

Desarrollo de Software con enfoque en el Negocio

Andrea Delgado
Instituto de Computación
Facultad de Ingeniería
Universidad de la República
11300, Montevideo, Uruguay
adelgado@fing.edu.uy

Resumen

Las Organizaciones intentan conjuntar dos visiones para realizar su negocio: la visión del negocio centrada en especificar y mejorar sus procesos mediante análisis del negocio, y la visión de TI centrada en informatizar dichos procesos evolucionando en la tecnología y metodologías de desarrollo de software. En general esta conjunción ha sido compleja y problemática sin alcanzar una visión común del negocio por ambas partes. Sin embargo las Organizaciones son cada vez más dependientes de sus sistemas informáticos, cuentan con diversidad de sistemas que tienen entre sí dependencias complejas donde estos sistemas han ido creciendo en forma separada y heterogénea. Los avances en tecnología y los cambios en los requerimientos del negocio se retroalimentan y deben ser gestionados.

Para conjuntar estas visiones, se hace necesario cambiar la forma en que se relacionan el Negocio y su informatización, permitiendo que los procesos sean definidos y gestionados por quienes tienen ese conocimiento, y la informatización de sea realizada a partir de dichas definiciones y pueda ser cambiada según los cambios de la tecnología sin afectar esta definición, y de la misma forma, minimizar el impacto de los cambios en los procesos en la implementación de los mismos. El enfoque de diseño Service Oriented Architecture (SOA) promete cumplir este desafío conjuntando el enfoque de Business Process Modeling (BPM) con el desarrollo orientado a servicios, el enfoque de desarrollo Model Driven Architecture (MDA) propone aportes a la automatización del desarrollo. En este trabajo se plantean diversos aspectos involucrados en el desarrollo de software con enfoque en el Negocio.

1. Introducción

En los últimos años se han experimentado grandes cambios en el área de la computación, tanto en la proliferación de nuevas tecnologías, metodologías y enfoques de desarrollo que han repercutido en las Organizaciones actuales, como a la inversa, cambios en los requerimientos y necesidades a nivel Organizacional han repercutido en la forma de hacer y ejecutar software. La explosión del uso de internet por las Organizaciones, plantea varias ventajas y desafíos para la forma en que éstas realizan su Negocio, y la forma en que informatizan sus procesos e interactúan con otras Organizaciones. Lo que se hace notorio es que una necesidad que antes pudo ser medianamente satisfecha con diversidad de enfoques y tecnologías, actualmente está requiriendo respuestas más integradas, que contemplen el centro del Negocio en las Organizaciones; esta es la necesidad de enfocar el desarrollo de software en los procesos del Negocio de la Organización.

Para satisfacer esta necesidad y cerrar las brechas existentes entre el desarrollo de software y el área del Negocio en las Organizaciones, han surgido varios enfoques. En este trabajo se presentan dos enfoques para realizar el modelado del Negocio en la sección 2, así como una comparación de las notaciones planteadas, según cumplimiento de patrones para la ejecución de procesos del Negocio, luego en la sección 3 se presenta el enfoque SOA [1] para pasar del Negocio al desarrollo de software y el enfoque MDA [2] para realizar desarrollo basado en modelos, finalmente en la sección 4 se presentan algunas conclusiones y trabajo futuro a realizar en la dirección de aportes metodológicos para los enfoques planteados.

