

Diseñando Transformaciones de Modelos CIM / PIM: desde un enfoque de negocio hacia un enfoque de sistema

Cecilia Ariste¹, Julieta Ponisio¹, Leopoldo Nahuel^{1,2}, Roxana Giandini^{1,2}

¹ Laboratorio de Innovaciones en Sistemas de Información, LINSI
Facultad Regional La Plata, Universidad Tecnológica Nacional
Calle 60 s/n° esq. 124, CP 1900, La Plata, Buenos Aires, Argentina
{cariste, jponisio}@linsi.edu.ar

² Laboratorio de Investigación y Formación en Informática Avanzada, LIFIA
Facultad de Informática, Universidad Nacional de La Plata
Calle 50 s/n° esq. 120, CP 1900, La Plata, Buenos Aires, Argentina
{lnahuel, giandini}@lifia.info.unlp.edu.ar

Resumen. El Desarrollo Dirigido por Modelos (MDD) es una alternativa innovadora en Ingeniería de Software, fundamentada en la construcción y transformación de modelos con distintos niveles de abstracción. En este contexto, proponemos partir de Modelos Independientes de aspectos Computacionales (CIM) con foco en modelado de Procesos de Negocio (utilizando BPMN), y alcanzar Modelos Independientes de la Plataforma tecnológica (PIM) con foco en modelos que representen workflows estructurados (utilizando UML). Para lograr esta transformación desde una perspectiva de negocios hacia una perspectiva de sistemas, presentamos un nuevo reporte de experiencias sobre diseño y propuesta de implementación de transformaciones empleando el lenguaje ATL. Específicamente, en este trabajo se presentan transformaciones de modelos de Proceso de Negocios expresados en BPMN a modelo de Diagramas de Actividades expresados en UML, y un caso de estudio para validar la especificación de la transformación. Estos modelos PIM resultan valiosos en etapas iniciales del proceso de desarrollo de sistemas.

Palabras Clave: MDD (Model Driven Development), BPMN (Business Process Model and Notation), UML (Unified Modeling Language), ATL (Atlas Transformation Language), OMG (Object Management Group).

1 Introducción

Uno de los cambios más importantes en el área de modelado y desarrollo de sistemas se evidencia con la consolidación del enfoque de Desarrollo Dirigido por Modelos (Model Driven Development, MDD) [1, 2, 3], una de las especialidades con grandes necesidades del campo de la Ingeniería de Software [4]. La idea troncal de este paradigma, es obtener mediante transformaciones automáticas, modelos más específicos o concretos, a partir de otros más abstractos.

Por otro lado, en el campo de la Gestión de Procesos de Negocio (Business Process Management, BPM) [5, 6] en las organizaciones, el lenguaje gráfico BPMN (Business Process Modeling Notation) [7, 8] es el estándar de facto en la industria para el modelado de procesos. Este lenguaje proporciona una notación fácil, legible y entendible por parte de todos los involucrados e interesados del negocio (stakeholders).

El presente trabajo persigue el objetivo de especificar transformaciones M2M (Model to Model) en etapas iniciales del ciclo de vida MDD: tomando como origen de transformación a Modelos CIM (Modelos Independientes de la Computación) construidos con lenguaje BPMN, particularmente Diagramas de Proceso de Negocio BPD, para la generación automática de Modelos PIM (Modelos Independientes de la Plataforma) [9] en lenguaje UML [10, 11], específicamente Diagramas de Actividades UML. De esta manera, se da continuidad a las líneas de investigación y resultados plasmados en trabajos publicados anteriormente en el marco del Proyecto de I&D MAPS [12, 13, 14, 15, 16] y se generan los diferentes modelos PIM, necesarios en el desarrollo de sistemas orientados a objetos.

Finalmente, se presenta una propuesta de implementación de esta transformación M2M para ser escrita en un futuro en el lenguaje ATL (Lenguaje de Transformación ATLAS) [17], que surge de un trabajo anterior en el que se logró una transformación automática hacia Diagrama de Clases y Diagrama de Casos de Uso en UML a partir de Procesos de Negocio en BPMN [18].

