

Usando técnicas BPM para agilizar la gestión de procesos software y mejorar la alineación con el negocio

Javier Berrocal

Dept. de Ingeniería de Sistemas
Informáticos y Telemáticos
Universidad de Extremadura
Avda. Universidad s/n
10071 Cáceres
jberolm@unex.es

José García-Alonso

Dept. de Ingeniería de Sistemas
Informáticos y Telemáticos
Universidad de Extremadura
Avda. Universidad s/n
10071 Cáceres
jgaralo@unex.es

Juan Manuel Murillo

Dept. de Ingeniería de Sistemas
Informáticos y Telemáticos
Universidad de Extremadura
Avda. Universidad s/n
10071 Cáceres
juanmamu@unex.es

Resumen

Actualmente, las compañías software utilizan los procesos de negocio con diversos propósitos. Por un lado, deben definir y gestionar sus procesos software, ya que son sus procesos de negocio. Sin embargo, la gestión de estos procesos conlleva la realización de una gran cantidad de actividades. Lo que requiere de un gran esfuerzo, e implica una disminución de la productividad de los gestores. El primer objetivo de nuestra investigación es reducir este esfuerzo. Para ello, se está adaptando un BPMS para que sea capaz de ejecutar procesos software y, así, automatizar el mayor número posible de estas actividades. Disminuyendo el esfuerzo y aumentando la productividad de los gestores. Por otro lado, algunas de estas compañías utilizan los procesos de negocio del cliente como artefacto inicial para desarrollar sus productos, ya que estos productos deben estar alineados con dichos procesos. Sin embargo, esta alineación es muy compleja de conseguir. La falta de esta alineación reduce la efectividad tanto de los sistemas IT como de los procesos. Para mitigar este efecto, el segundo de nuestros objetivos es favorecer la derivación de los casos de uso del sistema, y las relaciones entre estos, a partir de los procesos del cliente. Para ello, se han definido una serie de patrones que facilitan este paso. Mejorando la alineación entre ambos y la eficiencia final obtenida.

1. Introducción

Los procesos de negocio (PN) han ido adquiriendo cada vez un mayor protagonismo en el seno de las compañías software. Actualmente, estas compañías los están utilizando con diversos propósitos. Por un lado, utilizan herramientas e

implantan estándares que les facilitan la gestión y mejora de sus propios procesos de negocio (los procesos software) [8]. Por otro lado, analizan los procesos de negocio de sus clientes como uno de los primeros artefactos de un proyecto [12].

Nuestras líneas de investigación están encuadradas en ambos campos. La primera de ellas, se centra en reducir el esfuerzo que conlleva la gestión de los procesos software. La segunda, se centra en mejorar la alineación entre los procesos de negocio y los sistemas IT.

A partir de la primera línea de trabajo hemos detectado que, para ser competitivas, las compañías software están implantando nuevas técnicas, metodologías y modelos de negocio. Por ejemplo, están implantando estándares de calidad [8] o distribuyendo el desarrollo [13, 10]. Estas técnicas proporcionan una serie de beneficios [21], como el aumento de la calidad o la disminución de costes, pero también conllevan un incremento del número de actividades de gestión, que además suelen ser altamente repetitivas. Por ejemplo, son necesarias actividades para controlar el proceso o para gestionar a los desarrolladores y a las tareas distribuidas. La realización de estas actividades requiere de un gran esfuerzo. Consecuentemente, los gestores deben dedicar un mayor tiempo a estas actividades, disminuyendo el tiempo dedicado a otras que proporcionan un mayor valor e incrementándose los costes que se pretendían reducir.

