

Transformación del Modelo de Negocio al Modelo de Caso de Uso del Sistema Utilizando QVT

Ariel S. Arsaut¹, Marcela Daniele², Fabio A. Zorzan³, Daniel Riesco⁴

RESUMEN

Esta línea de investigación contribuye a la mejora del proceso de desarrollo de software, en el marco de Model Driven Architecture (MDA), en particular los especificados con el Proceso Unificado del proceso de software. La transformación propuesta tiene como objetivo generar una especificación de modelo de Casos de Uso de sistema a partir del modelo de negocio. Esto se logrará a través de la definición de una transformación en Relations, que forma parte de QVT, entre los metamodelos del modelo de negocio y el metamodelo del modelo de Casos de Uso del Sistema. De esta forma se dará la posibilidad de transformar automáticamente un Modelo de Negocio a su respectivo Modelo de Casos de Uso del sistema.

CONTEXTO

La línea de investigación presentada en este trabajo se desarrolla en el marco del proyecto “Ingeniería de Software: Automatización de Procesos de Desarrollo de Software”, presentado en la convocatoria 2009 para Proyectos y Programas de Investigación (PPI) de la secretaría de Ciencia y Técnica de la Universidad Nacional de Río Cuarto.

Palabras clave— Modelo de Negocio, Modelo de Casos de Uso del Sistema, MDA, QVT, Relations.

1 INTRODUCCIÓN

Un proceso de desarrollo de software es un método que sirve para organizar las actividades de análisis, creación y mantenimiento de un producto de software. La definición de requisitos del sistema de software es la primera actividad definida por cualquier modelo de desarrollo de software. El Proceso Unificado [1] es una metodología de desarrollo de software que sugiere para la primera etapa, la construcción de dos importantes modelos, el Modelo de Negocio y el Modelo de Casos de Uso del sistema. El primer modelo mencionado tiene por finalidad establecer una abstracción de la organización, y sobre la base de este conocimiento se construye el siguiente modelo para especificar los casos de uso y actores del sistema.

Lo que se propone en este trabajo es, en el marco de MDA [2], automatizar mediante transformaciones, la construcción del Modelo de Casos de Uso del sistema a partir del Modelo de Negocio.

Para lo cual necesitamos, como primer medida, la especificación del Modelo de Negocio (Metamodelo de Negocio) y también la especificación del Modelo de Caso de Uso del Sistema, para luego poder especificar la transformación QVT entre ambos metamodelos.

Para construir un modelo es importante elegir un lenguaje para su representación.

¹ Ariel S. Arsaut pertenece al Departamento de Computación de la Universidad Nacional de Río Cuarto, Argentina; tel.: +54-358-4676529; e-mail: aarsaute@exa.unrc.edu.ar.

² Marcela Daniele pertenece al Departamento de Computación de la Universidad Nacional de Río Cuarto, Argentina; tel.: +54-358-4676529; e-mail: marcela@dc.exa.unrc.edu.ar.

³ Fabio A. Zorzan pertenece al Departamento de Computación de la Universidad Nacional de Río Cuarto, Argentina; tel.: +54-358-4676529; e-mail: fzorzan@exa.unrc.edu.ar.

⁴ Daniel Riesco pertenece al Departamento de Informática de la Universidad Nacional de San Luis, San Luis, Argentina, tel.: +54-2652-424029 int 251, e-mail: driesco@unsl.edu.ar

Visualizar, especificar, construir y documentar sistemas orientados a objetos es exactamente el propósito de UML (Unified Modeling Language).[3]

En un trabajo previo relacionado [4], podríamos considerar al Modelo Genérico de Negocio y su definición (Metamodelo de Modelo de Negocio), cómo punto de partida (metamodelo fuente) para la propuesta de transformación de modelos que pretendemos realizar.

En el trabajo mencionado se muestra la definición un modelo genérico del Modelo de Negocio y un conjunto de reglas, que colaboran con el desarrollador de software en la creación de un modelo de negocio particular de una organización.

La transformación, cómo propuesta de automatización, que se plantea en este trabajo nos permitirá obtener una especificación de modelo de Casos de Uso de sistema a partir del Modelo de Negocio.

