

Uso de Modelos de Negocios y de Requisitos en Desarrollos basados en MDA

María Carmen Leonardi
cleonard@exa.unicen.edu.ar

María Virginia Mauco
vmauco@exa.unicen.edu.ar

INTIA
Departamento de Computación y Sistemas
Facultad de Ciencias Exactas
Universidad Nacional del Centro de la Pcia. de Buenos Aires
Argentina
Fax: (54)2293-431963

1. Motivación

La Ingeniería de Requisitos ha logrado grandes avances en los últimos tiempos, contribuyendo así a mejorar la calidad del producto final [8]. Sin embargo, tiene aún un gran desafío pendiente que es traspasar las primeras etapas de requisitos y volcar todos los modelos y técnicas de requisitos hacia la siguiente etapa de desarrollo, en la cual debe definirse la arquitectura del sistema [12, 16]. La tarea de convertir los requisitos del sistema de software en una arquitectura de software viable es una tarea difícil basada principalmente en la intuición [1]. Asimismo, es aún un desafío mostrar que una arquitectura de software dada satisface un conjunto de requisitos funcionales y no funcionales.

Existe un interés creciente en el estudio de la integración entre requisitos y arquitecturas, a partir del cual se han identificado los aspectos problemáticos de esta integración así como también algunas soluciones [14, 15]. Entre estos aspectos problemáticos, destacamos la necesidad de reducir la brecha inevitable entre una especificación de requisitos, generalmente informal, y una especificación formal de arquitectura de software; la necesidad de mantener la consistencia y *traceability* [6] entre modelos de requisitos y modelos de arquitecturas; y por último, la necesidad de lograr el desarrollo de una arquitectura sobre la base de requisitos no siempre completos que incluso pueden cambiar o definirse a partir de la arquitectura [1]. Este interés por relacionar los diferentes modelos ha tenido una fuerte influencia en la comunidad de software, surgiendo recientemente el *Model-Driven Architecture* (MDA) [11]. Dentro de un desarrollo MDA, el proceso es dirigido por la actividad de modelar el sistema de software en sus diferentes fases a través de lenguajes de transformación que permiten obtener en cada etapa un modelo del anterior.

La línea de investigación aquí presentada se enmarca en este contexto, ya que pretende definir una estrategia que permita reducir la brecha entre los modelos de requisitos y los modelos tempranos de arquitecturas de software orientadas a objetos basados en UML [2], favoreciendo la definición de arquitecturas adaptables a los requisitos y capaces de evolucionar a partir de los cambios en la organización. Esto será posible gracias a la estrategia de transformación que proveerá un conjunto de heurísticas que facilitarán la *traceability* entre los modelos generados.

2. Model-Driven Architecture (MDA)

MDA es un *framework* de desarrollo de software establecido por el *Object Management Group* (OMG) [13], que propone una arquitectura basada en modelos junto con reglas de transformación entre los mismos, buscando separar la lógica de un negocio de las plataformas de desarrollo específicas que puedan implementarlo. MDA provee un enfoque y facilidades para: incorporar herramientas para especificar un sistema independientemente de la plataforma que lo soporta, especificar plataformas, elegir una plataforma para el sistema, y transformar la especificación de un sistema en una especificación para una plataforma particular.

MDA propone la construcción de distintos modelos con distintos niveles de abstracción:

- *Computer Independent Model (CIM)*: es un modelo independiente del sistema de software a ser implementado que describe la lógica del negocio en el cual dicho sistema estará inserto.
- *Platform Independent Model (PIM)*: es un modelo que describe el sistema de software que soporta una parte de la lógica del negocio, independientemente de cualquier tecnología que lo implemente.
- *Platform Specific Model (PSM)*: es un modelo que describe el sistema de software en términos de una plataforma de implementación específica.

Las reglas de transformación se aplican para obtener uno o más modelos del anterior. A partir de un CIM se pueden obtener varios PIM definiendo diferentes sistemas de software, dependiendo de qué parte del negocio se quiera automatizar. No es posible definir una derivación totalmente automatizada del CIM al PIM, debido a que la elección de qué parte del negocio será soportado con un sistema de software es siempre humana. Un PIM se puede transformar en uno o más PSMs, con los detalles específicos de cada una de las plataformas elegidas.

3. Objetivo de la línea de investigación

El objetivo de esta línea es definir una estrategia de transformación de los modelos de negocios a los modelos iniciales de la arquitectura del sistema en desarrollo, como se muestra en la Fig. 1 (adaptada de [18]).