Con el objetivo de decrementar el esfuerzo necesario para realizar tales actividades, nos centramos en tratar de automatizar el mayor número posible de ellas. Para ello se ha decidido utilizar un motor de procesos software (BPMS). Este BPMS recibe dos modelos BPMN como entrada: el proceso software a ejecutar y los procesos de negocio correspondientes al sistema para el que se desarrolla el producto software. A

partir del primer modelo, el BPMS automatiza actividades de gestión, como la apertura y cierre de iteraciones, el inicio del desarrollo de un caso de uso (con su asignación, desglose en tareas, estimaciones de esfuerzo, etc.) o la notificación de tareas. El segundo proceso permite adaptar el proceso software al producto a desarrollar. Por ejemplo, identificando cuáles son los diferentes casos de uso que hay que desarrollar. Este motor reduce el esfuerzo que hay que dedicar a las actividades de gestión.

A partir de la segunda línea de trabajo hemos detectado que, cada vez es más necesario que los productos desarrollados se alineen con los procesos de negocio del cliente [6]. Esta alineación permite que los productos proporcionen un mayor rendimiento del negocio [7] y que su tecnología sea utilizada de forma estratégica [20]. Sin embargo, todavía existen dificultades para lograr esta alineación [19]. Además de por problemas de comunicación, la raíz de esta dificultad radica en que las técnicas de especificación de requisitos detallan correctamente las funcionalidades, pero no permiten definir todas las relaciones entre requisitos o entre éstos y el entorno. La pérdida de estas relaciones favorece la aparición de defectos. Los cuales, por ejemplo, pueden influir negativamente a la integración de cada caso de uso con el resto o del nuevo sistema con el resto de sistemas del entorno. Estos defectos o deben ser corregidos o pueden provocar una disminución de la eficacia del sistema y, por tanto, del negocio.

Con el objetivo de mejorar esta alineación, estamos trabajando en la definición de una serie de patrones que permitan la identificación de los casos de uso de un sistema a partir de la descripción (en BPMN) de los procesos de negocio del cliente. El proceso de identificación pone especial atención en mantener presentes las relaciones entre casos de uso, y entre éstos y el entorno. Además, se pretende preservarlas a lo largo de todo el desarrollo. Con ello se mejora la alineación entre los sistemas IT y los procesos de negocio que soportan.

Este artículo está organizado de la siguiente forma: la sección 2 describe las motivaciones de ambas líneas de trabajo. La sección 3 indica el estado actual de ambos trabajos. La sección 4 describe los trabajos relacionados. Finalmente, la sección 5 detalla el estado actual y los trabajos futuros.

2. Motivaciones

2.1. Herramientas de gestión de procesos

La gestión de procesos software es una de las actividades más importantes para garantizar el éxito de los proyectos. Por ello, las compañías están implantando herramientas que faciliten la realización de esta actividad.

Actualmente, se están implantando herramientas de gestión cada vez más completas y que cubren un mayor número de actividades. Así, por ejemplo, Rational Team Concert [17] permite modelar el proceso software a seguir para cada proyecto, gestionar las tareas modeladas, asignar tareas a los desarrolladores o controlar el estado de cada tarea. Otro ejemplo es Specula [11], la cual permite a los gestores definir métricas alineadas con los objetivos de cada proyecto.

Sin embargo, a pesar de que estas herramientas son muy útiles para reducir el esfuerzo empleado en la gestión de procesos, sigue siendo necesario realizar todas las actividades de gestión [3]. Además, algunas de estas actividades (como la asignación de tareas o la gestión de recursos) son altamente repetitivas. Esto implica que los gestores deben dedicar un gran esfuerzo y tiempo a realizar tareas repetitivas. Por lo tanto, resultan esenciales herramientas que no solo den soporte a estas actividades, sino que también automaticen el mayor número posible de ellas. De esta forma, los gestores pueden dedicar un mayor esfuerzo a realizar actividades que proporcionen un mayor valor al cliente.

2.2. Alineación de PN y sistemas IT

Las organizaciones están constantemente evaluando y mejorando sus procesos de negocio para que proporcionen el máximo valor al negocio. Para que este valor sea el máximo posible, es necesario que los sistemas IT que los soportan estén perfectamente alineados con ellos.