Para establecer esta transformación utilizaremos la definición de una transformación en Relations, que forma parte de QVT [5] (Query/View/Transformation), entre los metamodelos del Modelo de Negocio y el metamodelo del Modelo de Casos de Uso del Sistema.

Cabe destacar que dentro del grupo de investigación existe importante experiencia, como resultado de varios trabajos [6], en donde se han desarrollado transformaciones QVT en el marco de MDA.

El resultado de aplicar la transformación nos permitirá, partiendo de un Modelo de Casos de Uso de Negocio concreto obtener su respectivo Modelo de Casos de Uso del sistema.

1.1 ELEMENTOS BÁSICOS

1.1.1 ARQUITECTURA DIRIGIDA POR MODELOS.

La Arquitectura Dirigida por Modelos (“Model Driven Architecture o MDA”) es un acercamiento al diseño de software propuesto por el “Object Management Group (OMG)”

en el año 2001. MDA [7] se ha concebido para dar soporte a la ingeniería dirigida a modelos de los sistemas software, cuyo objetivo central es resolver el problema de que el cambio de tecnología de un sistema software no sea costoso, así como su integración en la plataforma que corresponda. La idea principal de MDA es usar modelos, de modo que las propiedades y características de los sistemas queden plasmadas de forma abstracta, y por tanto, los modelos no se vean afectados por los cambios tecnológicos.

MDA proporciona una serie de guías o patrones expresadas como modelos. MDA propone cuatro niveles de abstracción que componen la jerarquía o arquitectura de modelos. Estos son: Modelo independiente de cómputo CIM (Computation Independent Model), Modelo independiente de la plataforma PIM (Platform Independent Model), Modelos específicos de la plataforma PSM (Platform Specific Model), y la aplicación final.

Los modelos CIM se usan para describir el entorno en el que se usará el sistema, sin referencia directa a la forma en la que se implementará el sistema [8].

Los modelos PIM se usan para modelar la funcionalidad y la estructura del sistema de información sin considerar los detalles tecnológicos de la plataforma en la cual se implementará el sistema [8].

Los modelos PSM se usan para describir los modelos específicos de plataforma, concretamente de la plataforma tecnológica donde se ejecutará el sistema.

Para conseguir los beneficios de MDA plantea el siguiente proceso de desarrollo: de los requisitos se obtiene un modelo independiente de la plataforma (PIM), luego este modelo es transformado con la ayuda de herramientas en uno o más modelos específicos de la plataforma (PSM), y finalmente cada PSM es transformado en código. Por lo tanto, MDA incorpora la idea de transformación de modelos (PIM a PSM, PSM a código) y se necesitan herramientas para automatizar esta tarea. Estas

herramientas constituyen uno de los elementos básicos de MDA

1.1.2 *MODELO DE NEGOCIO.*

El Modelo de Negocio es una parte muy importante del ciclo de vida de desarrollo de software, que claramente ayuda a definir los requerimientos del sistema.

El equipo de proyecto define los requerimientos del sistema a partir del análisis del problema a resolver en el contexto del negocio, de manera que esto ayude a asegurar que el sistema que se construirá se adecua a los objetivos del negocio. En efecto, el modelo de negocio puede ser una entrada directa al modelo de requerimientos del sistema y es el punto de partida fundamental para los siguientes modelos en el ciclo de vida del desarrollo del software.

La finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas de negocio.

Un proceso de negocio se define como un conjunto de tareas lógicamente relacionadas, ejecutadas para obtener un resultado de negocio. Los procesos de negocio pueden ser controlados y administrados por un sistema basado en software.

Los modelos que constituyen el Modelo del Negocio son: el Modelo de Casos de Uso del negocio y el Modelo de Objetos del Negocio [4], llamado actualmente Modelo de Análisis del Negocio, y a partir de ellos, se evoluciona a los restantes modelos que componen el ciclo de vida de desarrollo de un sistema.

1.1.3 *MODELO DE CASOS DE USO DEL SISTEMA.*

La primer disciplina que se desarrolla dentro de cada iteración es la de requerimientos (posiblemente luego de realizar un modelado del dominio o del negocio). El objetivo de esta fase es determinar los requerimientos del sistema. Los requerimientos funcionales son

plasmados a través de casos de uso en un Modelo de Casos de Uso.