En adelante, el trabajo se desarrolla con la siguiente estructura y alcance: la sección 2 presenta el marco teórico. En la sección 3 se explica la especificación concreta de la transformación de CIM a PIM. En la sección 4 se analiza un caso de estudio práctico, con el objetivo de validar la especificación de la transformación. Finalmente en la sección 5 se detallan las conclusiones y se plantea el trabajo a futuro.

2 Marco Teórico: modelos, metamodelos y lenguajes en MDD

Dentro del paradigma MDD (Model Driven Development), los modelos son una parte fundamental al igual que las transformaciones automáticas que parten de modelos abstractos hasta llegar al código fuente de la aplicación.

Dentro del paradigma MDD, se crean diferentes modelos partiendo desde los que poseen niveles altos de abstracción respecto del sistema, hasta llegar a niveles más bajos [1], como se puede observar en la Figura 1.

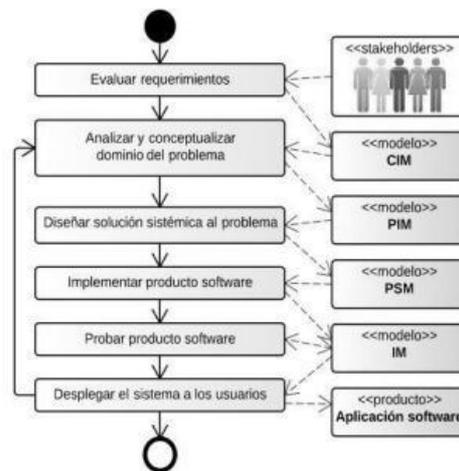


Fig. 1 Proceso iterativo para la construcción de software en un ciclo de vida MDD.

MDD utiliza cuatro tipos de modelos (de mayor a menor nivel de abstracción): CIM (Computation Independent Model), PIM (Platform Independent Model), PSM (Platform Specific Model) y CODE (código fuente de la aplicación) [1]. La idea troncal dentro de este paradigma, es obtener mediante transformaciones automáticas, modelos más específicos a partir de otros más abstractos; es decir, de un PIM obtener uno o varios PSM (según la tecnología de implementación) y de un PSM, obtener el código fuente en una tecnología específica.

2.1 Adopción del lenguaje BPMN para modelado de CIM

El lenguaje de modelado para procesos de negocio BPMN es un estándar regulado por OMG, siendo el más utilizado para modelar gráficamente los procesos de negocio. BPMN provee una notación estándar de fácil entendimiento por parte de todos los involucrados e interesados del negocio (stakeholders) que facilita la comunicación y apoya la eficiente captura de los requerimientos del proceso de negocio.

El modelado en BPMN se realiza mediante diagramas simples para modelar un proceso de negocio y obtener un Business Process Diagram (BPD), con un conjunto reducido de elementos gráficos [7, 8].

En un contexto MDD, el modelo independiente de la computación CIM, es el primer paso del proceso de desarrollo. CIM es un modelo que no tiene ningún vínculo con conceptos computacionales dado que tiene un nivel de abstracción muy alto. Teniendo en cuenta esto, un enfoque BPM proporcionará el soporte para la

construcción de un CIM a través del lenguaje notacional BPMN y que se coincide con la definición y el alcance de un modelo CIM.

2.2 Metamodelos para definir lenguajes de modelado y lenguaje ATL para Transformación M2M

Un metamodelo es un modelo que especifica los conceptos de un lenguaje, las relaciones entre ellos y las reglas estructurales que restringen los posibles elementos de los modelos válidos [3]. Meta-Object Facility (MOF) es el lenguaje para describir metamodelos que tiene como estándar OMG [19].

Una transformación de modelos es el proceso de convertir un modelo de un sistema en otro modelo del mismo sistema. En esencia, una transformación establece un conjunto de reglas que describen cómo un modelo expresado en un lenguaje origen puede ser transformado en un modelo en un lenguaje destino. 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 [3].