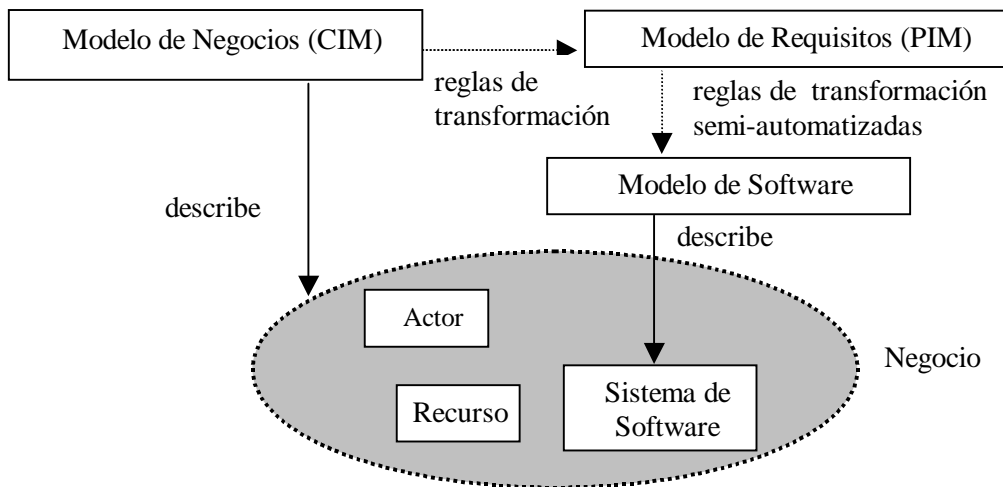


Fig. 1. Desarrollo basado en MDA

La primer etapa de la estrategia corresponde a la definición de un Modelo del Negocio que defina la organización en términos de componentes organizacionales, como por ejemplo objetivos, actores, recursos y reglas del negocio, permitiendo comprender los "por qué" que subyacen a los requisitos del sistema en lugar de "qué" debe hacer el sistema [19]. Si bien se trabajará con modelos basados en lenguaje natural, también se definirán modelos en UML complementado con el *Object Constraint Language*, conocido como OCL [17], siendo ambos lenguajes, estándares dentro de la comunidad de software. En la siguiente etapa este modelo se transformará en un Modelo de Requisitos del sistema de software también definido en UML/OCL. Finalmente, se plantearán estrategias para transformar el Modelo de Requisitos del sistema en descripciones tempranas de

arquitecturas orientadas a objetos que reflejen los requisitos funcionales y no funcionales obtenidos en las etapas anteriores.

En el contexto del MDA, el Modelo de Negocios propuesto en este trabajo es un modelo CIM. El Modelo de Requisitos que define la funcionalidad del software puede ser considerado como PIM, lo mismo que el Modelo Inicial de una arquitectura del sistema si no se toman en cuenta consideraciones técnicas. Si bien no está dentro del alcance de este trabajo, se podría estudiar alguna plataforma para definir un primer PSM de la arquitectura del sistema usando alguna plataforma específica, por ejemplo JAVA.

4. Contribuciones esperadas

A partir del objetivo planteado, se pretenden lograr los siguientes resultados:

- Definición de las componentes del Modelo de Negocios, ya sea planteando nuevas componentes organizacionales o adaptando existentes. En este punto es importante tener en cuenta el propósito de un Modelo de Negocios, ya que éste es definido con diferentes objetivos [4]. En nuestro trabajo, el objetivo del mismo es encontrar los requisitos del sistema de software y proveer un modelo base para el desarrollo del modelo de los subsecuentes modelos; por ende, sólo se describirán aquellos conceptos que sean relevantes al futuro sistema de software descartando otros aspectos de la organización. Debido a que durante esta etapa es crucial la comunicación con los *stakeholders*, se trabajará en primer lugar con modelos basados en lenguaje natural. El Modelo de Negocios debe ser posteriormente transformado en modelos UML para unificar el lenguaje a lo largo de todo el desarrollo de software, permitiendo que la transición entre el modelado conceptual y el diseño sea un proceso más natural [7] y mejorando las transformaciones entre los modelos. Como consecuencia del uso de UML/OCL para la especificación, puede surgir la necesidad de definir un UML *profile* [5] para poder modelar las componentes del Modelo de Negocios y del Modelo de Requisitos.
- Definición de la estrategia de transformación entre el Modelo de Negocios y el Modelo de Requisitos. Se deben proponer heurísticas de transformación de las componentes del negocio en los requisitos del sistema, definiendo las relaciones de *traceability* entre los modelos. El Modelo de Requisitos debe definir tanto los requisitos funcionales como los no funcionales; para estos últimos nos basaremos principalmente en [3]. Al haber unificado el lenguaje de documentación, sería posible automatizar parcialmente esta transformación entre los modelos, teniendo en cuenta que el proceso de transferir la información del Modelo de Negocios al Modelo de Software no es simple o totalmente automático [4].
- Definición de las principales componentes del Modelo Inicial de una arquitectura del sistema a ser utilizada en nuestra estrategia. Esta descripción corresponde a un modelo PIM ya que en un primer momento se pueden describir las principales componentes de una arquitectura sin tener en cuenta una plataforma específica de desarrollo. Al igual que los modelos anteriores, este modelo también será descrito en UML/OCL, por lo que puede ser necesario la definición de un nuevo UML *profile* para permitir su documentación.
- Definición de la estrategia para transformar el Modelo de Requisitos en el Modelo Inicial de una arquitectura del sistema. Es importante definir un conjunto de heurísticas de transformación de los modelos definiendo relaciones de *traceability* entre los mismos. De esta forma se podría definir una arquitectura que refleje los requisitos y sea adaptable a los cambios constantes, inevitables y necesarios de la organización.

