

Lenguajes específicos y herramientas de soporte al Modelado de Procesos del Negocio en el contexto del Desarrollo Dirigido por Modelos

R. Giandini^(1,2), P. Pandolfo⁽¹⁾, A. Radic⁽¹⁾,
C. Imparato⁽¹⁾

¹ LISI – Escuela de Sistemas – Universidad Argentina John F. Kennedy
{ppando, aradic, cimparato}@kennedy.edu.ar

² LIFIA - Facultad de Informática – Universidad Nacional de La Plata
giandini@lifia.info.unlp.edu.ar

RESUMEN

En el desarrollo de software dirigido por modelos (MDD), los lenguajes de modelado juegan un rol central. Abarcan desde los más genéricos como UML hasta los llamados lenguajes específicos del dominio (DSL). En el dominio de los procesos de negocio, existen distintas notaciones y también lenguajes estándares para modelado (como BPMN) y tal como sucede en otras áreas, el diseño de un modelo BPMN necesita ser mejorado a través de técnicas de refactorización. En MDD, la refactorización de modelos es vista como una forma particular de transformaciones de modelos. Es deseable entonces contar con lenguajes de transformaciones específicos para soportar refactorización de modelos de proceso.

En este proyecto proponemos brindar asistencia al desarrollador proveyendo construcciones específicas para refactorización de modelos de proceso. Específicamente se analizarán los distintos lenguajes existentes para modelado de procesos; se logrará llegar a un manejo adecuado de herramientas específicas para modelado de procesos; se compararán las distintas herramientas existentes actualmente en el mercado y se definirán herramientas específicas y maduras para Gestión de Procesos de Negocios (BPM), que incluirán refactorizaciones básicas y avanzadas de modelos de proceso del negocio a través de la técnica de transformaciones de modelos.

Palabras clave: Ingeniería de Software, Desarrollo Dirigido por Modelos, Lenguajes de

modelado, Modelado de Procesos de Negocio, Refactorización, Herramientas CASE.

CONTEXTO

Esta línea de Investigación se enmarca dentro de las actividades en Investigación y Desarrollo que promueve el Laboratorio de Investigación en Sistemas de Información (LISI) de la Escuela de Sistemas de la Universidad Argentina John F. Kennedy (UK). En este contexto, este proyecto propone profundizar, extender y desarrollar una línea de investigación dentro de la metodología MDD: el Modelado de Procesos del Negocio. Este es uno de los tópicos de interés de su titular, la Dra. Giandini, quien viene desarrollando sus actividades de investigación en el área de MDD en los últimos años.

INTRODUCCION

El desarrollo dirigido por modelos (MDD) [1, 2, 3] ha emergido recientemente como un nuevo y prometedor paradigma en Ingeniería de Software. MDD promueve el uso de modelos no sólo para documentación y comunicación sino como artefactos de primera clase para generar otros productos durante el proceso de desarrollo, tales como otros modelos y código fuente. Los lenguajes de modelado juegan un rol central en MDD. Abarcan desde los más genéricos como UML [6] hasta los llamados lenguajes específicos del dominio (DSL), es decir lenguajes cuyos constructores representan conceptos de un dominio de problema específico.

La propuesta de Gestión de Procesos de Negocios (BPM) [9, 12, 13] ha adquirido una atención considerable recientemente tanto por las comunidades de administración de negocios como por la de ciencia de la computación. BPM provee un conjunto de metodologías para el análisis, comprensión y documentación de los procesos de negocios. En este dominio, el lenguaje de modelado estándar del OMG (Object Management Group), Business Process Modeling Notation (BPMN) [7, 11] se ha tornado popular, así como otras notaciones existentes en el mercado. Similarmente a lo que sucede en otras áreas de modelado en Ingeniería de Software (IS), el diseño de un modelo en BPMN necesita ser adaptado con el fin de mejorar su mantenimiento y calidad. El mecanismo típicamente usado para mejorar la calidad de modelos y código es la *refactorización*.