ATL es un lenguaje de transformación de modelos desarrollado sobre EMF (Eclipse Modeling Framework) [20]. Los desarrolladores de ATL se inspiraron en QVT para su creación, y, como en este, OCL forma parte del lenguaje. Actualmente ATL es uno de los lenguajes de transformación M2M más utilizado tanto a nivel científico como en la industria por su buena *performance*.

3. Diseño de la Transformación de modelo CIM a modelo PIM

La especificación de transformación propuesta en este trabajo utiliza los metamodelos de los lenguajes BPMN y UML para ser definida. Así describe el mapeo de transformación entre artefactos de un modelo origen (Modelo de Procesos de Negocios) hacia un modelo destino (Diagrama de Actividades UML). En la figura 2 se grafica el proceso de transformación que este trabajo especifica.

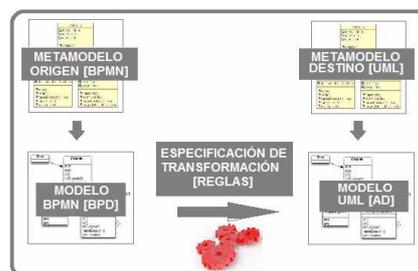


Fig. 2 Proceso de transformación M2M

3.1 Transformaciones M2M: de Procesos de Negocio en BPMN a Diagrama de Actividades en UML

Teniendo en cuenta las especificaciones y los metamodelos de los lenguajes BPMN y UML [7, 11] de OMG, se hace una especificación gráfica de transformación entre modelos. Se optó por utilizar modalidad gráfica para especificar, porque ambos lenguajes son de notación gráfica. De esta manera se define en forma clara cada una de las Reglas de Transformación que involucran un artefacto del metamodelo BPMN origen y uno o más artefactos del metamodelo UML destino. De esta forma, esta especificación comprende tanto transformaciones 1 a 1 (elemento simple BPMN a elemento simple UML) como transformaciones 1 a N (elemento simple BPMN a elemento compuesto o conjunto de elementos simples UML).

A este trabajo lo llamamos “Especificación de Transformaciones de Procesos de Negocio en BPMN a Diagrama de Actividades en UML”, dado que es un mapeo completo entre artefactos de un Modelo de Procesos de Negocio BPD en BPMN y un Modelo de Comportamiento UML, específicamente un Diagrama de Actividad UML.

El metamodelo que describe el lenguaje BPMN utilizado se toma de un trabajo de investigación publicado [21]. Por otra parte, como ya se indicó, el metamodelo de UML utilizado se toma directamente de la especificación de OMG [11], pero utilizando para este caso solo las metaclases correspondientes a Diagramas de Actividad UML.

En la Tabla 1 puede verse parte de esta especificación de transformación. El detalle completo de este trabajo puede encontrarse en un Informe Técnico [22].

Para llevar a cabo este mapeo se consideraron cada uno de los artefactos BPMN definidos en el metamodelo BPMN que forma parte de un Diagrama de Procesos de Negocio BPD. Se buscó su artefacto correspondiente en la parte del metamodelo de UML que define un Diagrama de Actividades UML.

Hay casos especiales en este mapeo en los que dos o más elementos del metamodelo origen son mapeados a un único elemento del metamodelo destino. Tal es el caso de los artefactos Pools y Lanes en el metamodelo BPMN ya que existe sólo una metaclase UML, ActivityPartition en la que derivan. Esto es propio de la naturaleza de los Diagramas de Actividad UML, en los que se trabaja solo con segmentos de responsabilidad (particiones) en contrapartida con participantes (pools) y roles (lanes) de un Diagrama de Proceso de Negocio. Otro caso especial lo constituyen las distintas especializaciones del elemento Task de BPD en BPMN (Human Task, Rule Task, Service Task, Message Task), ya que se mapean siempre al único elemento Action del Diagrama de Actividades en UML.

Tabla 1 Especificación Gráfica de la Transformación de Proceso BPD en BPMN hacia Diagrama de Actividades en UML.