MDA es una propuesta de desarrollo de software relativamente joven que ya ha sido ampliamente aceptada por la comunidad de software.

Teniendo en cuenta trabajos previos de los integrantes de esta línea en el área de requisitos y modelado conceptual [9, 10], creemos que se pueden realizar aportes interesantes para este nuevo enfoque de desarrollo de software.

Referencias

- [1] Bastos L., Castro J., Mylopoulos J. Integrating Organizational Requirements and Socio-Intentional Architectural Styles. Proceedings of STRAW03: Second International Workshop from Software Requirements to Architectures. 2003.
- [2] Booch G., Rumbaugh J., Jacobson I. The Unified Language User Guide. Addison Wesley. 1999
- [3] Chung L., Nixon B.A, Yu E., Mylopoulos J. Non-Functional Requirements in Software Engineering. Kluwer Academic Publishers. 2000.
- [4] Eriksson, H., Penker, M. Business Modeling With UML: Business Patterns at Work. John Wiley & Sons. 2000.
- [5] Fontoura M., Pree W., Rumpe B. The UML Profile for Framework Architectures. Addison-Wesley Pub Co. 2001.
- [6] Gotel O., Finkelstein A. An analysis of the Requirements Traceability Problem. Proceedings International Conference on Requirements Engineering, pp.94-101. 1994.
- [7] Kotonya G., Sommerville I. Requirements Engineering. J. Wiley and Sons. 1998.
- [8] Leffingwell D., Widrig D. Managing Software Requirements: A Unified Approach. Addison-Wesley. 1999
- [9] Leonardi, M. C. Enhancing RUP Business Model with Client-Oriented Requirements Models.UML and the Unified Process, ed Liliana Favre. Chapter 6 pp. 80-115. 2003. IRM Press ISBN 1-931777-44-6.
- [10] Mauco, M.V. A Technique for an Initial Specification in RSL. Master Thesis. Universidad Nacional de La Plata. Argentina. Submitted March 2004.
- [11] Model Driven Architecture. <http://www.omg.org/mda/>
- [12] Nuseibeh B. Requirements Engineering: A Roadmap. The Future of Software Engineering, A. Finkelstein (Ed), ACM Press. 2000.
- [13] OMG: Object Management Group. <http://www.omg.org/>
- [14] Proceedings of First International Workshop from Software Requirements to Architectures (STRAW01). <http://www.cin.ufpe.br/~straw01>. 2001.
- [15] Proceedings of Second International Workshop from Software Requirements to Architectures (STRAW03). <http://se.uwaterloo.ca/~straw03>.2003.
- [16] van Lamsweerde A. Requirements Engineering in the Year 00: a Research Perspective. Proceedings of the 22nd International Conference on Software Engineering, Limerick Ireland. 2000
- [17] Warmer J., Kleppe A. The Object Constraint Language: Getting Your Models Ready for MDA, Second Edition. Addison Wesley. 2003.
- [18] Warmer, J., Kleppe, A., Bast, J. MDA Explained: Practice and Promise of the Model Driven Architecture. Addison Wesley. 2003.
- [19] Yu E. Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering. Proceedings 3rd IEEE International Symposium on Requirements Engineering. Maryland. pp 226-235. 1997.