

Integración de modelos BPMN en ambientes MDA

Martinez Ignacio, Mendez Lautaro, Nahuel Leopoldo, Perelli Julián, Pérsico Mauricio,
Giandini Roxana

Laboratorio de Innovaciones en Sistemas de Información, LINSI.
Facultad Regional La Plata, Universidad Tecnológica Nacional.
La Plata, Buenos Aires, Argentina
{imartinez, lmendez, lnahuel, jperelli, mpersico, rgiandini}@linsi.edu.ar

Resumen. En este trabajo se presenta la utilización de modelos CIM para representar los procesos de negocio en un contexto de desarrollo MDA, y la obtención de modelos PIM, mediante transformación entre modelos. Se mostrará la adopción de dos lenguajes de modelado, BPMN y UML, para plasmar el comportamiento sistémico de la organización. El primero de ellos es utilizado con el fin de representar el funcionamiento de la organización, en forma clara y precisa a través de la creación de un CIM. El segundo de ellos permite representar modelos PIM, obtenidos a partir de los diagramas BPMN mediante el uso del lenguaje de transformaciones estándar QVT. El modelado con BPMN en etapas iniciales de un proyecto permite obtener, luego de la transformación a artefactos UML, modelos PIM con requerimientos de negocios más precisos y una mejor visión para el grupo de desarrollo.

Palabras claves: MDD, BPMN, UML, QVT, Plugin Eclipse.

1 Introducción

La ingeniería del software nace como una necesidad de consolidar maneras de crear software confiable, robusto y que cumpla con la funcionalidad requerida. En la última década se consolidó MDA (Model Driven Architecture) [1] como un nuevo paradigma para la construcción de aplicaciones software ofreciendo un nuevo enfoque metodológico y tecnológico basado en la utilización de modelos como motor productivo. Esta metodología se plantea como un proceso de construcción de software dirigido por modelos iterativo, automático y basado en estándares. Su ciclo de vida incluye distintos modelos a lo largo del proyecto de desarrollo. De esta manera se observa cómo MDA se centra en el modelado y en transformaciones automáticas a partir de los modelos más abstractos, hasta llegar al código fuente de la aplicación. En este contexto, si la generación de modelos definidos con mayor rigurosidad en las primeras instancias del proceso de construcción, los modelos siguientes, productos de la transformación, serán más específicos. Es por esto que son necesarias técnicas y tecnologías que den soporte a la construcción de modelos sin ambigüedades. Una aplicación de esto, surge de la utilización de BPMN (Business Process Management

Notation) [2], un lenguaje notacional gráfico para la modelización de procesos de negocio [3]. Al utilizar BPMN para definir un CIM, le aportamos formalidad y robustez propia de un lenguaje estandarizado, que permite generar un modelo más preciso. BPMN no fue pensado para ser usado por personas con conocimientos de informática exclusivamente, es por eso que pueden intervenir directamente en el modelado individuos que tengan que ver en la ejecución de los procesos de la organización pero que nada tengan que ver con la creación del sistema, logrando que interactúen dos tipos de actores que tradicionalmente no podían comunicarse muy bien dado los conocimientos técnicos que los separaba y de esta manera, se obtiene una participación directa de personas relacionadas con el dominio sobre el cual se está trabajando.

En el presente trabajo se expone un marco metodológico que inicialmente integre el uso de BPMN para definir la construcción de modelos CIM en un contexto MDA, con la finalidad de obtener modelos más precisos, a su vez el uso de UML [4] y transformaciones a través de QVT [5], para generar modelos PIM [6]. Para demostrar este tipo de transformaciones en MDA, se desarrolló un plugin en la plataforma EMF (Eclipse Modeling Framework) [7], que define un metamodelo BPMN y un metamodelo de UML (Unified Modeling Language), en conjunto con las transformaciones específicas de QVTO (Operational Query View Transformation) del proyecto Eclipse M2M [8] para la correcta generación de modelos CIM y PIM.