Por esta razón, las compañías software especifican los requisitos de los sistemas IT a desarrollar en base a las necesidades del negocio del cliente. Sin embargo, en muchas ocasiones, una vez desarrollados los sistemas, éstos no cubren todas las necesidades o aquéllas cubiertas

no están completas [22]. Uno de los principales problemas por lo que esto viene ocasionado es porque la especificación de los requisitos no da lugar a una representación exacta del negocio y sus necesidades [20]. Técnicas como la especificación de los Casos de Uso, las Historias de Usuario o los Goals, permiten identificar y detallar los requisitos que cumplen dichas necesidades. Sin embargo, no facilitan la identificación y especificación de las relaciones entre requisitos o entre éstos y el entorno, perdiendo parte de la información de dichas necesidades y del negocio [19].

Con el objetivo de minimizar este problema, diversas investigaciones han definido métodos que permiten deducir los requisitos a partir del negocio. Así, en [18] se indica cómo obtener diagramas de casos de uso a partir de procesos de negocio. En [9] indican cómo controlar el nivel de detalle de los casos de uso. En [23] se definen algunos patrones para identificar las relaciones entre casos de uso. No obstante, estos trabajos no permiten identificar todas las posibles relaciones [4], ya que no contemplan el tratamiento de casos de uso en ramas paralelas, sub-procesos, flujos de excepción, etc.

La falta de esta información implica que los requisitos sean desarrollados sin tener en cuenta estas relaciones. Una consecuencia negativa de esto es el aumento de la dificultad de integración entre requisitos. O lo que es peor, del sistema desarrollado con el resto de sistemas y procesos del negocio. Esta falta de integración podría conllevar una disminución de la eficacia del sistema o incluso la necesidad su posterior modificación para mejorar dicha integración.

Esto pone de manifiesto la necesidad de métodos que permitan identificar las relaciones entre requisitos y entre requisitos y el entorno. Con este objetivo, se propone el uso de una serie de patrones que, aplicados sobre los procesos de negocio, facilitan la identificación de esta información. De esta forma se contribuye a mejorar la alineación de los sistemas con el negocio, aumentando la eficacia de ambos.

3. Uso de técnicas BPM para mejorar la gestión y alineación de procesos

A continuación se detallan las técnicas BPM (Business Process Management) utilizadas para

automatizar el mayor número posible de actividades de gestión del proceso software y para mejorar la alineación de los productos con el negocio del cliente.

3.1. Automatizando la gestión de procesos

Para poder automatizar las actividades de gestión es necesario saber el estado exacto de cada proceso, qué tareas se están haciendo y cuáles son las que se deben realizar a continuación.

Estas necesidades son en parte ya cubiertas por los BPMS. Los BPMS son motores que ejecutan procesos de negocio, normalmente modelados en BPMN o BPEL, para orquestar y realizar las tareas definidas en ellos.

Por lo tanto, se decidió utilizar estos BPMS para ejecutar modelos de procesos software y, así, cubrir las necesidades anteriores. Para ello, tanto la notación BPMN utilizada para modelar los procesos software, como el propio BPMS, tuvieron que ser adaptados para soportar las características específicas de la ejecución de estos procesos. Así, en primer lugar se decidió utilizar un BPMS libre, Apache ODE [1], para poder adaptarlo a éstas características sin necesidad de tener que desarrollar uno nuevo. En segundo lugar, se añadieron nuevas funcionalidades al BPMS y nuevos atributos a la especificación BPMN para incluir información específica del proceso software. Algunos de estos atributos son:

- Artefactos y modelos que deben ser generados como resultados de cada tarea.
- Herramientas o funcionalidades que deben ser utilizadas para completar cada tarea.
- Actividad automática o manual. Las actividades automáticas son aquellas que pueden ser completadas sin necesidad de intervención de los usuarios. Las actividades manuales son aquellas en las que al menos un usuario debe trabajar.