2. Enfoques para el modelado del Negocio

Los sistemas de software son cada vez más, herramientas de todos los días en el trabajo y hogares de las personas. En las Organizaciones en que son usadas estas aplicaciones deben “encajar” en el trabajo diario de las personas, dando valor a las tareas realizadas, así como permitiendo cambios asociados con la realización de las mismas. Un objetivo importante de las Organizaciones actuales es el modelado e informatización de sus procesos del Negocio, el monitoreo y la mejora de los mismos a partir de los datos de ejecución obtenidos. Se hace necesario contar con elementos y enfoques para realizar este modelado, diseño e implementación de procesos del Negocio, de forma de cubrir las expectativas de las Organizaciones. En esta sección se presentan dos enfoques para realizar el modelado del Negocio, en la sección 2.1 el enfoque de Business Process Management (BPM) [3] y en la sección 2.2 el enfoque del Rational Unified Process (RUP) [4]. En la sección 2.3 se presenta una comparación entre las notaciones definidas en cada enfoque, en cuanto al cumplimiento de patrones de workflow según [5] para ejecución de procesos del Negocio.

2.1. Enfoque del Business Process Management (BPM)

En [6] se define Business Process Management (BPM) como el conjunto de actividades que realizan las Organizaciones para optimizar o adaptar sus procesos de negocio a las nuevas necesidades organizacionales. Para [7] involucra el descubrimiento, diseño y distribución de procesos de negocio, así como el control ejecutivo, administrativo y supervisión de dichos procesos. Tiene que ver entonces con manejar el cambio para mejorar los procesos de negocio, que por años han sido gestionados con distintas técnicas y herramientas (ej. workflows), pero sin estándares definidos y ciclo de vida completo para diseñarlos y ejecutarlos. El manejo del cambio requiere control y entendimiento de los procesos, y para eso son necesarios estándares de modelado y ejecución de procesos.

Business Process Management Initiative (BMPI) [6] promueve tres estándares para el modelado,

definición y ejecución de procesos del Negocio: Business Process Modeling Notation (BPMN), para modelado de procesos, como estándar de notación para especificarlos; Business Process Modeling Language (BPML), para ejecución de procesos, como estándar de Business Process Execution Language (BPEL); y Business Process Query Language (BPQL), para distribución y ejecución de procesos, como interface de gestión estándar. Los procesos de Negocio especificados en BPMN y traducidos a BPML serán entonces ejecutados por motores de procesos en Business Process Management Systems (BPMS).

BPMN es una notación estándar para modelar visualmente flujos de procesos que tiene como objetivo proveer notación común para analistas del negocio que crean los flujos iniciales de los procesos y desarrolladores de software responsables por tecnología e implementación de los procesos. Está basado entre otros en Diagramas de Actividad de UML y Diagramas de Flujo Actividad-Decisión. Especifica un único tipo de diagrama, Business Process Diagram (BPD) con un conjunto de elementos núcleo y un conjunto de elementos completo, donde el conjunto núcleo serviría para modelar la mayoría de los procesos de negocio. Se mapea a BPML pero puede ser el front-end de modelado del negocio para sistemas diseñados con UML. Actualmente es un estándar aprobado por la OMG [8] como lo es UML. En la figura 1 se presenta un ejemplo de BPD básico que muestra algunos de los elementos de modelado utilizados en BPMN.

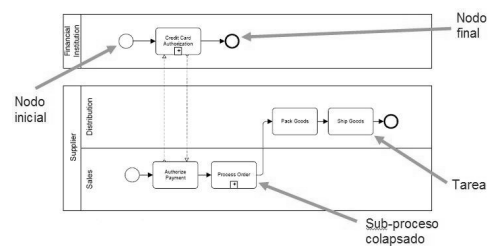


Figura 1. Ejemplo de Business Process Diagram (BPD) básico de [3]

BPML es una notación estándar para lenguajes de ejecución de procesos (BPEL) basado en XML[9], que establece un formato estándar para expresión e intercambio de procesos independiente de la implementación. BPMN mapea directamente

sobre BPML y otros como BPEL4WS [10]. El lenguaje desarrollado tiene base matemática rigurosa con el objetivo de que los sistemas construidos sobre éste puedan ser igual de resistentes que los construidos hoy por ejemplo, sobre bases de datos. Para esto se emplea semántica declarativa basada en cálculo de procesos y modelo de procesamiento concurrente.

