

Transformación de Actividades SPEM por Medio de su Transformación en Relations a subProcesos BPMN

Fabio Zorzan

Departamento de Computación
Universidad Nacional de Río Cuarto,
Ruta 36 Km 601, 5800, Río Cuarto, Córdoba, Argentina
fzorzan@dc.exa.unrc.edu.ar

y

Daniel Riesco

Departamento de Informática
Universidad Nacional de San Luís,
Ejercito de los Andes 950, 5700, San Luís, Argentina
driesco@unsl.edu.ar

Resumen

El modelado de los procesos de negocio es de vital importancia en el desarrollo de toda industria, en particular, en la industria del software. Una forma de optimizar la producción es mediante la automatización de los procesos de negocio. Este trabajo aporta a la automatización total o parcial de la gestión de las actividades de los procesos de desarrollo de software, especificados con el Software Process Engineering Metamodel(SPEM), mediante workflows. Para lograr esta automatización, se transforman las actividades del proceso de desarrollo de software especificadas en SPEM en una especificación de subprocesos de la Business Process Modeling Notation (BPMN) definido por la Object Management Group (OMG). Esta transformación se realiza por medio del lenguaje Relations que forma parte de Query/Views/Transformations (QVT). La definición de subprocesos BPMN obtenida, puede ser transformada en una especificación en un lenguaje estándar para la implementación de procesos workflow, como ser Business Process Execution Language for Web Services (BPEL4WS) o XML Process Definition Language (XPDL). Así, de esta manera, la especificación en el lenguaje de implementación elegido será la entrada a cualquier motor Workflow. De esta forma es posible automatizar total o parcialmente la gestión de actividades de procesos de desarrollo de software especificados bajo SPEM.

Palabras claves: Workflow, SPEM, BPMN, QVT, Relations.

1 INTRODUCCIÓN

Los procesos de negocio son un conjunto de tareas lógicamente relacionadas, ejecutadas para obtener un resultado de negocio.

Los procesos de negocio pueden ser controlados y administrados por un sistema basado en software. Los procesos de negocio automatizados de esta manera se denominan workflow. Esta automatización resulta en una importante potenciación de las virtudes de dicho proceso. Se obtienen mejoras en cuanto a rendimiento, eficiencia y productividad de la organización.

El paradigma workflow ofrece interoperabilidad con otros sistemas, ejecución en ambientes distribuidos, facilidades para el monitoreo y manejo de recursos humanos[1].

El caso de la industria del desarrollo de software, no es diferente al del resto de las industrias. Dentro de ella, se encuentran los procesos de negocios tendientes a la construcción o generación de un producto (software) de calidad en un tiempo determinado[2]. El proceso de negocio mas importante dentro de la industria de desarrollo de software es conocido como “metodologías de desarrollo”, encargadas de guiar la producción. Actualmente, los ingenieros de software trabajan para optimizar los procesos de desarrollo. Los desarrolladores de las herramientas de ingeniería de software pueden explotar la conexión entre la administración de proceso de desarrollo de software y workflow[3].

Este trabajo aporta a la optimización del proceso de producción de software mediante la automatización total o parcial de la gestión de actividades de las metodologías de desarrollo de software especificados en SPEM. Para lograr esta automatización, se propone una traducción de las actividades del proceso de desarrollo de software especificado en SPEM[4] a una especificación de sub-procesos Workflow basado en el estándar BPMN[5] aceptado por la OMG. Esta traducción se obtiene a través de una transformación definida mediante el lenguaje Relations que forma parte de QVT[6]. La transformación está definida entre el Metamodelo SPEM y el metamodelo BPMN. La hipótesis de trabajo es plantear al proceso de desarrollo de software como un tipo proceso de negocio particular automatizándolos en todo o en parte a través de un motor de workflow.

Es importante a la hora de dar claridad a los procesos de negocio, la descomposición de estos en subprocesos de alto nivel. El objetivo es transformar las actividades del proceso de desarrollo de software en sub-procesos de un workflow para poder lograr la automatización total o parcial de su gestión.