En adelante, el presente trabajo se presenta de la siguiente forma. En la sección 2, se desarrolla una serie de conceptos correspondientes al marco teórico. En la sección 3 se desarrolla la propuesta elaborada en este trabajo. En la sección 4, se explica el proceso, desarrollando un ejemplo de transformación de modelos BPMN a diagramas de actividades UML. Finalmente en la sección 5, se presenta la construcción de una herramienta CASE que brinda el soporte necesario para poner en práctica la propuesta desarrollada.

2 Marco teórico: la visión MDA y la Gestión de Procesos de Negocio

En la última década se ha dado lugar a un paradigma incipiente para el desarrollo de software dado su potencial para subsanar varios problemas inherentes a la construcción de aplicaciones.

La arquitectura dirigida por modelos, en adelante MDA (Model Driven Architecture), es una iniciativa iniciada en el año 2001 del consorcio OMG [9], destinada a la interoperabilidad entre herramientas, portabilidad y estandarización. Los pilares fundamentales de MDA se depositan en la abstracción de la arquitectura, a través de la separación de los detalles de implementación de un sistema, de la especificación del mismo. La utilización de estándares de la OMG, como es el caso de QVT, dan soporte al desarrollo dirigido por modelos a través de MDA.

Desde el punto de vista de MDA, inicialmente se crean los modelos más abstractos, llamados CIM, que describen la lógica del dominio del negocio desde una perspectiva independiente de la computación. Luego se obtiene un PIM, el cual describe de forma abstracta la funcionalidad del sistema de forma independiente a cualquier tecnología de implementación. De la construcción de estos modelos de alto nivel se procede la obtención de otros mediante transformaciones automáticas entre modelos. Un PIM se

transforma en uno o más modelos PSM, que describen el sistema en términos de un modelo con características específicas ligadas a las restricciones de una tecnología concreta. Por último, a partir del PSM se obtiene el IM que describe el sistema en término del código fuente (modelo texto) en una tecnología concreta (.Net, Java, etc.) o un lenguaje de modelado de procesos de negocio (XPDL, BPEL, etc.). En la Fig. 1, se presenta el ciclo de vida desde el punto de vista de MDA.

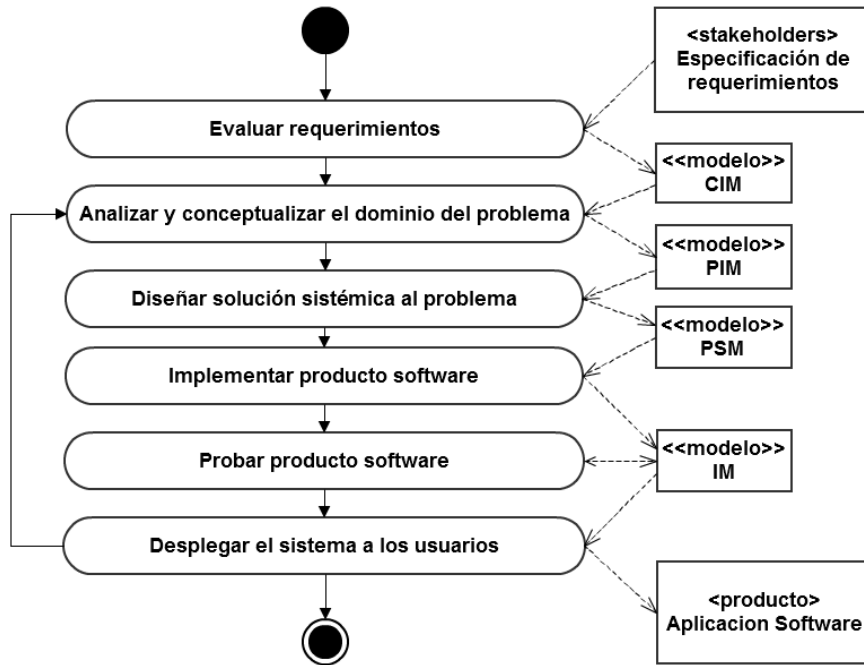


Fig. 1. Proceso iterativo para la construcción de software en un ciclo de vida MDD