BPML define lo que se requiere para establecer un estándar para procesos, cubriendo aspectos como actividades del negocio de complejidad variable, transacciones de negocio y sus compensaciones, manejo de datos del proceso, concurrencia, manejo de excepciones y semántica operacional. Este estándar de modelado formal de procesos, deberá ser soportado por los Business Process Management System (BPMS) para su ejecución y exposición al negocio de los procesos vía lenguajes de consulta de procesos y herramientas de modelado de procesos. Estas herramientas deberán permitir realizar el modelado de los procesos con BPMN que será traducido directamente a BPML para su ejecución. En la figura 2 se muestra como se mapea un proceso básico en BPMN a su equivalente en BPML.

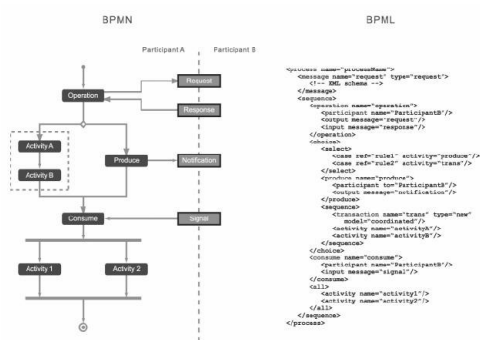


Figura 2. Ejemplo de mapeo entre BPMN y BPML básico de [7]

Otro objetivo importante planteado por BPMI para la definición de BPML fue consolidar los workflow orientados al usuario con los procesos de máquina, por lo que la comunidad de workflow en WfMC [11], expertos en el área, participó activamente en esta definición desde sus inicios.

El planteo entonces siguiendo este enfoque sería modelar los procesos del Negocio con BPMN, que

será automáticamente traducido en BPML por las herramientas utilizadas. Estos procesos especificados en BPML entonces podrán ser ejecutados por los BPMS que soportan el lenguaje definido. Los monitoreos, cambios, y acciones necesarias sobre los procesos podrán ser realizadas directamente utilizando lenguajes de interacción asociados.

2.2. Enfoque del Rational Unified Process (RUP)

Desde el punto de vista de la Ingeniería de Software, han habido varias iniciativas para realizar el modelado del negocio como parte de los proyectos de desarrollo de software. El Rational Unified Process (RUP) [4] propone una disciplina de Modelado del Negocio en la cual desarrollar actividades para obtener entregables relacionados con los procesos del negocio. Según [4] el modelado del Negocio comprende las técnicas que se pueden utilizar para modelar visualmente el negocio. Subconjunto de las técnicas que se utilizan para Ingeniería del negocio que refiere al diseño del negocio según objetivos específicos. Se define además un proceso del negocio como un grupo de actividades lógicamente relacionadas que utiliza los recursos de la Organización para proveer resultados definidos en soporte de los objetivos de la Organización, y una regla del negocio como la declaración de políticas o condición que debe ser satisfecha en el negocio, que puede ser capturada en modelos, documentos o ambos.

El RUP plantea como objetivos para la Disciplina de Modelado del Negocio comprender la estructura y dinámica de la Organización que requiere el software (Organización Objetivo), asegurar que clientes, usuarios finales, y desarrolladores tienen un entendimiento común de la Organización Objetivo, comprender problemas e identificar potenciales mejoras, y derivar los requerimientos para el sistema. Plantea también que el esfuerzo de modelado del negocio puede tener distinto alcance dependiendo del contexto y necesidades de la Organización, incluyendo reingeniería del Negocio. Como elementos para modelar los procesos del negocio propone los Casos de Uso del Negocio como descripción

textual y los Diagramas de Actividad como notación gráfica para los mismos, ambos en UML.