El trabajo está organizado de la siguiente manera. En la sección 2 se presenta al SPEM con sus características. La sección 3 presenta la tecnología workflows y la notación BPMN. En la sección 4 se presenta el lenguaje de transformación Relations, parte de QVT. La sección 5 describe la transformación del metamodelo SPEM al metamodelo BPMN. Por ultimo, en la sección 6 están las conclusiones.

2 SPEM

Los procesos en el desarrollo de software pueden ser vistos como productos, ya que están constantemente cambiando y evolucionando. También deben ser administrados y configurados para adaptarlos a las organizaciones y a las nuevas necesidades del entorno, agregando de esta forma la necesidad de un estándar unificado en esta área, esto debido a que cada una de estas técnicas y procesos definió sus propios estándares y terminologías usando incluso diferentes significados para la misma palabra.

Para especificar las actividades propuestas por un proceso de desarrollo particular y de esta forma proveer una solución a la necesidad antes planteada, la OMG definió un metamodelo para la Ingeniería de Procesos de Software (SPEM).

Para la definición de nuevos Lenguajes la OMG define una arquitectura basada en cuatro niveles de abstracción que van a permitir distinguir entre los distintos niveles conceptuales que intervienen en el modelado de un sistema. Esos niveles se denominan M0, M1, M2 y M3 y se describen a continuación:

- **El nivel M0 – Las instancias.** El nivel M0 modela el sistema real, y sus elementos son las *instancias* que componen dicho sistema. Un elemento de este nivel es por ejemplo el cliente llamado Pablo Gonzalez.
- **El nivel M1 – El modelo del sistema.** Los elementos del nivel M1 son los *modelos* de los sistemas concretos. Existe una relación muy estrecha entre los niveles M0 y M1, los conceptos del nivel M1 definen las *clasificaciones* de los elementos del nivel M0. En este nivel está definido el concepto cliente.
- **El nivel M2 – El modelo del modelo (el metamodelo).** Los elementos del nivel M2 son los lenguajes de modelado. El nivel M2 define los elementos que intervienen a la hora de definir un modelo del nivel M1. Aquí también existe una gran relación entre los conceptos de los niveles M1 y M2 donde los elementos del nivel superior definen las *clases* de elementos válidos en un determinado modelo de nivel M1. Conceptos de este nivel son Clase, Atributo, etc.
- **El nivel M3 – El modelo de M2 (el meta-metamodelo).** Finalmente, el nivel M3 define los elementos que constituyen los distintos lenguajes de modelado. La OMG ha definido un lenguaje para describir los elementos del M3 llamado Meta-Object Facility (MOF) [7], un elemento de este nivel en el clasificador, el concepto de clase definido en M2 es un clasificador.

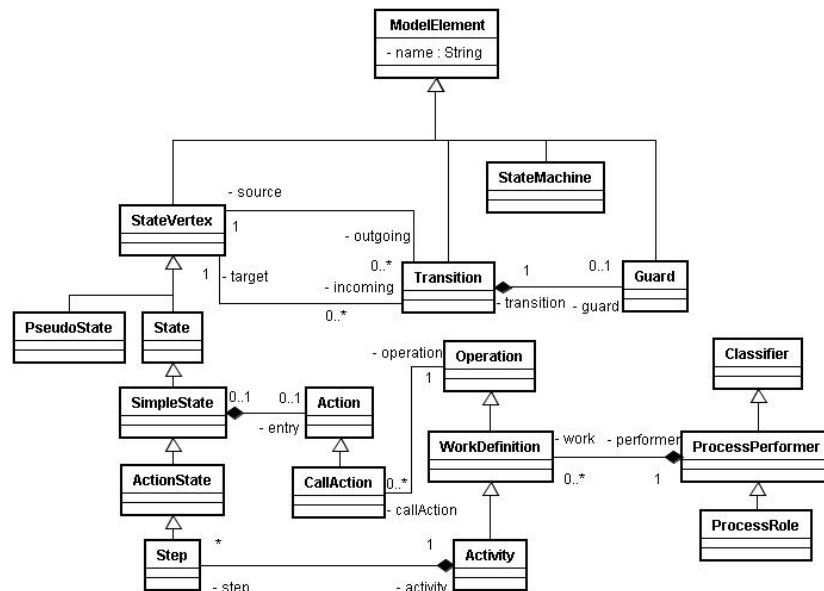


Fig. 1. Vista del metamodelo SPEM con las metaclasses involucradas en este trabajo.