A pesar de estas adaptaciones, el funcionamiento de este BPMS es similar al de cualquier otro motor de procesos. Así, cuando el jefe de un proyecto debe decidir qué proceso software seguir, lo modela en BPMN o lo elige de entre los ya modelados. A continuación, se despliega una nueva instancia del proceso elegido en el BPMS. A partir de dicha instancia, el

proceso es monitorizado y, a medida que el proyecto avanza, las tareas definidas como manuales son añadidas a una herramienta de gestión de proyectos, para que sean estimadas y asignadas. Aquéllas definidas como automáticas, son completadas por el BPMS reutilizando información ya insertada. Algunas actividades ya automatizadas son: la apertura y cierre de iteraciones, notificación de cambios de estado o cierre de requisitos ya finalizados.

A pesar de que los proyectos desarrollados siguen el mismo proceso software (o similares), a medida que el proyecto progresa, su desarrollo ha de adaptarse a las características del producto que se desarrolla. Para obtener dicha flexibilidad, este BPMS también ha sido adaptado para que reciba una segunda entrada. Esta entrada recibe los procesos de negocio del producto a desarrollar, o los diferentes artefactos generados a lo largo del ciclo de vida del proyecto. Estos artefactos son utilizados por el BPMS para ejecutar diferentes subprocesos software dependiendo de la información recibida. Así, por ejemplo, cuando el BPMS recibe los procesos de negocio del producto a desarrollar, se ejecuta el subproceso encargado de identificar los casos de uso, y las relaciones entre casos de uso, a partir de dichos procesos. Además, este subproceso gestiona las tareas a realizar para identificar dicha información.

Un beneficio adicional de ejecutar, con un BPMS, el proceso software como un proceso de negocio es el hecho de que, para cada proyecto, la compañía de desarrollo puede poner en marcha diferentes procesos software dependiendo de requisitos tales como la agilidad requerida, el nivel de control de calidad o la tecnología con la que se va a desarrollar. Para ello, simplemente es necesario contar con la descripción BPMN relativa a cada proceso.

3.2. Alineando los sistemas IT con el negocio

Normalmente, los procesos software definen actividades encaminadas a tener en consideración el negocio de las organizaciones, tales como definir los objetivos del negocio o los casos de uso de negocio [12]. Sin embargo, estas actividades están orientadas a exponer el negocio como punto de partida, y no a conseguir una alineación PN-IT. Ello conlleva que en numerosas ocasiones, el

sistema IT desarrollado no se alinee correctamente con el negocio para el que fue desarrollado.

En esta línea de trabajo se propone modelar los procesos de negocio de una organización (utilizando la notación BPMN) como punto inicial para identificar los requisitos del sistema a desarrollar. El objetivo no es solo obtener los requisitos sino mantener presentes las relaciones entre ellos. Para facilitar esta identificación, se han definido una serie de patrones que permiten identificar tanto los casos de uso como las relaciones entre casos de uso (CUs) a partir de los procesos de negocio.

Para definir estos patrones se ha adoptado el concepto de “Step” introducido en [9] (donde se indica que un “step” es la secuencia de tareas que pueden ser realizadas sin interrupción por un actor y, por lo tanto, un “step” es un caso de uso).

Tabla 1. Identificación de casos de uso a partir de procesos de negocio. Ampliado en [4]

Paso	Descripción
1	Se comienza en el primer evento o actividad del proceso de negocio
2	Todas las actividades conectadas mediante un flujo de control son un CU.
3	Un caso de uso acaba cuando:
	A. El flujo pasa de un actor a otro
	B. Existe una transición de tiempo entre actividades
	C. Se llega al final del proceso

La aplicación de estos patrones se basa en la realización de los siguientes pasos:

- Identificar los actores del sistema. Un actor será: todo agente externo al negocio (pools del proceso), los agentes internos del negocio (pools y lanes) o sistemas heredados.
- Marcar las actividades de los procesos que deben estar soportadas por el sistema, marcando cada actividad (como se indica en [9]): como A (Automática), S (Soportada) o M (Manual)
- Identificar los CUs a partir de los procesos de negocio. Para lograrlo, se deben detectar los conjuntos de tareas que pueden ser realizadas por un actor sin interrupciones, siguiendo los pasos indicados en la tabla 1 (basados en el concepto de “Step” introducido en [9]).