Como actividades principales propone evaluar estado del negocio identificando aspectos de la Organización en que se realizará el desarrollo y del negocio, e identificar los procesos del negocio, describiendo los procesos que realiza la Organización como Casos de Uso del Negocio, identificando actores y relaciones. Como principales entregables se generan la Evaluación de la Organización Objetivo y Visión del Negocio, y el Modelo de Casos de Uso del Negocio asociado a los procesos identificados. En la figura 3 se presenta el flujo de actividades definido en el RUP para la Disciplina Modelado del Negocio.

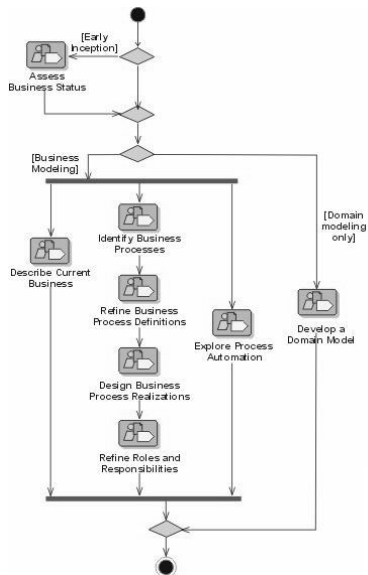


Figura 3. Flujo de actividades de la Disciplina Modelado del Negocio del RUP de [RUP]

El planteo del RUP entonces consiste en modelar los procesos del Negocio como Casos de Uso del Negocio mediante la descripción textual de los mismos, y modelar este flujo en Diagramas de Actividad como notación gráfica asociada. Ambos artefactos serán entrada luego para la Disciplina de Requerimientos, donde se definirán los Casos de Uso del Sistema asociados a los del Negocio identificados. Es posible la generación de código asociado a los flujos definidos en los Diagramas

de Actividad mediante herramientas con enfoque Model Driven Architecture (MDA) [2] como es AndroMDA [12].

2.3. Notaciones de modelado de procesos y patrones de Workflow

En [5] se realiza una comparación entre las notaciones para modelado de procesos vistas, BPMN y UML, estableciendo el cubrimiento que realiza cada una de los patrones de workflow identificados en [13] según las definiciones establecidas en [11]. Los veintiún patrones de workflow identificados en [13] describen el comportamiento de los procesos del negocio, y por lo tanto las capacidades que debe brindar un motor de workflow para la ejecución de dichos procesos.

El artículo compara los elementos de modelado existentes en BPMN y UML para cada uno de los patrones identificados, en los Business Process Diagram (BPD) y Activity Diagram (AD), estableciendo el cumplimiento de la definición de cada patrón provista en [11] y la complejidad y variaciones presentadas en cada una. A modo de ejemplo se muestra la comparación realizada para el patrón de workflow “parallel split” o partición paralela, definida en [11] como “mecanismo que permite que las actividades sean realizadas en forma concurrente en vez de secuencialmente. Un camino único en el proceso es particionado en dos o más caminos de forma que dos o más actividades puedan comenzar al mismo tiempo.”

En BPD se presentan tres formas distintas de modelar la partición paralela, la primera para flujo no controlado donde de un objeto en el flujo pueden salir dos o más objetos, la segunda utilizando un objeto especial denominado Gateway que controla el flujo subsecuente, y la tercera para flujo no controlado sin evento de inicio. En la figura 4 se muestran los mecanismos.

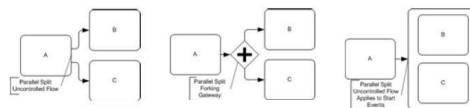


Figura 4. Mecanismos para el patrón de workflow “Parallel Split” en BPMN de [refIBM]

En AD se presenta una sola forma de modelar la partición paralela utilizando un nodo fork para crear un conjunto de caminos paralelos. En la figura 5 se presenta esta notación.