SPEM describe un metamodelo genérico para la descripción de procesos software concretos que está basado en MOF y utiliza UML como notación de modelado. Por tanto, se basa en los principios de orientación a objetos.

El metamodelo SPEM sirve como plantilla para la creación de modelos de procesos concretos, como podrían ser el “Proceso Unificado de Desarrollo de software de Rational” (RUP) o el modelo de evaluación y mejora de procesos de ISO 15504. Por tanto, SPEM es un metamodelo del nivel M2 de MOF, mientras que estos procesos citados se definirían en base a SPEM en el nivel M1.

Este trabajo se centra en el paquete estructura de procesos de SPEM que define los elementos estructurales principales en la construcción de la descripción de un proceso y el paquete maquina de estados que define principalmente las transiciones entre las “actividades” del proceso. La Figura 1 muestra una vista del metamodelo SPEM con las metaclases involucradas en la transformación.

3 WORKFLOW

Un workflow se define como la automatización total o parcial de un proceso de negocio, durante la cual documentos, información o tareas son intercambiadas entre los participantes conforme a un conjunto de reglas procedimentales preestablecidas [8].

Un workflow comprende un número de pasos lógicos, conocidos como actividades. Una actividad puede involucrar la interacción manual o automática con el usuario.

Un motor workflow es un sistema de software que controla la ejecución de las actividades definidas en el workflow. La WfMC ha definido un Modelo de Referencia Workflow (Workflow Reference Model). Este modelo define 5 interfaces para la interoperabilidad de diferentes productos con un motor workflow.

En este trabajo interesa la interfaz 1 que especifica el formato de intercambio común para soportar la transferencia de definiciones de procesos entre productos diferentes, utilizando un lenguaje de definición de procesos como el XML Process Definition Language – (XPDL)[9] definido por la WfMC o el Business Process Execution Language for Web Services(BPEL4WS)[10] adoptado por OASIS. XPDL permite escribir especificaciones de procesos workflow de manera estandarizada. Esto significa que cualquier definición de proceso que cumpla con todos los requisitos establecidos en la interfaz 1 podrá ser tomada como entrada por cualquier motor workflow que respete el estándar establecido por la WfMC, por ejemplo OFBiz Workflow Engine [11] o Open Business Engine [12].

BPEL4WS es un lenguaje para la especificación de procesos de negocio, el cual permite especificar procesos de negocio basados en servicios Web, esto es, que sólo pueden importar y exportar funcionalidad mediante servicios Web. La especificación inicial (BPEL4WS 1.0) fue desarrollada por IBM, Microsoft y BEA . WebSphere Process Server de IBM[13] y BPEL Process Manager[14] de Oracle son ejemplos de motores de workflow que implementan BPEL4WS.

Es importante a la hora de modelar un proceso de negocio poder utilizar una herramienta independiente de la implementación, así, de esta manera, poder utilizar la especificación del proceso de negocio para diferentes plataformas. Una herramienta de estas características que esta siendo muy utilizada por grandes empresas es BPMN.

3.1 BPMN

La OMG junto con la Business Process Modeling Initiative(BPMI) han desarrollado la notación BPMN para el modelado de procesos de negocio. BPMN define una notación para la definición de procesos de negocio, lo que es una plataforma independiente con respecto a definiciones específicas de procesos de negocio, por ejemplo XML Process Definition Language (XPDL)[9] o Business Process Execution Language for Web Services (BPEL4WS)[10]. Esta notación define una representación abstracta para la especificación de procesos ejecutables de negocio que se ejecutan dentro de una empresa (con o sin intervención humana); y puede colaborar con otro proceso de

objetos. El trace de los elementos de los modelos involucrados en las transformaciones son creados explícitamente.