El modelo de casos de uso ayuda al cliente, a los usuarios, y a los desarrolladores a llegar a un acuerdo sobre cómo utilizar el sistema.

Cada tipo de usuario del sistema se representa mediante un actor que define un rol de utilización del sistema. [1].

Los Artefactos que componen el Modelo de Casos de Uso son: Actores del Sistema, Casos de Uso del Sistema, Diagrama de Casos de Uso del Sistema y Descripción de Casos de Uso del Sistema.

Los actores modelan el entorno del sistema, y los casos de uso especifican el sistema.

Un diagrama de casos de uso describe parte del modelo de casos de uso y muestra un conjunto de casos de uso y actores asociados.

Cada Caso de Uso tiene una descripción que especifica la funcionalidad que se incorporará al sistema propuesto.

1.1.4 *QVT (Query/View/Transformation)*

El planteamiento QVT se basa principalmente en la definición de un lenguaje para las consultas (Queries) sobre los modelos Meta Object Facility (MOF) [9], la búsqueda de un estándar para generar vistas (Views) que revelen aspectos específicos de los sistemas modelados, y finalmente, la definición de un lenguaje para la descripción de transformaciones (Transformations) de modelos MOF.

En este trabajo se utiliza el componente de QVT que tiene como objetivo definir transformaciones. Estas transformaciones describen relaciones entre un metamodelo fuente F y un metamodelo objetivo O, ambos metamodelos deben estar especificados en MOF. Luego esta transformación definida se utiliza para obtener un modelo objetivo, el cual es una instancia del metamodelo O, a partir de un modelo fuente que es una instancia del metamodelo F. Una característica muy importante de estas transformaciones es que pueden ser bidireccionales (multidimensionales también).

1.1.4.1 EL LENGUAJE RELATIONS

La especificación de QVT que se utiliza tiene una naturaleza híbrida declarativa/imperativa. El lenguaje Relations es una especificación declarativa de relaciones entre metamodelos MOF. Este lenguaje permite realizar “pattern matching” de objetos complejos y definir “templates” de creación de objetos.

2 LINEAS DE INVESTIGACION Y DESARROLLO

2.1 TRANSFORMACION PROPUESTA

Como se mencionó en la introducción, esta investigación tiene como objetivo particular la especificación de una transformación, mediante el lenguaje Relations de QVT, entre el metamodelo del Modelo de Negocio y el metamodelo del Modelo de Caso de Uso del Sistema. Una vez especificada la transformación, se podrá transformar, de manera automática, un modelo de negocio a su correspondiente modelo de Caso de Uso del Sistema. En la figura 1 se puede ver la secuencia completa de transformaciones, esto es, desde la especificación del Modelo de Negocio.

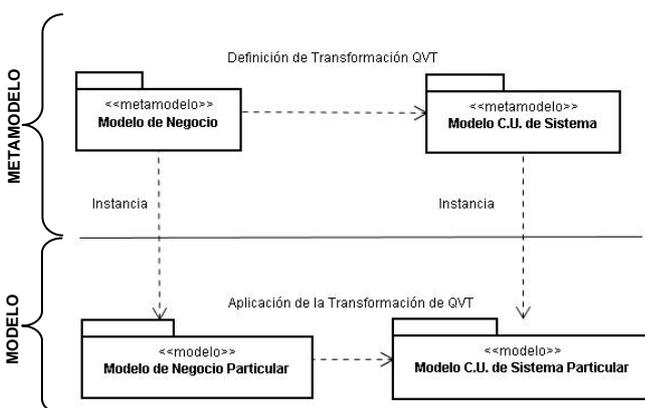


figura 1: Secuencia de transformaciones.

2.1.1 ESTADO DE AVANCE

Hasta el momento se ha avanzado principalmente en el estudio de los metamodelos especificados por UML, tanto

para el modelo de Negocio, cómo para el modelo de Caso de Usos de Sistema,

En particular la idea es poder definir el Modelo de Negocio particular en formato ecore [10], el formato ecore es el formato soportado por la herramienta MediniQVT [11] para la representación de modelos/metamodelos. MediniQVT es una herramienta que utiliza a Eclipse Modeling Framework (EMF) [12] para representar los modelos/metamodelos involucrados en las transformaciones. Cabe destacar que como parte de este trabajo debemos definir el metamodelo del Modelo de Negocio en formato ecore.