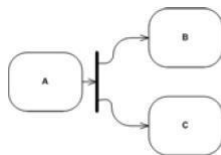


Figura 5. Mecanismos para el patrón de workflow “Parallel Split” en UML de [refIBM]

En la comparación se plantea que la notación planteada en BPMN es más simple que la de UML, ya que en el primer y tercer mecanismo provisto se realiza directamente el modelado de la partición desde el objeto origen, si se desea un objeto que controle esta partición es posible agregar el Gateway. En UML por el contrario siempre hay que agregar un nodo fork para indicar la partición en paralelo. Lo que es distinto en realidad en las dos notaciones, es el manejo del flujo de control, mientras en BPMN se realiza siempre con el diamante (notación que proviene de los diagramas de flujo) donde las marcas internas determinan si es paralelo o alternativo, en el ejemplo el + indica paralelismo, en UML se indica con una barra, el nodo fork, la partición paralela, y con un diamante las alternativas.

Luego hay varios patrones donde las notaciones son similares y varias donde son idénticas. Finalmente se concluye de las comparaciones realizadas para los veintiún patrones de workflow, que ambas notaciones pueden modelar adecuadamente la mayoría de los patrones presentados. La única excepción es para el patrón “Interleaved parallel routing” que establece que dos o más actividades en el patrón se deben realizar en realidad en forma secuencial pero sin orden establecido, ya que necesitan el mismo recurso, para el cual en UML no hay una notación gráfica adecuada aunque en el metamodelo del AD si tiene la estructura para crearlo. Que ambas notaciones compartan varios elementos, y proveen similares notaciones para la mayoría de los patrones, indica lo cerca que están una de la otra, lo que es resultado de que ambas hayan sido creadas para resolver el mismo problema: el modelado de procesos del Negocio. Las

diferencias que existen sin embargo en ambas notaciones, se plantea que son debido a la audiencia esperada de uso para cada una, mientras BPMN está orientada a analistas del negocio en su mayoría no informáticos, UML está orientada a desarrolladores de software. Se plantea también que es posible que converjan en el futuro, dado que ambas son ahora estándares de la OMG [8].

3. Del Negocio al desarrollo de Software

Se presentaron dos enfoques para modelar los procesos del negocio, desde el punto de vista del desarrollo de software: con entrada de modelado realizado por analistas del Negocio luego traducidos, con BPMN y BPML; como parte del modelado realizado directamente por el proyecto de software, con UML. Se plantea entonces la pregunta de si alcanza con realizar el modelado del negocio para que el desarrollo de software sea exitoso en cuanto al cubrimiento de las expectativas del desarrollo por parte del negocio. No parece ser suficiente, se hace necesario también un cambio de enfoque en la construcción del software. Los conceptos manejados por las aplicaciones deben ser los del negocio, además de la infraestructura de software. El enfoque de diseño orientado a servicios promete ayudar a cerrar ese gap entre el negocio y el software para soportarlo. El enfoque de desarrollo basado en modelos promete proveer herramientas que ayuden a la automatización de los modelos realizados.

3.1. El enfoque Service Oriented Architecture (SOA)

Service Oriented Architecture (SOA) según [14] es un estilo de Arquitectura de Software basado en la definición de servicios reutilizables con interfaces públicas bien definidas, donde proveedores y consumidores de servicios interactúan desacopladamente para realizar los procesos del negocio, y donde los servicios se componen en secuencias definidas para realizar los procesos de negocio (orquestración, coreografía). Como meta principal se plantea la reusabilidad e interoperabilidad de las aplicaciones obtenidas, mediante la definición de servicios que puedan ser reutilizados por la Organización y fuera de ésta. Los servicios serán

pasos, sub-procesos y procesos de la Organización, como se muestra en la figura 6.