En [15], los autores adaptan refactorizaciones convencionales en la IS a las necesidades de los modelos de proceso y las completan con otras específicas para BPM. La aplicación de estas operaciones transforma un modelo de proceso P en un nuevo modelo de proceso P'. Varias de las técnicas descritas no sólo se aplican a actividades (tareas o subprocessos), sino también a artefactos (grupos) con una sola entrada y una sola salida. El término *Fragment* es utilizado para unificar estas posibilidades.

Algunas de las *refactorizaciones básicas* propuestas en [15], más comúnmente usadas son:

Rename Activity: La aplicación de *Rename Activity* genera que el nombre de una actividad x sea cambiado a y. Es comparable a *Rename Method* [16].

Substitute Fragment: Usando *Substitute Fragment* los diseñadores de procesos pueden reemplazar un fragmento por otro. Esto puede resultar útil en ciertos casos, por ejemplo porque el nuevo fragmento sea más simple o realice la misma tarea en forma diferente; se compara con la refactorización *Substitute Algorithm* [16].

Extract Fragment: La aplicación de *Extract Fragment* nos permite extraer un fragmento generando un nuevo subprocesso, con el fin de eliminar redundancias en el modelo, promover reuso o bien para reducir el tamaño del modelo. Es similar a *Extract Method* [16].

Replace Fragment: Aplicando *Replace Fragment by Reference*, un fragmento puede reemplazarse por una actividad compleja (subproceso) cuyo contenido coincide con el fragmento, haciendo referencia a ella. Esta refactorización es generalmente usada en combinación con *Extract Fragment*.

Por otro lado, es un desafío en el modelado de procesos manejar *variantes de proceso* pertenecientes a una familia de procesos. Usualmente, las variantes se derivan desde un proceso genérico aplicándole un conjunto de *operaciones de cambio*. Esto da lugar a *refactorizaciones para variantes de procesos* que permiten a los diseñadores “levantar” cambios comunes a varias variantes y “ponerlos” en el modelo genérico (en forma similar a *Pull Up Method* y *Push Down Method* en IS [16]), eliminando así redundancias y decrementando costos de futuros cambios.

Otras refactorizaciones avanzadas sobre procesos aparecen en la Evolución de Modelos. Estas técnicas se aplican cuando los procesos se ejecutan [19, 20]. En este contexto, “Eliminar bifurcaciones no usadas” (*Remove Unused Branches*) permite a los diseñadores eliminar caminos no usados de un modelo de proceso. Mientras que “Levantar cambios de instancia” (*Pull Up Instance Change*) habilita la generalización de cambios de instancia frecuentes pasándolos al nivel de proceso. Ya existen varios trabajos en minería de datos [18, 19] para descubrir estas situaciones. Por lo tanto, no nos ocuparemos de técnicas de minería, sino que se utilizarán para la realización de refactorizaciones basadas en datos históricos.

En MDD las técnicas para refactorización de modelos son vistas como una forma particular de transformaciones de modelos. Una transformación de modelos consiste de un conjunto de reglas que describen cómo un modelo escrito en un lenguaje fuente es mapeado a un modelo escrito en un lenguaje destino. Hay diversas propuestas para especificar, implementar y ejecutar transformaciones de modelos [21]. Algunos ejemplos de lenguajes de propósito general para transformación de modelos, son el estándar QVT [14] y otros inspirados en él como ATL [22] [23] y RubyTL [24]. Estos lenguajes permiten especificar diversas variantes de transformaciones de modelos.

En [17], las autoras definieron un Lenguaje específico para transformaciones de modelos de Proceso del Negocio, que incluye algunas de las refactorizaciones básicas mencionadas. El alcance de este lenguaje llamado Business Process Modeling Transformation Language (BPMTL) se limita a la definición de operaciones aplicables a modelos BPMN. La verificación de la preservación semántica de estos modelos queda fuera de su alcance. Por esta razón, es necesario integrar el lenguaje a un editor BPMN que sea capaz de verificar la preservación de invariantes del modelo previamente a la aplicación de las refactorizaciones.