3 RESULTADOS OBTENIDOS/ESPERADOS

Esta propuesta tiene como objetivo hacer una contribución a la mejora de los procesos de desarrollo de software, viendo a éste como un proceso de negocio particular. Esta investigación pretende agregar un eslabón mas para lograr la automatización completa de la traducción de un Modelo de Negocio a su correspondiente modelo de Caso de Uso de Sistema.

Los principales objetivos de esta línea de investigación son:

- Formalizar la automatización del desarrollo de software utilizando la tecnología de flujo de trabajo con teorías como lenguajes formales de especificación y otros.
- Encarar trabajos conjuntos con universidades nacionales y con centros internacionales de excelencia como se viene realizando desde el año 2006, cuyos resultados fueron publicados en conferencias nacionales e internacionales.
- Servir como marco para dar un fuerte respaldo a la elaboración de trabajos finales de grado y tesis de posgrado.

4 FORMACION DE RECURSOS HUMANOS

Los estudios realizados en esta línea de investigación sirven como marco para la elaboración de una tesis de Maestría que está actualmente en curso. Los temas abordados en esta línea de investigación brindan un fuerte aporte al proceso de perfeccionamiento continuo de los autores de este trabajo, que se desempeñan como docentes de carreras de computación en Universidades Nacionales como del exterior.

5 BIBLIOGRAFIA

- [1] Jacobson, I. El Proceso Unificado de Desarrollo de software. Addison-Wesley, EE.UU., 2000.
- [2] OMG, Model Driven Architecture (MDA) Guide. Miller, J., Mukerji, J. (eds) 2003. document number ormsc/2001-07-01. Obt. en Mayo de 2007 de <http://www.omg.org/docs/omg/03-06-01.pdf>
- [3] UML, Unified Modeling Language (UML) Resource Page. <http://www.omg.org/#UML2.0>
- [4] Marcela Daniele, “ Un Modelo Genérico para el Modelo de Negocio y su Formalización en Object-Z”. Trabajo de Tesis para optar a la titulación de postgrado a la Maestría en Ingeniería de Software, Universidad Nacional de San Luis, 2007.
- [5] Object Management Group, “Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification” Final Adopted Specification ptc/05-11-01, <http://www.omg.org/docs/ptc/05-11-01.pdf>, último acceso Febrero 2009.
- [6] N. Debnath, F. A. Zorzan, G. Montejano and D. Riesco, “Transformation of BPMN Subprocesses Based in SPEM Using QVT”, 2007 IEEE INTERNATIONAL CONFERENCE on ELECTRO/INFORMATION TECHNOLOGY, May 17-20, 2007, Marriott O Hare, Chicago, IL, USA. <http://www.eit-conference.org/eit2007/>
- [7] Object Management Group, “ Software & Systems Process Engineering Metamodel Specification, v2”; Proposed Available Specification ptc/2007-08-07 of the Object Management Group, Inc; <http://www.omg.org/docs/formal/07-11-01.pdf> , último acceso Marzo 2008.
- [8]. Miller, J., Mukerji, J., MDA Guide Version 1.0.1 Document number omg/2003-06-01, Disponible en: <http://www.omg.com/mda>, 2003.
- [9] Object Management Group “Meta Object Facility (MOF) Core Specification” OMG Available Specification. Version 2.0. formal/06-01-01, <http://www.omg.org/docs/formal/06-01-01.pdf>, último acceso, Noviembre 2008.
- [10] Budinsky, F., Steinberg, D., Ellersick, R. y Grose, T.J., Book, Chapter 5 “Ecore Modeling Concepts”. 2004, Addison Wesley Professional
- [11] ikv++: medini QVT. <http://www.ikv.de/>, último acceso, Febrero 2009.
- [12] “Eclipse Modeling Framework”, URL: <http://www.eclipse.org/modeling/emf/>. último acceso, Marzo 2009