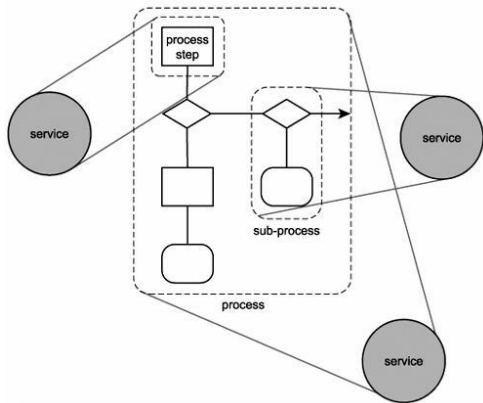


Figura 6. Servicios asociados a procesos del Negocio en desarrollo SOA de [15]

En [14] se definen cuatro abstracciones básicas para el estilo SOA: servicios, application front-end, repositorio de servicios y bus de servicios. El paradigma descubrir-ligar-invocar es la base del enfoque para el desacoplamiento de servicios, donde los productores de servicios los registran en el repositorio, los consumidores de servicios los buscan en el repositorio, y si existen obtienen una referencia para realizar el ligamiento e invocarlos, como se muestra en la figura 7.

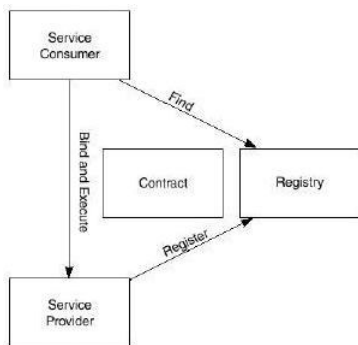


Figura 7. paradigma buscar-ligar-invocar en SOA de [16]

Según [14] un servicio consiste en una implementación que provee lógica de negocio y datos, un contrato de servicio que especifica las operaciones y las pre y post condiciones, una

interfaz que expone físicamente la funcionalidad. Los servicios representan grupos lógicos de operaciones relacionadas con algún concepto del negocio. Las application frontend consumen los servicios y/o los exponen, el repositorio de servicios almacena los contratos de servicios, y el bus de servicios interconecta las application frontend y los servicios. Además los servicios pueden clasificarse según su propósito en servicios orientados a procesos que realizan los procesos de negocio, servicios intermediarios, básicos y públicos empresariales (B2B). Aparecen dos nuevas capas de abstracción: procesos de negocio y servicios, en las que se modelan los tipos de servicios y su composición. En la figura 8 se muestran estos elementos.

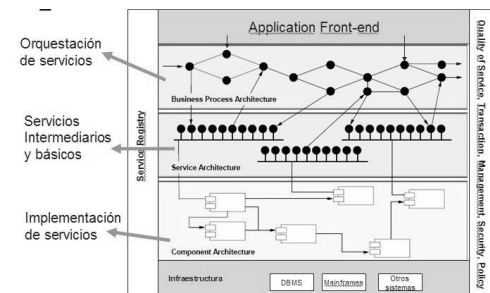


Figura 8. Elementos y capas en enfoque SOA de [17]

En [14] se plantean distintos niveles para que una Organización pueda adoptar este enfoque: Fundamental SOA donde se identifican servicios básicos que las aplicaciones pueden compartir; Networked SOA en la que se agregan los servicios intermediarios que componen varios servicios básicos y finalmente Process-enabled SOA donde los procesos del Negocio se modelan con servicios centrados en procesos, orquestando los servicios definidos. Este nivel es el más completo donde se conjuntan los enfoques SOA y BPM, de forma que la definición de los procesos (información y reglas) queda separada del código de la aplicación. En este nivel se facilita la modificación, re-configuración y optimización de los procesos en forma gráfica, minimizando el impacto en la implementación, por ejemplo mediante un BPMS, y se facilitan también los cambios tecnológicos con menor impacto en el Negocio, los procesos no cambian pero si como se implementan, sin afectarlos. Se hace necesario contar con una metodología para desarrollo de

software con enfoque SOA que permita realizar un diseño acorde a los requerimientos planteados. Una metodología para desarrollo con enfoque SOA se propone en [16].