Además el lenguaje debería incluir operaciones que no fueron consideradas en ese trabajo inicial y otras más complejas que fueron descritas anteriormente (refactorizaciones avanzadas), para variantes de proceso y evolución de modelos de proceso.

Concluyendo, resultan necesarios esfuerzos que conviertan estos tópicos y técnicas relacionados a la definición de lenguajes de modelado para procesos del negocio y para transformación de estos modelos, en estándares concretos que sean soportados por herramientas amigables y maduras.

LINEAS DE INVESTIGACION y DESARROLLO

- Etapa 1: Estudio de estándares para modelado de Software
 - La utilidad y características de los modelos a través del proceso de desarrollo de software.
 - El lenguaje de restricciones para objetos OCL [5]
 - Integración de restricciones OCL a diagramas UML.
 - La técnica del metamodelado. Análisis de la base formal (metamodelo) de UML
- Etapa 2: Manejo de herramientas para edición y validación de diagramas UML/OCL.
 - Estudio de diversas herramientas UML/OCL.
 - Características generales de cada herramienta, ventajas y desventajas de su uso.
 - Estudio comparativo entre las herramientas.

- Etapa 3: La filosofía del Desarrollo Dirigido por Modelos
 - El lenguaje estándar MOF (Meta Object Facility) [4]
 - Ejemplos de transformaciones en MDD y su anatomía.
 - Lenguajes de transformación de modelos. (El estándar QVT, ATL, etc.)
 - Herramientas para MDD
- Etapa 4: Modelado de Procesos del Negocio en el Desarrollo Dirigido por Modelos
 - Estado del arte sobre lenguajes de modelado de procesos de software (SPEM [8], etc.)
 - Notaciones para procesos del negocio y metodologías asociadas [10] (Diagramas de Actividad UML, BPMN, etc.).
 - Comparación de distintos lenguajes para modelado de procesos y estudio de sus características a través de su metamodelo.
 - Manejo de herramientas CASE para modelado de procesos.
- Etapa 5: Herramientas para Modelado de Procesos del Negocio en MDD
 - Desarrollo de nuevos prototipos/herramientas para modelado en BPM (Business Process Modeling)
 - Desarrollo de prototipos/herramientas de transformación para refactorización de modelos BPM.

RESULTADOS Y OBJETIVOS

- Avanzar en la capacitación continua de los miembros del grupo de investigación para fomentar la cultura I/D en el área de Sistemas de Información de la UK.
- Extender el lenguaje propuesto en [17] para que incluya diversas refactorizaciones, tanto básicas como avanzadas.
- Diseñar y construir prototipos/herramientas de transformación de modelos que permitan refactorizaciones básicas y avanzadas sobre modelos BPM.
- Integrar dicha herramienta de transformación a editores para Modelado de Procesos de Negocio ya existentes.

FORMACION DE RECURSOS HUMANOS

La formulación del Proyecto I/D en el que se enmarcan estas líneas de trabajo, surge como parte de un proceso de incentivación para el desarrollo de actividades I/D, que promueve la Escuela de Sistemas de la UK (en particular el Laboratorio LISI). Por lo tanto, se está poniendo énfasis y esfuerzo en llevar a cabo los siguientes ítems:

- Formación inicial en investigación para el grupo de trabajo (compuesto por docentes y alumnos avanzados en la carrera de Sistemas), a través del desarrollo de un seminario que abarque los conceptos básicos y avanzados en MDD.
- Generación de material y conocimiento, a partir del seminario, para el dictado de un curso optativo sobre MDD en la carrera de Sistemas de la UK.
- Formación de alumnos avanzados de la carrera a través de propuestas específicas sobre este tópico de investigación para desarrollar sus trabajos finales de carrera.
- Incorporación de Docentes participantes del grupo en Carreras de Posgrado de la Facultad de Informática, UNLP.