A diferencia del desarrollo de software tradicional, las transformaciones de modelos en MDA, las cuales incluyen transformaciones M2M (Model To Model) y M2T (Model To Text), se desarrollan de forma automática. Existen distintas estrategias en lo que respecta al desarrollo de software basado en MDA y la transformación entre modelos. Algunas de estas involucran transformaciones horizontales (en el mismo nivel), mientras que otras transformaciones se desarrollan en sentido vertical. Dependiendo del objetivo deseado, cada una de estas transformaciones necesita una herramienta y un lenguaje particular. Por ejemplo, en el caso de transformaciones M2T, como las presentadas en [10], partiendo de modelos BPMN se llega a modelos XPDL 2.0 haciendo uso del lenguaje ATL como medio de transformación.

Algunos autores [11], [12] toman el modelado de negocios como PIM, en nuestro caso consideramos el modelado como parte del CIM. Esto nos permite definir modelos con mayor formalidad y abstracción que los diagramas que provee el lenguaje UML. Es más intuitivo y fácil de manipular por los usuarios de la organización, permitiendo abocarse a cuestiones prácticas como la especificación precisa de comportamiento y el modelado de los flujos principales y alternativos [13]. Sin embargo a la hora de realizar transformaciones de modelos de alto nivel a modelos de bajo nivel de abstracción,

UML es más apropiado, dado que permite representar diversos aspectos de un negocio en distintos diagramas, por ejemplo: diagramas de actividad para describir procesos complejos, diagramas de clase para representar entidades de dominio y sus relaciones, diagramas de máquinas de estados para reflejar los eventos y la transición de estados, entre otros.

Existen distintas estrategias para realizar transformaciones desde modelos BPMN. Estas formalizan transformaciones a un modelo en particular utilizando distintos lenguajes de transformación. En [14], [15], se presenta la transformación de procesos de negocios a diagramas de actividades y luego a modelos XPDL, tomando los procesos de negocio como modelos PIM. Otros autores, plantean distintas estrategias desde el punto de vista de MDA, como en [16], [17], donde se plantea la obtención de BPEL a través de diagramas de actividad.

3 Propuesta de integración de BPMN y UML en ambientes MDA

La utilización del modelado de los procesos de negocio a través de BPMN, permite al analista mejorar la obtención y refinación de requerimientos, con elementos conocidos por los usuarios, no sólo por los expertos. Sin embargo es necesaria la especificación de los procedimientos necesarios para que el sistema de software funcione. Previo a la construcción del sistema desde el punto de vista del código, la utilización de artefactos UML favorecen la comunicación de requerimientos hacia las áreas de programación. Es por eso en el presente trabajo se propone la utilización de BPMN en primer lugar, para favorecer la creación de dichos artefactos UML, necesarios para la posterior construcción del sistema.

Para realizar una transformación entre modelos se debe contar con información de los metamodelos que describen las representaciones de cada elemento del modelo y define las restricciones que debe cumplir un modelo. De esta manera, una transformación es una correlación entre un modelo representado por un metamodelo, a un segundo modelo, representado por un segundo metamodelo.

La solución propuesta consta inicialmente de un metamodelo BPMN para el modelado de procesos de negocios. A partir de allí se realiza una transformación a distintos artefactos de UML, como lo son los casos de uso y los diagramas de actividades, según el metamodelo de UML [18]. En la figura Fig. 2. se grafica el proceso de transformación mencionado.