3.2. El enfoque Model Driven Architecture (MDA)

Model Driven Architecture (MDA)[2] a diferencia del anterior, es un enfoque de desarrollo de software, que no plantea en forma explícita la realización del modelado del negocio como requerimiento para el desarrollo, ni la orientación a servicios para el diseño de las aplicaciones, pero sí los permite y promueve. Se basa en el estándar de la OMG [2] y en varios estándares ya provistos por OMG [8] como MOF, XMI para intercambio de modelos en XML, UML, OCL, QVT, que pueden verse en [8]. Plantea realizar tres vistas del desarrollo de software como modelos: Computation Independent Model (CIM) o modelo independiente de la computación en el cual especificar los requerimientos del desarrollo con artefactos como Modelo de Casos de Uso, de dominio, entre otros; Platform Independent Model (PIM) o modelo independiente de la plataforma como modelo de diseño del software en el cual incluir diagramas de subsistemas y clases, entre otros; Platform Specific Model (PSM) o modelo específico de la plataforma, donde se transforma el PIM para obtener un modelo para una plataforma en particular o el código asociado en forma directa.

Este enfoque sigue el principio básico de la Ingeniería de Software de separación de intereses, donde en cada vista se plantea la obtención de distintos intereses asociados al desarrollo. Como meta principal se plantea la portabilidad, interoperabilidad y reusabilidad de las aplicaciones obtenidas. El aspecto central del enfoque es la transformación de modelos, que según la definición provista por el estándar, es el proceso de convertir un modelo en otro modelo del mismo sistema, la cual se realiza especificando la transformación de un objeto desde un modelo origen a uno o más objetos en un modelo destino, siguiendo distintos enfoques. Para permitir estas transformaciones se proveen también los mapeos entre modelos y el marcado de modelos. Un mapeo brinda las especificaciones

para realizar las transformaciones de un PIM en un PSM para una plataforma específica, mediante mapeos o equivalencias de elementos en el modelo origen al modelo destino, en un lenguaje como QVT. Las marcas de modelos permiten marcar elementos en un modelo de forma de identificar la transformación que se desea realizar sobre el mismo, pueden ser por ejemplo estereotipos de un perfil UML.

La automatización de la transformación de un PIM hacia uno o más PSM permite crear desde una misma solución conceptual especificada en el PIM, aplicaciones que ejecutan en plataformas distintas, como J2EE o .NET, simplemente generando desde el PIM los PSM o el código asociado a cada plataforma elegida. Esto permite entonces que los modelos constituyan la base del desarrollo, donde los cambios requeridos se realicen en el PIM asociado y sean luego impactados en las plataformas correspondientes. En la figura 9 se muestra una transformación genérica en MDA, donde a partir de un PIM y otros elementos, se obtiene el PSM para alguna plataforma definida.

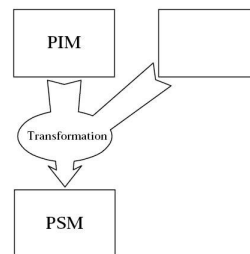


Figura 9. Transformación genérica en MDA de [2]

Es posible entonces, que la especificación de los procesos del Negocio constituyan una entrada más para estas transformaciones, en el CIM o PIM definido, y que se pueda especificar el diseño del software con orientación a servicios en el PIM, obteniendo entonces en forma automática desde los procesos del negocio especificados y los servicios definidos, el software para la o las plataformas deseadas. Es deseable contar con una metodología para el desarrollo de software con enfoque MDA que permita cumplir con los requerimientos planteados. Una metodología para desarrollo con enfoque MDA se propone en [18].

4. Conclusiones y trabajo futuro

Los enfoques presentados plantean resolver varios de los desafíos que se presentan actualmente para el desarrollo de software acorde a las necesidades que se plantean en la construcción de aplicaciones para las Organizaciones de hoy día. Se observa como principal objetivo la centralización de estas aplicaciones en los procesos del Negocio que realiza, permitiendo reaccionar ágilmente a los cambios en los mismos, a la vez que a los cambios en las tecnologías, reutilizando elementos de software tanto de diseño como de ejecución.