- A partir de este punto, se deben identificar las relaciones entre CUs mediante los patrones definidos. La Tabla 2 muestra algunos de estos patrones. Una extensión de los mismos puede encontrarse en [4].

Tabla 2. Patrones para la identificación de relaciones entre casos de uso

Descripción	Diagrama
Si existen varios CUs unidos por un Gateway de tipo “exclusivo” o “inclusivo”, el CU en la ramificación por defecto se relacionará con el CU inicial mediante “Include” (puesto que éste es el CU por defecto a seguir), y el CU en la rama alternativa mediante “Extends”, puesto que éste sólo se realizará cuando se cumpla su condición.	
<i>Tratamiento de excepciones.</i> Si una excepción cubre varios casos de uso, se crea un nuevo CU para el flujo de excepción. Éste estará relacionado mediante “Extend” con los CUs cubiertos por el flujo excepción. De esta forma el control y tratamiento de la excepción estará encapsulado en un CU.	
<i>Tratamiento de subprocesos.</i> Para cada subproceso se crea un caso de uso de alto nivel que englobe y controle a todos los casos de uso identificados dentro del subproceso, estando relacionado con ellos mediante la relación “Include”. Mediante este CU se mantiene las relaciones existentes entre el resto de actividades del proceso y el subproceso. Los CUs existentes dentro del subproceso estarán relacionados entre sí según los patrones definidos.	

A través de estos pasos y patrones se facilita la identificación de los requisitos del sistema a desarrollar y de las relaciones entre requisitos, o de éstos con otros elementos del entorno (como sistemas heredados). Las relaciones se mantienen presentes durante todo el proceso de desarrollo para conseguir mejorar la alineación e integración entre el negocio y los sistemas IT.

4. Trabajos relacionados

4.1. Herramientas de gestión de procesos

A medida que las compañías software han implantado nuevos estándares o modelos de negocio, han visto como sus procesos se han sobrecargado con actividades de gestión. Esto ha conllevado a que cada vez sean más necesarias herramientas que den soporte a estas actividades.

Orientadas en este sentido se han empezado a desarrollar herramientas que proporcionan un mayor control de los procesos, llamadas software cockpit [16]. Ejemplos de estas herramientas pueden ser Rational Team Concert [17] o Artemis 7 [2]. Estas herramientas dan soporte a la gestión de diferentes procesos software y entornos de desarrollo. Además, facilitan la realización de un gran conjunto de actividades, como el control de asignaciones o la generación de métricas. Sin embargo, sólo dan soporte a la realización de estas actividades, no las automatizan.

Por otro lado, se están realizando investigaciones orientadas a mejorar este tipo de herramientas. Un ejemplo de estas investigaciones es Specula [11]. Esta investigación se centra en la definición de una metodología y de una herramienta capaz de definir componentes reusables para la gestión de procesos. Así, cada compañía puede definir sus componentes y reutilizarlos según sus objetivos y los objetivos de cada proyecto. Sin embargo, al igual que ocurría con las herramientas anteriores, estos componentes sólo dan soporte a la realización de actividades de gestión, no las automatizan.

A diferencia de los trabajos mencionados, una de las líneas de trabajo presentadas en este artículo se centra en el desarrollo de un motor de procesos capaz de automatizar un gran número de actividades de gestión. No existe en estos momentos ningún motor con dichas características.

Finalmente, actualmente existen trabajos orientados a facilitar la ejecución de procesos de negocio, como [5, 24], así como para aumentar la variabilidad y adaptabilidad de los mismos [14, 15]. Estos trabajos están orientados a mejorar estas características para cualquier proceso de negocio. Nosotros nos centramos únicamente en poder ejecutar los procesos software y soportar

cierta adaptabilidad de los mismos con respecto a la situación de los proyectos software.