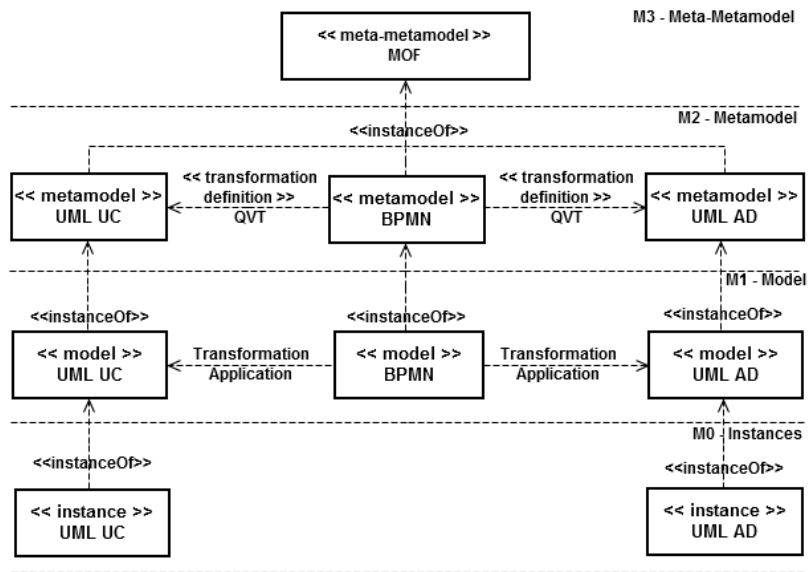


Fig. 2. Proceso de Transformación

Para definir estas transformaciones se utilizó QVT, dado que es un lenguaje estándar promovido por la OMG. QVT posee una naturaleza híbrida declarativa e imperativa. Comprende tres diferentes lenguajes M2M: 2 lenguajes declarativos llamado *relation* y *core* y tercer lenguaje de naturaleza imperativa llamado *operational mappings*. Esta naturaleza híbrida fue introducida para cubrir diferentes tipos de usuarios con diferentes necesidades, requisitos y hábitos [19].

En la transformación, se mapea un objeto del primer metamodelo a uno que sea semánticamente equivalente en el segundo metamodelo. La traducción entre los elementos de modelado de UML y BPMN se realizó basándose en las tablas de equivalencias expuestas en varios artículos [20], [21], [14].

Las reglas de transformación hacia casos de uso están basadas en trabajos anteriores [12], [15] así como también las transformaciones realizadas a través de la herramienta Agilian de Visual Paradigm. En la Fig. 3. se detallan algunas de las transformaciones utilizadas.

BPMN	Casos de uso
Pool, Lane	Actor
Task, Subprocess	Use Case

Fig. 3. Reglas de transformación de BPMN a Casos de Uso

En el caso de las reglas de transformación a diagramas de actividades, existe una mayor cantidad de recursos disponibles actualmente. En la Fig. 4. se detallan algunas de las transformaciones utilizadas, basadas en [14].

BPMN	Diagrama de actividad
Pool, Lane	Partition
Task	Activity Node
Sub Process	Structured Activity Node

Fig. 4. Reglas de transformación de BPMN a Diagramas de Actividad

4 De BPMN a diagramas de actividades UML

A continuación se muestra un ejemplo práctico de aplicación sobre un caso de estudio particular de transformación de un modelo CIM a un modelo PIM. Para esto, se utilizó una herramienta construida especialmente para realizar una transformación de un modelo BPMN a un diagrama de actividades correspondiente a UML 2.0.

En la Fig. 5 puede verse el modelo BPMN inicial, que está reglado por un metamodelo BPMN.

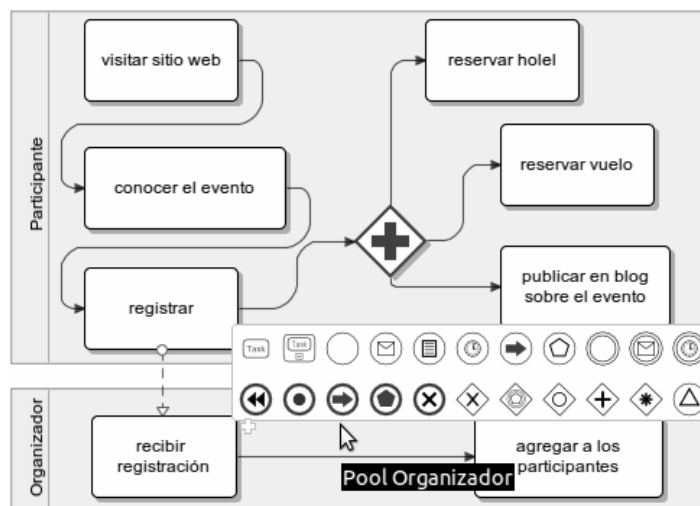


Fig. 5. Diagrama BPMN

El ejemplo describe el proceso simplificado de inscripción a un evento a través de una página web. Como paso inicial, un participante visita una página web, luego debe conocer o seleccionar el evento al cual desea asistir y por último debe registrarse. Este último evento, genera que se activen 3 procesos del participante y uno del organizador del evento. El organizador debe recibir la registración y como segundo paso agregar al participante a la lista de participantes del evento seleccionado. Por su parte, luego de haberse registrado, el participante realiza 3 acciones en paralelo: reserva un hotel, reserva un ticket de avión y crea una entrada en su blog personal.

