

## Transformaciones y Desarrollos Dirigidos por Modelos

Ana Funes, Aristides Dasso,  
Universidad Nacional de San Luis, Ejército de los Andes 950  
5700 San Luis, Argentina  
{afunes, aridas}@unsl.edu.ar

### Resumen

El objetivo principal de la línea de investigación aquí presentada es el estudio y definición de transformaciones, tanto horizontales como verticales, de modelos de software, así como el desarrollo de aplicaciones, siguiendo los lineamientos de la propuesta Model Driven Architecture (MDA).

### Contexto

El presente trabajo de investigación se viene llevando a cabo dentro del SEG (Software Engineering Group), en el ámbito de la Universidad Nacional de San Luis. Se encuentra enmarcado dentro de una de las líneas de investigación del Proyecto de Incentivos código 22/F822 "Ingeniería de Software: Aspectos de alta sensibilidad en el ejercicio de la profesión de Ingeniero de Software" (Director: Daniel Riesco, Co-Director: Roberto Uzal. Acreditado con evaluación externa para el período 1/1/2012 - 31/12/2014. Financiamiento: Universidad Nacional de San Luis).

**Palabras clave:** Transformaciones de modelos, MDA, QVT, ATL, MDE, DSL.

### Introducción

La Ingeniería Dirigida por Modelos, más conocida por la sigla MDE (Model Driven Engineering) es un enfoque de desarrollo de software que se basa en el uso de modelos de software como forma primaria de expresión con el objetivo de desarrollar,

mantener y hacer evolucionar el software por medio de transformaciones de modelos [10], [30].

Cuando construimos modelos y herramientas de modelado, es necesario conocer la estructura y las reglas de buena formación del lenguaje en el cual los modelos serán expresados. Tales modelos son llamados metamodelos. Contar con metamodelos precisos es un prerequisite fundamental para poder llevar a cabo transformaciones automáticas entre modelos así como para poder definir Lenguajes de Dominio Específico (Domain Specific Language, DSL). En este sentido, MDE combina los siguientes conceptos:

- Lenguajes de Dominio Específico, que formalizan la estructura de la aplicación, el comportamiento y los requisitos dentro de un dominio particular. Estos lenguajes son descritos usando metamodelos, los cuales definen las relaciones existentes entre los elementos de un dominio.
- Motores de transformación y generadores. Analizan ciertos aspectos de los modelos y después crean varios tipos de artefactos, tales como código fuente, entradas de simulación, descripciones de uso XML o representaciones alternativas de dichos modelos.

Mediante los DSL se consiguen notaciones de modelado distintas para cada tipo de sistema, las cuales se encuentran definidas formalmente por medio de su metamodelo. De esta manera, el ingeniero de software cuenta con herramientas específicas para cada tipo de sistema, lo cual le permite modelarlos de una manera

más detallada y precisa de acuerdo al dominio al que pertenecen.

Mediante los motores de transformación se facilita la evolución de estos modelos, pudiendo llevar a cabo transformaciones de unos a otros, según reglas de transformación.

El consorcio OMG ha desarrollado la propuesta Arquitectura Dirigida por Modelos (Model Driven Architecture, MDA) [20] como una implementación de MDE. MDA nace con la idea establecida de separar, en el sistema, por un lado la especificación y por el otro la lógica operacional; es decir, separar la especificación del sistema de los detalles que definen cómo el sistema usa las capacidades de la plataforma tecnológica donde es implementado.

Por lo tanto, el desarrollador sólo se preocupa de la lógica del negocio mientras que las herramientas específicas generan todo el código relacionado con las plataformas de implementación.

Un elemento crucial en MDA es el uso de la tecnología MOF (Meta Object Facility) [21] y la definición de metamodelos que sean instancias del meta-metamodelo MOF. Un metamodelo es un modelo para definir modelos, que nos dice cuáles son los elementos que podemos usar para construir nuestros modelos así como cuáles son las relaciones existentes entre dichos elementos. Cada uno de estos metamodelos define un DSL, que presenta una solución al modelado de distintos tipos de sistemas de software. Así, por ejemplo, existe el metamodelo UML [24] para modelar la arquitectura de sistemas discretos orientados a objetos, o el metamodelo SPEM [28] para modelar procesos de software.

