dominios como Orientación a Objetos, XML Schema y Gramáticas de Lenguajes.
- 4) Actualmente se encuentra en curso en el marco de la línea de investigación y desarrollo trabajos que tienen por objetivo:
 - a. El desarrollo del metamodelo para Patrones de Diseño de Software.
 - b. El desarrollo un XML Schema para la representación XMI de instancias del metamodelo de Patrones de Diseño de Software
 - c. El diseño de un módulo reutilizable por las transformaciones para extenderlas con la capacidad de procesar representaciones del metamodelo de Patrones de Diseño de Software a la luz de información del dominio recibida como parámetro u obtenida por

medio de consultas a otros objetos del dominio

Para el presente año, el objetivo es recolectar e integrar resultados obtenidos por la finalización de algunos de los trabajos en curso o de resultados intermedios de los mismos y plasmarlos en publicaciones correspondientes al próximo año.

Formación de Recursos Humanos

En el área de la temática planteada, se desarrollan diferentes actividades que van desde la inclusión de trabajos prácticos, laboratorios y ejercicios de investigación, en materias de las carreras afines hasta el desarrollo de tesis y trabajos finales, algunos finalizados y otros en curso, en las mismas.

En particular, durante el año 2015, finalizó una tesis de maestría que experimentó con este enfoque y que será presentada en los próximos meses.

Referencias

- [1] L. Roqué Fourcade y L. Arakaki, Soporte a la actividad de diseño basado en patrones de diseño, XVII WICC, Salta, 2015.
- [2] Object Management Group, MDA - The Architecture of Choice for a Changing World, <http://www.omg.org/mda>.
- [3] Object Management Group, About OMG, <http://www.omg.org/gettingstarted/gettingstartindex.htm>.
- [4] L. Roqué Fourcade y L. Arakaki, Derivando el diseño a partir de especificaciones de requisitos basadas en Casos de Uso, XIV WICC, Posadas, Misiones, Argentina, 2012.
- [5] L. Roqué Fourcade, Derivando el Diseño a Partir de Especificaciones de Requisitos Basadas en Casos de Uso, 13th ASSE, La Plata, Buenos Aires, Argentina, 2012.
- [6] O. M. Group, Documents associated with Unified Modeling Language™ (UML®), v2.4.1, <http://schema.omg.org/spec/UML/2.4.1>.
- [7] Object Management Group, OMG Document -- ormsc/14-06-01 (MDA Guide revision 2.0), <http://www.omg.org/cgi-bin/doc?ormsc/14-06-01>.
- [8] Object Management Group, OMG's MetaObject Facility, <http://www.omg.org/mof/>.
- [9] P. Ralph y Y. Wand, de A Proposal for a Formal Definition of the Design Concept, Design Requirements Workshop, Berlin, 2009.
- [10] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad y M. Stal, Pattern-Oriented Software Architecture: A System of Patterns, Chichester, England: Wiley, 2001.