REFERENCIAS

1. C. Pons, R. Giandini, G. Pérez. “Desarrollo de Software Dirigido por Modelos. Conceptos teóricos y su aplicación práctica”. 1er. Edición. EDULP & McGraw-Hill Educación. (2010). ISBN: 978-950-34-0630-4.
2. Stahl, T., Völter, M. Model-Driven Software Development. John Wiley & Sons, Ltd. (2006)
3. Kleppe, A. and Warmer J., and Bast, W. MDA Explained: The Model Driven Architecture: Practice and Promise. Addison-Wesley Longman Publishing Co., Inc., USA. (2003)
4. Meta Object Facility (MOF) 2.0. OMG Adopted Specification. <http://www.omg.org>. (2005)
5. The Object Constraint Language Specification – Version 2.0, for UML 2.0, revised by the OMG, <http://www.omg.org>, April 2004.
6. The Unified Modeling Language Version 2.0., OMG Final Adopted Specification. formal/2005-07-04. <http://www.omg.org> (2005).
7. OMG Business Process Modeling Notation (BPMN) - Version 1.2 www.omg.org/spec/BPMN/1.2
8. OMG Software & Systems Process Engineering Meta-Model Specification (SPEM) - Version 2.0 <http://www.omg.org/spec/SPEM/2.0>
9. Weske Mathias, “Business Process Management: Concepts, Languages, Architectures”. Springer, Pag 3-67. ISBN 978-3-540-73521-2, 2008
10. A Notation Evaluation of BPMN and UML Activity Diagrams - Eloranta, Kallio, Terho (2006)
11. Stephen White, Derek Miers BPMN Modeling and Reference Guide: Understanding and Using BPMN; Business & Economics, 2008
12. Jeston and Nelis. Business process management: practical guidelines to successful implementations; Publisher: Butterworth-heinemann, 2008
13. Martin Owen, Jog Raj. “BPMN and Business Process Management Introduction to the New Business Process Modeling Standard”. Popkin Software. 2003
14. MOF QVT Adopted Specification 2.0. OMG Adopted Specification. November 2005. <http://www.omg.org>
15. B. Weber and M. Reichert, Refactoring Process Models in Large Process Repositories. Bellahs`ene and L´eonard (Eds.): CAiSE 2008, LNCS 5074, pp. 124–139, 2008. Springer-Verlag Berlin Heidelberg 2008
16. Fowler, M.: Refactoring - Improving the Design of Existing Code. Addison-Wesley, Reading (2000)
17. R. Giandini, G. Pérez, C. Pons. “Un lenguaje de Transformación específico para Modelos de Proceso del Negocio”. XXXVI Conferencia Latinoamericana de Informática (CLEI 2010). 18 al 22 de Octubre de 2010, Asunción, Paraguay.
18. Li, C., Reichert, M., Wombacher, A.: Issues in process variants mining. Technical Report

TR-CTIT-08-10, CTIT, University of Twente, Enschede (2008)

19. Van der Aalst, W., van Dongen, B., Herbst, J.: Workflow Mining: a Survey of Issues and Approaches. *Data and Knowledge Engineering*, 237–267 (2003)

20. Rinderle, S., Reichert, M., Jurisch, M., Kreher, U.: On Representing, Purging, and Utilizing Change Logs in Process Management Systems. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) *BPM 2006*. LNCS, vol. 4102, pp. 241–256. Springer, Heidelberg (2006)

21. Czarnecki, Helsen. Feature-based survey of model transformation approaches. *IBM System Journal*, V.45, N3, 2006.

22. ATLAS team: ATLAS MegaModel Management (AM3) Home page, <http://www.eclipse.org/gmt/am3/>. (2006)

23. Jouault, F., Kurtev, I.: Transforming Models with ATL. In: *Satellite Events at the MoDELS 2005 Conference*. Volume 3844 of *Lecture Notes in Computer Science*, Springer-Verlag (2006) 128–138

24. Sánchez Cuadrado, J., García Molina, J. and Menarguez Tortosa, M. : RubyTL: A Practical, Extensible Transformation Language. In *proceedings of ECMDA 4066*. Springer. (2006)