Regla	Artefacto del metamodelo BPMN	Artefacto del metamodelo UML
1	Lane and Pool 	ActivityPartition
2	Subprocess 	Activity
3	DataObject 	ObjectNode
4	DataStore 	DataStore
5	ExclusiveGateway InclusiveGateway	DecisionNode (with control flows)
6	ParallelGateway Synchronizing ParallelGateway	ForkNode JoinNode
7	ComplexGateway 	JoinNode+DecisionNode
8	SequenceFlow	ControlFlow
9	Task HumanTask, RuleTask ServiceTask MessageTask	Action
10	Activity Looping 	Action + ControlFlow + DecisionNode + ControlFlow (de retorno)
11	Timer Start Event 	InitialNode + ControlFlow + Timer

12	StartEvent 	InitialNode 
13	EndEvent 	FinalNode 

Respecto a las transformaciones 1 a N, en este mapeo y en el informe técnico más completo [22] puede verse que existen muchos artefactos en el metamodelo BPMN que no pueden ser mapeados directamente a un único artefacto de un diagrama de actividad UML, sino a una composición de éstos. Tal es el caso de los distintos tipos de Eventos en BPMN: por ejemplo el elemento TimeStartEvent en BPMN se mapea a una composición de elementos InitialNode, ControlFlow y Timer. Otro caso es el del elemento ComplexGateway en BPMN que se mapea a la composición de los elementos JoinNode y DecisionNode.

3.2 Propuesta de implementación con ATL

Para automatizar la transformación CIM2PIM especificada en este paper, se realizaron pruebas en el entorno de desarrollo Eclipse Modelling Tool JUNO [23] en combinación con ATL SDK [24].

Se desarrolló un proyecto que implementa las reglas de transformación mediante su escritura en el lenguaje ATL (archivo .atl) que a su vez requiere como input los metamodelos de BPMN y UML. Esto puede visualizarse en el gráfico de la Figura 3. También se diseñaron modelos de procesos de negocio BPD en BPMN para distintos casos de estudio, para ser transformados automáticamente hacia modelos de diagramas de actividad UML.

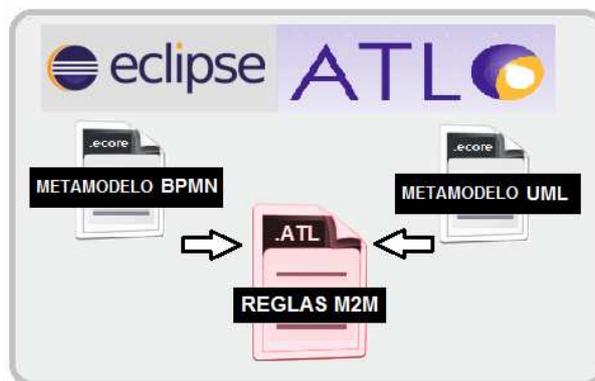


Fig. 3 Propuesta de implementación con lenguaje de transformación ATL.

4. Utilizando un Caso de Estudio para validar la transformación

El caso de estudio elegido para validar la especificación de la transformación de este trabajo se llama “Gestión de Beneficios de Empleados”, es un proceso típico que da soporte a las necesidades del Departamento de Recursos Humanos de una empresa, en el que se identifican tres roles: Empleado, Recursos Humanos y Sistema.

Este caso de estudio describe un proceso que permite a un Empleado ‘Generar Solicitud de Beneficios’ y ‘Agregar Certificados a la Solicitud’, para que el sistema de Recursos Humanos pueda ‘Evaluar la solicitud (Análisis Estratégico)’, y luego ‘Aceptar Solicitud’ o ‘Rechazar Solicitud’.

En la Figura 4 puede visualizarse el BPD (Diagrama de Procesos de Negocio) escrito en BPMN llamado “Gestión de Beneficios del Personal”.

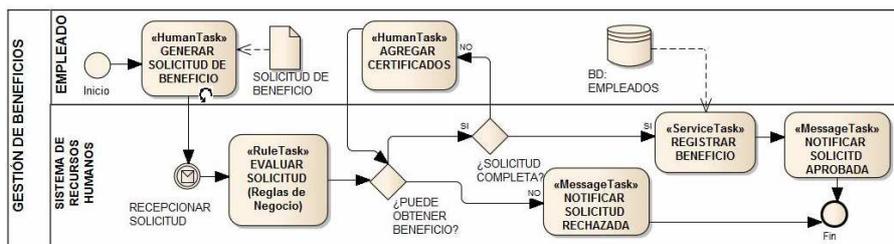


Fig. 4 Modelo BPD en BPMN “Gestión de Beneficios del Personal”.