Junto a la automatización y al empleo de lenguajes de modelado, el uso de estándares es el tercer pilar en el que se sustenta MDA. En ese sentido, SBVR (Semantics of Business Vocabulary and Business Rules) [22] y QVT (Query/View/Transformation) [23] son dos de los estándares propuestos

por OMG que son una parte integral de MDA.

SBVR es un estándar de OMG creado con el objeto de ser usado para expresar requisitos del negocio o del software en la forma de reglas. Provee un metamodelo estándar para capturar el conocimiento contenido en una regla del negocio. Esto permite definir notaciones que pueden ser traducidas sin ambigüedad a otras instancias del metamodelo, además de permitir la creación de herramientas que soporten la creación, validación y administración de reglas del negocio así como dar soporte a los expertos en IT en convertir reglas del negocio a reglas de implementación para sistemas automatizados.

Si bien QVT es el lenguaje estándar que OMG propone para la definición de transformaciones de modelos, existen otras propuestas tales como Atlas Transformation Language (ATL) [4]. ATL es un lenguaje ampliamente usado y de libre distribución que adhiere al RFP (Request for Proposal) de OMG para MOF/QVT [29]. Sirve para definir transformaciones entre modelos basadas en sus metamodelos, permitiendo que una instancia cualquiera de un metamodelo origen pueda ser transformado en una instancia del metamodelo destino.

En este trabajo, presentamos una línea de investigación en la que, adhiriendo a los principios de MDA y haciendo uso de los estándares antes resumidos, nos abocamos al estudio y definición de diversas transformaciones así como al desarrollo de aplicaciones con el objeto de agilizar el proceso de desarrollo de software.

### **Líneas de Investigación y Desarrollo**

Dentro de la línea de investigación aquí presentada, varias vertientes de trabajos se encuentran en curso:

- 1) *Ingeniería de Requerimientos: De los casos de uso del sistema a una especificación formal.* Sobre la base de

trabajo previo llevado a cabo por integrantes de esta línea de investigación y desarrollo [11], en el cual se propuso una aproximación para la obtención de una especificación formal inicial de la funcionalidad de un sistema a partir de sus casos de uso, en esta línea apuntamos a definir una transformación horizontal que haga uso de SBVR como lenguaje natural controlado en lugar de hacer uso del lenguaje natural tal como fue propuesto en [11].

De esta manera los eventos externos del sistema asociados a cada uno de los casos de pueden ser descriptos por medio de hechos en SBVR, permitiendo que dichos hechos sean representados en diferentes notaciones pero con una semántica única dada por SBVR.

Dicha transformación se hará haciendo uso del lenguaje ATL, definiendo las reglas de transformación necesarias para producir una especificación formal inicial en RSL a partir de los eventos asociados a los casos de uso expresados como hechos en SBVR.

2) *Validación y verificación del modelo conceptual de un sistema.* Existen muchos trabajos en la literatura que permiten, sobre la base de un modelo conceptual expresado por medio de un diagrama de clases UML e invariantes en OCL, llevar adelante la verificación de las propiedades deseables del modelo [11][12][13][14].

Por otro lado, tan importante como la verificación del modelo es la validación del mismo con el cliente. Para facilitar dicha tarea, creemos que es de utilidad contar con una herramienta que implemente una transformación horizontal para la obtención de reglas expresadas en español estructurado SBVR. Dicha transformación se definirá haciendo uso de ATL.

3) *Transformación de modelos de procesos del negocio.* Esta línea es la continuación de una línea de trabajo (ver [1], [2], [3]) en la cual definimos una transformación horizontal desde modelos de procesos de negocio dados por medio de diagramas de actividades en UML (UML AD) [24] a

modelos expresados en BPMN [26] y otra transformación del modelo BPMN a XPDL [31]. En particular, para el caso de la transformación UML AD a BPMN fue necesario crear un metamodelo para BPMN, el cual fue construido a partir de descripciones verbales de la notación.

A la luz de la experiencia ganada en este trabajo, una extensión importante al trabajo original es la definición de reglas de transformación que permitan la traducción no sólo de UML AD y BPMN sino también de otros lenguajes de modelado de procesos de negocio a un metamodelo independiente de la notación, sirviendo este metamodelo como intermediario entre las dos notaciones diferentes en la perspectiva PIM.