4.2. Alineación sistemas IT - PN

A medida que las empresas han puesto un mayor énfasis en la necesidad de sistemas IT alineados con su negocio, el número de investigaciones y perspectivas con las que atacar este objetivo se ha incrementado.

Así, en [9] indican una serie de mapeos para obtener actores, casos de uso o, incluso, algunas relaciones entre casos de uso a partir de procesos de negocio modelados con Diagramas de Actividad de UML. Sin embargo, sólo indican cómo obtener relaciones de tipo "extend". No permiten capturar todas las relaciones existentes (como las relaciones de tipo "include", casos de uso en sub-procesos, flujos de excepción, etc.).

En [23], extienden los mapeos indicados en [9] añadiendo tres patrones para la identificación de las relaciones entre casos de uso. No obstante, siguen sin contemplar todas las posibles opciones para la identificación de las relaciones (como pueden ser, casos de uso en ramas paralelas, sub-procesos, flujos de excepción, etc.).

Por último, en [18] extienden los diagramas BPMN y los Diagramas de Actividad de UML para que puedan representar procesos de negocios con conceptos de seguridad. Además, presentan una serie de transformaciones en QVT y reglas para obtener los casos de uso, y los casos de uso que proporciona la seguridad, a partir de los procesos de negocio. Sin embargo, no se indica cómo obtener las relaciones entre casos de uso.

A diferencia de los trabajos mencionados, la segunda línea de investigación presentada en este artículo define una serie de patrones que no sólo facilitan la identificación de casos de uso, sino que además permiten identificar las relaciones entre éstos o de éstos con el entorno, a partir de procesos de negocio.

5. Estado actual y trabajos futuros

En este artículo se han presentado dos líneas de trabajo que utilizan técnicas BPM para mejorar tanto la gestión de los procesos de las compañías software como la eficacia de los productos que éstas desarrollan.

La primera línea de investigación está enfocada hacia la agilización de la gestión de los procesos software. Con este fin, se ha adaptado un BPMS para que sea capaz de ejecutar procesos software. Esta ejecución permite automatizar un gran número de actividades, tales como el control de las iteraciones, el control de cambios de estado o el control de los casos de uso. De esta forma, se reduce el esfuerzo dedicado a la gestión de procesos, permitiendo a los gestores centrarse en actividades que proporcionan más valor al cliente. Como trabajo futuro, estamos desarrollando nuevas funcionalidades que incrementan el número de actividades automatizadas, así como la capacidad de adaptación de los procesos software a las características del producto que se desarrolla.

La segunda línea de trabajo se orienta hacia la mejora de la alineación de los sistemas IT con los procesos de negocio del cliente. Como resultado de este trabajo, se han definido una serie de patrones que permiten identificar los casos de uso de un sistema, junto con las relaciones entre éstos, a partir de sus procesos de negocio. Por ejemplo, se han definido patrones para identificar relaciones de tipo extends e include o para el tratamiento de subprocesos. Estos patrones mejoran la alineación IT- Procesos de negocio, mejorando el rendimiento e integración de dichos sistemas con el negocio. Como trabajo futuro, estamos definiendo nuevos patrones que permitan identificar un mayor conocimiento del negocio. Además, se pretende extender un lenguaje de definición de requisitos para que dicha información pueda ser modelada y mantenida a lo largo de todo el desarrollo.

Finalmente, se pretende utilizar la información extraída del negocio para facilitar el desarrollo de sistemas IT. En concreto, de los sistemas desarrollados con una arquitectura multicapa y utilizando frameworks de desarrollo. La construcción de este tipo de aplicaciones requiere de un gran conocimiento. Por ello, se están definiendo reglas que, reutilizando dicha información, faciliten su implementación.

Agradecimientos

Este trabajo ha sido financiado por PDT08A034, TIN2008-02985, Fondos FEDER, PRE09156 y la Fundación Valhondo Calaff.