Los estándares de BPMN, BPML y las herramientas BPMS pueden proveer las bases para cerrar la brecha entre el modelado de los procesos del negocio y su implementación. Sin embargo, UML es el estándar “de facto” utilizado para desarrollo de Software, y como se vió, tanto los BPD de BPMN como los AD de UML pueden ser utilizados para modelar los flujos de los procesos del Negocio. El diseño del software orientado a servicios de SOA provee la infraestructura para desarrollar sistemas orientados al Negocio, con definición y especificación de procesos del negocio independiente de la implementación. La conjunción con el enfoque basado en modelos de MDA permitirá el desarrollo de servicios basado en modelos con generación de código automática.

Sin embargo, de la mano de estos enfoques también quedan planteadas varias dudas, referidas a su utilización tanto en forma aislada como en conjunto. Existen esfuerzos principalmente promovidos desde el OMG para unificar las visiones que permitan realizar desarrollos con BPM, MDA y SOA, acortando las distancias entre los requerimientos del Negocio y los del desarrollo de software. Parece esperable que los próximos años sean de cambios sustanciales en la forma de trabajo que se viene aplicando desde el área de la Ingeniería de Software.

Como trabajos futuros se plantea realizar la conjunción de las metodologías de desarrollo con enfoque SOA y MDA definidas en [16] y [18] respectivamente, investigando la conjunción con los planteos de BPM y los estándares promovidos de BPMN, BPML y BPMS para ejecución de los procesos del negocio definidos.

Referencias

- [1] Service Oriented Architecture (SOA) en el Object Management Group (OMG) <http://www.omg.org/soa> (acceso junio 2007)
- [2] Model Driven Architecture (MDA) en el Object Management Group (OMG) <http://www.omg.org/mda> (acceso junio 2007)
- [3] Business Process Management (BPM) en el Object Management Group (OMG) <http://www.omg.org/bpm> (acceso junio 2007)
- [4] IBM Rational Unified Process (RUP) en <http://www-130.ibm.com/developerworks/rational/products/rup> (acceso junio 2007)
- [5] White, S. Process modeling notation and Workflow patterns. IBM, 2004.
- [6] Business Process Management Initiative (BPMI) <http://bpmi.org> (acceso junio 2007)
- [7] Smith, H., Fingar, P. Business Process Management, the third wave. Meghan-Kieffer Press, 2003.
- [8] Object Management Group (OMG) <http://www.omg.org> (acceso junio 2007)
- [9] eXtensible Markup Language (XML) <http://www.w3.org/XML> (acceso junio 2007)
- [10] IBM et al, “Business Process Execution Language for Web Services, Version 1.1”, Mayo 2003.
- [11] Workflow Management Coalition (WfMC) <http://www.wfmc.org> (acceso junio 2007)
- [12] AndroMDA <http://www.andromda.org> (acceso junio 2007)
- [13] Van der Aalst, W., Ter Hofstede, A., Kiepuszewski, B., Barros, A. <http://tmitwww.tn.tue.nl/research/patterns/patterns.htm> (acceso junio 2007)
- [14] Krafzig, D. Banke, K. Slama, D., Enterprise SOA, Service Oriented Architecture Best Practices, Prentice Hall, 2005
- [15] Erl, T., Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall, 2005.
- [16] Delgado, A., Metodología de desarrollo para aplicaciones Service Oriented Architecture (SOA), XXXII CLEI’06, sesión 4, artículo 265, Santiago de Chile, Chile, Agosto 2006.
- [17] IBM RedBooks, Patterns: Service Oriented Architecture and Web Services, IBM, 2004.
- [18] Delgado A., Carballal N., Rapetti C., Extensión MDA para proceso basado en RUP, VI JIISIC’07, Lima, Perú, Febrero 2007.