Una característica que deberían tener las nuevas transformaciones de metamodelos es la capacidad de ser bi-direccionales. Esta aproximación promete ser más flexible de cara a la incorporación de una nueva notación, ya que sólo haría falta establecer sus correspondencias con el metamodelo independiente de la notación para que pueda ser utilizada como modelo de origen o destino en el marco definido. El metamodelo que nosotros proponemos como candidato a cumplir el rol de metamodelo independiente de la notación es el BPDM [27] del grupo OMG.

4) *Creación y evaluación de modelos LSP en un contexto MDA.* Aquí nos abocamos al desarrollo de una herramienta de software que facilite la creación y evaluación de modelos de sistemas, definidos de acuerdo al DSL del método LSP [5][6][7]. Para esto, fue necesario definir nuestro DSL o metamodelo LSP que nos permitiese especificar cuáles son los elementos con los que contamos para construir modelos LSP, así como cuáles son las relaciones existentes entre dichos elementos de modelado.

Dicha herramienta no sólo permitirá la creación de diversos modelos LSP (nivel M1 en la jerarquía MOF) como instancias del metamodelo LSP (nivel M2 en MOF) sino que también deberá permitir la

instanciación a nivel M0 para cada sistema que se desee evaluar.

### Resultados y Objetivos

Este trabajo de investigación tiene un doble objetivo. Por un lado, se espera poder definir formalmente, haciendo uso de ATL, las transformaciones mencionadas en las sublíneas antes descritas y desarrollar herramientas que den soporte a dichas transformaciones dirigidas por modelos. Por otro lado, completar el desarrollo de la herramienta para la creación y evaluación de modelos LSP.

### Formación de Recursos Humanos

La línea de investigación relativa a transformaciones dirigidas por modelos ha dado lugar a una tesis de maestría en Ingeniería de Software [3] defendida en el ámbito de la Universidad Nacional de San Luis. Asimismo, otra tesis de maestría en Ingeniería de Software se encuentra en desarrollo sobre la línea 4.

### Referencias

- [1] Mauro Argañaraz, Ana Funes, Aristides Dasso. "An MDA Approach to Business Process Model Transformations" en "Software Engineering in Argentina: present and future trends. Extended version of selected papers ASSE 2009", Electronic Journal SADIO (EJS), Volumen 9, Numero 1, 2010.
- [2] Mauro Argañaraz, Ana Funes, Aristides Dasso. "An MDA Approach to Business Process Model Transformations", ASSE 2009 (39 JAIIO), 30/8/2010 al 3/9/2010, Buenos Aires, Argentina.
- [3] Mauro Cesar Argañaraz, "Uso de transformaciones de modelos para la definición de procesos de negocio", Tesis de "Maestría en Ingeniería de Software", Directores: Ana Funes, Aristides Dasso, Universidad Nacional de San Luis. Res. N° 687-09.
- [4] ATLAS Transformation Language <http://www.eclipse.org/atl/>
- [5] Jozo J. Dujmovic, "Continuous Preference Logic for System Evaluation", IEEE Transactions on Fuzzy Systems, Vol. 15, N° 6, December 2007.
- [6] Jozo Dujmovic, "Characteristic forms of generalized conjunction/disjunction"; En Fuzzy Systems, 2008 (FUZZ-IEEE 2008). (IEEE World Congress on Computational Intelligence). 1-6 June 2008, pp. 1075 – 1080, ISSN: 1098-7584, E-ISBN: 978-1-4244-1819-0, Print ISBN: 978-1-4244-1818-3.
- [7] Jozo Dujmovic, "A Method for Evaluation and Selection of Complex Hardware and Software Systems", The 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise Computing Systems. CMG96 Proceedings, vol. 1, pp.368-378, 1996.
- [8] Eclipse Modeling Framework Project (EMF). <http://www.eclipse.org/modeling/emf/>. Ultimo acceso: 21/12/2011.
- [9] Eclipse Project. Graphical Modeling Framework (GMF). <http://www.eclipse.org/modeling/gmf/>. Ultimo acceso: 22/06/2011.
- [10] J.M. Favre, Towards a basic theory to model model driven engineering. In: Proc. 3rd Workshop in Software Model Engineering (Satellite workshop at the 7th International Conference on the UML), 2004.
- [11] Ana Funes y Aristides Dasso, An Integration of Semi formal and Formal Specifications: From Use Cases to RSL Signatures, ASSE 2012 (41 JAIIO), 30 y 31 agosto 2012, La Plata, Argentina.
- [12] Ana Funes y Chris George, Capítulo 8: "Formalizing UML class diagrams" en "UML and the Unified Process", ISBN 1931777446, Idea Group Publishing; Abril 2003. pp 129-198.
- [13] Narayan Debnath, Ana Funes, Aristides Dasso, Germán Montejano, Daniel Riesco, Roberto Uzal, "Integrating OCL Expressions into RSL Specifications", IEEE-EIT 07, May 17-20, 2007, Chicago, IL, USA.
- [14] Ana Garis, Alcino Cunha, Daniel Riesco, Translating Alloy Specifications to UML Class Diagrams Annotated with OCL, SEFM 2011, LNCS 7041, pp 221-236, Springer, 2011
- [15] Graphical Editing Framework (GEF). [http://wiki.eclipse.org/index.php/Graphical\\_Editing\\_Framework](http://wiki.eclipse.org/index.php/Graphical_Editing_Framework). Ultimo acceso: 21/12/2011.
- [16] IBM, Create an Eclipse-based application using the Graphical Editing Framework. <http://www.ibm.com/developerworks/opensource/library/os-gef/>. Ultimo acceso: 21/12/2011.