Referencias

- [1] Apache ODE. <http://ode.apache.org/>
- [2] Artemis 7. <http://us.aisc.com>
- [3] Berrocal, J., García-A., J., Murillo, J.M.: Lean Management of Software Processes and Factories using Business Process Modeling Techniques. PROFES 2010, pp 321-335
- [4] Berrocal, J., García-A., J., Murillo, J.M.: Patrones para la Extracción de Casos de Uso a partir de Procesos de Negocio. PNIS 2009, pp 1-11 (2009)
- [5] Cánovas, J.L., Sánchez, O., García, J.J., Castillo, C.; Un caso de estudio para la adopción de un BPMS. Taller de Procesos de Negocio e Ingeniería Software (2007)
- [6] Chan Y, Huff S, Barclay D, Copeland D (1997) Business strategic orientation, information systems orientation and strategic alignment. Information System Research Vol. 8, No. 2, pp. 125-150
- [7] Cragg P, King M, Hussin H (2002) IT alignment and firm performance in small manufacturing firms. Journal of strategic information systems. Vol. 11, pp. 109-132.
- [8] CMMI. <http://www.sei.cmu.edu/cmmi/>
- [9] Dijkman, R.M and Joosten, Stef (2002) Deriving Use Case Diagrams from Business Process Models. CTIT Technical Reports Series, 08 (02). ISSN 1381-3625
- [10] Greenfield, J., Short, K., Cook, S. and Kent, S.: Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools. (2004)
- [11] Heidrich, J, Münch, J.: Goal-Oriented Setup and Usage of Custom-Tailored Software Cockpits. Product-focused software process improvement (2008)
- [12] Jacobson, I., Booch, G., Rumbaugh, J. (1999) The Unified Software Development Process. Addison-Wesley Object Technology Series. ISBN:0-201-57169-2
- [13] Johnson, James R.: The Software Factory: Managing Software Development and Maintenance. John Wiley & Sons. ISBN-10: 047157225X (1991)
- [14] La Rosa, M., Gottschalk, F., Dumas, M., van der Aalst, W.: Linking Domain Models and Process Models for Reference Model Configuration. 10th Workshop on Reference Modeling, Brisbane, Australia (2007)
- [15] Montero, I., Peña, J., Ruiz-Cortés, A.: Business Family Engineering: Does it make sense? Primer taller de Procesos de Negocio e Ingeniería Software (2007)
- [16] Münch, J., Heidrich, J.: Software Project Control Centers: Concepts and Approaches. Journal of Systems and Software 70(1), 3-19 (2003)
- [17] Rational Team Concert. <http://www-01.ibm.com/software/awdtools/rtc/>
- [18] Rodríguez, A.; Fernández-Medina, E; Piattini, M. (2008): Towards Obtaining Analysis-Level Class and Use Case Diagrams from Business Process Models. ER Workshops 2008
- [19] Salinesi C, Thevenet L (2007) Aligning IS to organization's strategy: the INSTAL method. International Conference on Advanced Information Systems Engineering, Norway
- [20] Sase Narine Singh Carson Woo, (2009). Investigating business-IT alignment through multi-disciplinary goal concepts. Requirements Engineering Journal Vol. 14, No. 3 pp. 177-207
- [21] Sengupta, B., Chandra, S. and Sinha, V.: A research agenda for distributed software development. (ICSE 06) Proceeding of the International Conference on Software Engineering, Shanghai, China, May 20-28 of 2006, pp 731-740. (2006)
- [22] Standish-Group (1994) The Chaos report. In: The Standish Group
- [23] Svatopluk Stolfa, Ivo Vondrák (2005) Mapping from Business Processes to Requirements Specification.
- [24] Vanderfeesten, I.T.P., Reijers, H.A., van der Aalst, W. Product Based Workflow Support: A Recommendation Service for Dynamic Workflow Execution. 20th International Conference on Advanced Information Systems Engineering (CAiSE'08), volume 5074 of Lecture Notes in Computer Science, pages 571-574. Springer-Verlag, Berlin, 2008.