Este flujo de acciones puede ser descrito tanto por un modelo BPMN como por un modelo de actividades UML desde distintas perspectivas, pero lo importante es que un

modelo BPMN es más accesible a todo tipo de usuarios a la hora de su análisis, manteniendo un alto grado de especificación sobre los procesos de negocio. Al realizar la transformación entre dos modelos que tienen metamodelos diferentes, no se podrá expresar exactamente lo mismo, sino que cada modelo contendrá información más específica del dominio para el que fue diseñado.

De esta forma, el modelo destino posiblemente carezca de una visión otorgada en el modelo origen. Es por esto que el modelo origen es mejor representado por varios modelos destino, para mostrar de forma completa la expresión del primer modelo.

En nuestro caso práctico, el diagrama de actividades muestra el mismo comportamiento que el diagrama BPMN, pero lo expresa de una manera diferente.

La transformación se realiza mediante el lenguaje estándar QVTO para transformación de modelos, en base a los mapeos ilustrados en la Fig. 4. En la Fig. 6 se puede ver el diagrama de actividades resultante de aplicar la transformación QVTO al modelo BPMN original.

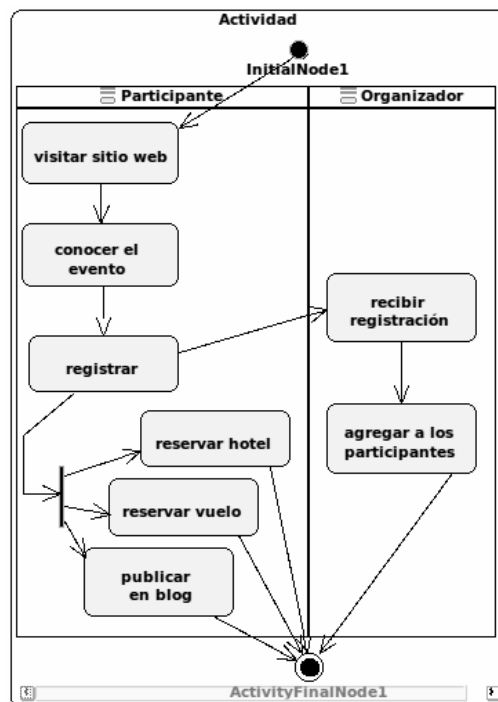


Fig. 6. Diagrama de Actividades correspondiente a UML 2.0

5 Desarrollo de herramienta CASE

Se desarroll3 una herramienta para dar soporte a la propuesta realizada anteriormente. La herramienta fue desplegada utilizando Eclipse RCP [22], [23], y se estructur3 integrando dos plugins principales para el soporte de diagramas BPMN y UML, y la incorporaci3n de las transformaciones utilizando QVTO. El plugin STP BPMN de SOA Tools Platform Project [24] provee las herramientas necesarias para la edici3n de

diagramas BPMN, mientras que el plugin gvcase-uml2 de moskitt [25]. La utilización del lenguaje QVT, a través de su implementación QVTO de parte del proyecto Eclipse M2M, brinda el lenguaje y las dependencias necesarias para la traducción entre modelos. Con respecto a UML, se utilizó el metamodelo desarrollado en MDT/UML2 [26]. El plugin de GMF [27] brinda el soporte para la edición gráfica de modelos Ecore. En la Fig. 7. se observa un esquema conceptual de las tecnologías empleadas para este proyecto.