En la Figura 5 puede verse el Modelo de Actividades UML que corresponde al BPD “Gestión de Beneficios del Personal”.

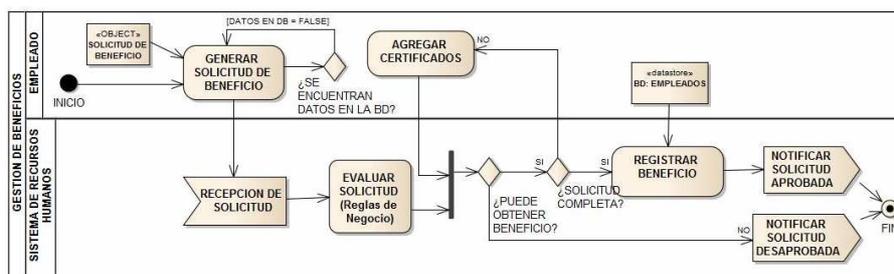


Fig. 5 Modelo Diagrama de Actividades en UML “Gestión de Beneficios del Personal”.

Puede tomarse la especificación de transformación de este trabajo para pasar del Modelo BPD-BPMN de la Figura 4 al Modelo AD-UML de la Figura 5. De este modo, el elemento Lane principal y cada uno de los Pools del modelo BPMN se mapea a Particiones del Modelo UML. Lo mismo ocurre con los distintos tipos de Task de BPMN que pasan a ser Action de UML. Una tarea repetitiva User Task en BPMN pasa a ser la composición de estos elementos UML: Action, ControlFlow, DecisionNode y otro ControlFlow de retorno al elemento Action. El resto del mapeo es trivial y muchos de los casos pueden verse en detalle en el informe técnico [22], exceptuando la correspondencia entre el elemento ComplexGateway en BPMN y la composición de los elementos JoinNode y DecisionNode de UML.

5. Conclusiones y Trabajo Futuro

El presente trabajo especifica las reglas de transformación necesarias para obtener, desde un modelo CIM (en este caso Modelos del Negocio), modelos PIM (Diagramas de Actividad UML). Esta especificación permite la independencia de la tecnología a utilizar, es decir que las reglas puedan ser desarrolladas de diversas maneras sin más ataduras tecnológicas que los propios lenguajes de modelado estándares de la OMG.

Particularmente, es destacable que este trabajo se basa fuertemente en las especificaciones técnicas de los lenguajes BPMN y UML, lo que le da criterio y lo hace independiente de tecnología, plataformas o proyectos. Además, las transformaciones aquí especificadas fueron contrastadas con trabajos de investigación y publicaciones a los que se pudo acceder y que tienen una línea de trabajo similar. Esto último le da aún más validez.

Como trabajo futuro, en lo inmediato, esta especificación de transformación nos va a permitir como equipo de investigación desarrollar un *plugin* en la plataforma EMF (Eclipse Modeling Framework) [20], que trabaje con un metamodelo BPMN y un metamodelo de UML (Unified Modeling Language), en conjunto con reglas de transformación escritas en lenguaje ATL (Lenguaje de Transformación ATLAS). Dicho *plugin* permitirá llevar a cabo una correcta transformación automática desde un proceso BPD construido en BPMN, a un Diagrama de Actividades UML, contribuyendo así a la generación de un modelo PIM necesario para la etapa de inicio del proceso de desarrollo de sistemas orientados a objetos.

Referencias

1. C. Pons, R. Giandini, G. Pérez. Desarrollo de Software Dirigido por Modelos. Conceptos teóricos y aplicación. McGraw- Hill, 1er. Edición, (2010).
2. J. García, F. O. García, V. Pelechano, A. Vallecillo, J.M. Vara, C. Vicente-Chicote. Desarrollo de Software Dirigido por Modelos. ISBN 978-84-9964-215-4 (2013).