- [17] S. Mellor, K. Scott, A. Uhl, MDA Distilled, Principles of Model Driven Architecture, Addison-Wesley Professional, 2004, ISBN 0-201-78891-8.
- [18] T. Mens, P. Van Gorp, A Taxonomy of Model Transformation. Electronic Notes in Theoretical Computer Science, Volume 152, 27 March 2006, Pages 125-142.
- [19] Object Management Group, Model-Driven Architecture: Vision, Standards And Emerging Technologies.  
[http://www.omg.org/mda/mda\\_files/Model-Driven\\_Architecture.pdf](http://www.omg.org/mda/mda_files/Model-Driven_Architecture.pdf). Ultimo acceso: 21/12/2011.
- [20] Object Management Group, Model Driven Architecture (MDA)  
<http://www.omg.org/mda/index.htm>
- [21] Object Management Group, Meta Object Facility (MOF) v.2.4.1 (2013)  
[http://www.omg.org/mof/..](http://www.omg.org/mof/)
- [22] Object Management Group: Semantics of Business vocabulary and Rules (SBVR), OMG Standard, v. 1.2 (2013),  
<http://www.omg.org/spec/SBVR/1.2/index.htm>
- [23] Object Management Group: Query/View/Transformation (QVT), v. 1.1 (2011), <http://www.omg.org/spec/QVT/>
- [24] Object Management Group: . Unified Modeling Language, Version 2.4.1 (2011)  
[http://www.omg.org/spec/UML/.](http://www.omg.org/spec/UML/)
- [25] Object Management Group: The Object Constraint Language Specification. Version 2.4 (2014), [http://www.omg.org/spec/OCL/2.4/.](http://www.omg.org/spec/OCL/2.4/)
- [26] Object Management Group: Business Process Modeling Notation Specification. version 2.0.2 (2013), <http://www.omg.org/spec/BPMN/2.0.2/>
- [27] Object Management Group: Business Process Definition MetaModel (BPDM), Version 1.0, (2008). <http://www.omg.org/spec/BPDM/1.0/>
- [28] Object Management Group: Software & Systems Process Engineering Metamodel specification (SPEM). Version 2.0 (2008)  
<http://www.omg.org/spec/SPEM/2.0/>
- [29] Object Management Group: MOF 2.0 Query / Views / Transformations RFP (2002)  
<http://www.omg.org/cgi-bin/doc?ad/2002-4-10>
- [30] D.C. Schmidt, "Model-Driven Engineering". IEEE Computer 39 (2). Febrero 2006
- [31] XPDL: XML Process Definition Language,  
<http://www.xpdl.org/>