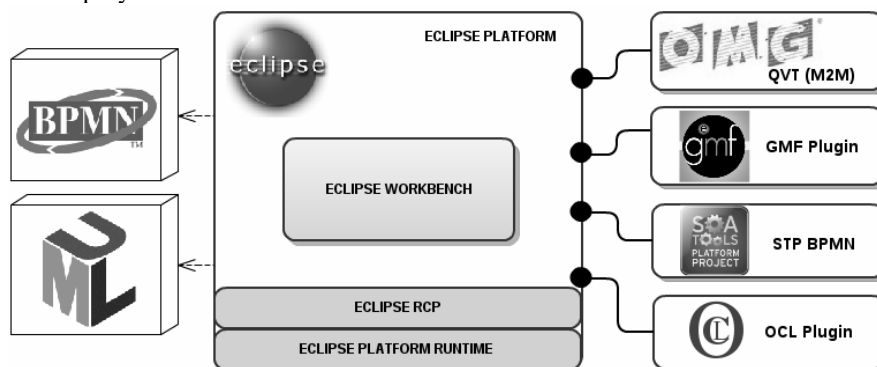


Fig. 7. Componentes del proyecto realizado

6 Conclusiones y trabajo a futuro

Este trabajo introdujo la parte inicial de un proceso de desarrollo MDA, donde el modelo más abstracto del cual se inicia, corresponde a los modelos BPMN. Esta propuesta expone una formalización concreta a la hora de incorporar modelos de procesos de negocio en un ambiente MDA.

Respecto a la utilización de software libre para la creación de herramientas que permitan poner en práctica estos conceptos, la plataforma Eclipse otorga una gran cantidad de soluciones. Sin embargo, se encontraron ciertas dificultades en la adopción de estas herramientas, debidas a la poca documentación existente y a la falta de actualización de algunos proyectos mantenidos por la comunidad.

La ventaja de utilizar una interfaz visual permite a los analistas especificar de una manera sencilla los distintos aspectos funcionales de la organización. La obtención de modelos más específicos hace posible la interacción con otras áreas de forma transparente desde etapas tempranas del proyecto.

Como trabajo futuro, uno de los objetivos es continuar desarrollando las reglas de transformación contempladas en esta aplicación. Esto no sólo posibilita ampliar en detalle y cantidad los modelos de destino logrados, sino que nos acerca al objetivo de generar código a partir de modelos.

También se pretende obtener modelos PSM, a través transformaciones a otros lenguajes y plataformas, cerrando un ciclo completo dirigido por modelos. Lograr esto, permitirá combinar la herramienta con las emergentes metodologías ágiles, para mostrar su importancia en el marco del proceso de desarrollo de software.

El objetivo es ampliar la funcionalidad del plugin para generar un producto software que exhiba la implementación del ciclo completo MDA. Teniendo esta base sólida, podrán modelarse patrones de refactoring, trazabilidad e impacto de cambios que permitan lograr un esquema de evolución de modelos iterativo.

Referencias

- [1] Warmer J., Kleppe A., Bast W.: MDA Explained: The Model Driven Architecture: Practice and Promise. Addison-Wesley Professional. ISBN-10: 032119442X,2003
- [2] Business Process Model and Notation (BPMN) 2.0. Request for Proposal, BMI/2007-06-05, Object Management Group, 2007
- [3] Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer, pp. 3-67. (2010). ISBN-10: 3642092640
- [4] Booch G., Jacobson I., Rumbaugh J.: El lenguaje unificado de modelado. Segunda Edición. Pearson Education, S.A., pp. 3-40 (2006). ISBN-13: 9788478290765.
- [5] Object Management Group: MOF Query / Views / Transformations. Version 1.0, April 2008 . http://www.omg.org/technology/documents/modeling_spec_catalog.htm.
- [6] Rodríguez, A., Fernández-Medina, E., Piattini, M.: CIM to PIM Transformation: A Reality. In Research and Practical Issues of Enterprise Information Systems II. Springer Boston. pp. 1239-1249. ISBN 978-0-387-763, 2008
- [7] Steinberg D., Budinsky F., Paternostro M., Merks E.: EMF: Eclipse Modeling Framework. Segunda Edición (2009). Addison-Wesley Professional. ISBN-10: 0-321-33188-5
- [8] Eclipse. Model To Model (M2M). <http://www.eclipse.org/m2m/>
- [9] Object Management Group (OMG). www.omg.org
- [10] Mora, B., Ruiz, F., García, F., Piattini, M.: Experiencia en Transformación de Modelos de Procesos de Negocios desde BPMN a XPDL. In: Ideas'07, X Workshop Iberoamericano de Requisitos y Ambientes de Software, Venezuela (2007).
- [11] García, I., Rodríguez, A., Fernández-Medina, E., Piattini, M.: Implementación de Heurísticas en QVT para la obtención de Clases de Análisis a partir de Modelos de Proceso de Negocio Seguros. In: IV Taller sobre Desarrollo de Software Dirigido por Modelos, MDA y Aplicaciones (DSDM'07). Zaragoza, España (2007).
- [12] Rodríguez, A., Fernández-Medina, E., Piattini, M.: Using QVT to obtain Use Cases from Secure Business Processes modeled with BPMN. In: 8° Workshop on Business Process Modeling, Development, and Support (BPMDs), pp. 319--323. Trondheim, Norway (2007).
- [13] Birkmeier D., Klöckner S., Overhage S.: An Empirical Comparison of the Usability of BPMN and UML Activity Diagrams for Business Users. In: 18th European Conference on Information Systems (2010)
- [14] Argañaraz, M., Funes, A., Dasso, A.: An MDA Approach to Business Process Model Transformations. In: SADIO Electronic Journal of Informatics and Operations Research (EJS), vol. 9, no. 1, pp. 24--48 (2010).
- [15] Rodríguez, A., Fernández-Medina, E. y Piattini, M.: Towards Obtaining Analysis-Level Class and Use Case Diagrams from Business Process Models. In: 4° International Workshop on Foundations and Practices of UML (FP-UML), Barcelona, España. Lecture Notes in Computer Science Volumen 5232, pp. 103--112 (2008).
- [16] Bordbar, B., Staikopoulos, A.: On behavioural model transformation in Web services. In: 5th International Workshop on Conceptual Modeling Approaches for e-Business eCOMO 2004, November 8-12 (2004).
- [17] Bézin, J., Hammoudi, S., Lopes, D., Jouault, F.: Applying MDA Approach to B2B Applications: A Road Map. In: Workshop on Model Driven Development (WMDD 2004) at ECOOP 2004, Springer-Verlag, LNCS, vol. 3344 (2004).

- [18] Giandini R., Pérez G., Pons C.: Un lenguaje de Transformación específico para Modelos de Proceso del Negocio. In: XXXVIª Conferencia Latinoamericana de Informática (2010).
- [19] Pons C., Giandini R., Pérez G.: Desarrollo de Software Dirigido por Modelos: conceptos teóricos y su aplicación práctica. 1er. edición. EDULP & McGraw-Hill Educación, Argentina, pp. 28-38 (2010). ISBN-13: 9789503406304.
- [20] Nguyễn Quốc Bảo: A proposal for a method to translate BPMN model into UML activity diagram. In: Vietnamese-German University - BIS 2010 (2010)
- [21] White S.: Process Modeling Notations and Workflow Patterns. In: BPTren Marzo 2004
- [22] Eclipse. Rich Client Platform http://wiki.eclipse.org/index.php/Rich_Client_Platform
- [23] McAffer, LeMieux, J., Aniszczyk, C.: Eclipse Rich Client Platform. Segunda Edición. Addison-Wesley Professional. (2010). ISBN-10: 0321612345.
- [24] Eclipse. SOA Tools Platform Project. STP BPMN (BPMN). <http://www.eclipse.org/stp/>
- [25] Generalitat Valenciana. Conselleria d'Infraestructures, Territori i Medi Ambient. Moskitt. <http://www.moskitt.org/moskitt-093f0100/>
- [26] Eclipse. Model Development Tools. Unified Modeling Language (UML) 2.x <http://wiki.eclipse.org/MDT-UML2>
- [27] Graphical Modeling Framework (GMF) Project. <http://www.eclipse.org/gmf/>