Estas transformaciones describen relaciones entre un meta-modelo fuente F y un meta-modelo objetivo O, ambos metamodelos deben estar especificados en MOF. Luego esta transformación definida se utiliza para obtener un modelo objetivo que es una instancia del metamodelo O a partir de un modelo fuente que es una instancia del metamodelo F. Una característica muy importante de estas transformaciones es que pueden ser bidireccionales (multidimensionales también).

Una transformación especifica un conjunto de relaciones que deben cumplir los elementos de los modelos involucrados. Una relación especifica una relación entre elementos de los modelos candidatos y consiste de dos o mas dominios, y dos restricciones denominadas cláusula **guard** (o cláusula **when**) y cláusula **where**. Cada dominio define a un modelo de candidato. Cada dominio tiene patrones. Un patrón puede ser considerado un template para los objetos y sus propiedades, que deben ser encontrados, modificados o creados en los modelos involucrados para satisfacer la relación entre los modelos.

5 TRANSFORMACIÓN DEL METAMODELO SPEM AL METAMODELO BPMN

En este punto se presenta la transformación del metamodelo SPEM al metamodelo BPMN, primero dando un contexto general de las componentes intervinientes en la transformación y luego se muestra, en lenguaje Relaciones de QVT, la definición de la transformación.

5.1 Contexto General de la Transformación

La vista general de la transformación de procesos de desarrollo de software basados en SPEM a workflows puede ser particionada en tres niveles: Metamodelo, Definición/Modelo y Ejecución, como lo muestra la figura 3.

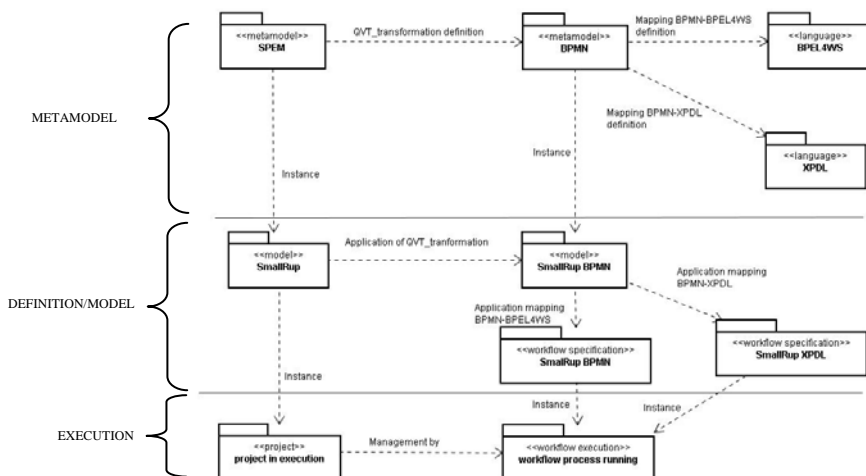


Fig. 3. Vista general de la transformación del metamodelo SPEM al metamodelo BPMN.

A nivel metamodelo se encuentran los metamodelos objetivos de la transformación definida en este trabajo, el metamodelo SPEM y el metamodelo BPMN, entre los cuales esta definida la transformación mediante el lenguaje QVT. A su vez en este nivel se encuentran las definiciones de los mapping entre el metamodelo BPMN y los diferentes lenguajes de implementación de workflow, en este caso los lenguajes XPDL y BPEL4WS.

Pasando al nivel de modelo/definición se encuentran los modelos específicos que definen un proceso de desarrollo de software concreto, como por ejemplo SmallRUP[16], y a partir de éste, por aplicación de la transformación definida en QVT a nivel metamodelo, se obtiene el modelo BPMN que define a SmallRUP como un proceso de negocio. También en este nivel se encuentran la aplicación de los mapping entre el metamodelo BPMN y los diferentes lenguajes de definición de procesos, que como resultado de la aplicación de estos mapping se obtiene la definición de SmallRUP en un lenguaje (XPDL o BPEL4WS). Ésta definición se utiliza como entrada para la definición de procesos en un motor Workflow que implemente el lenguaje.

Por último, en el nivel de ejecución, se encuentran los proyectos de desarrollo de software que siguen como metodología de desarrollo de software a SmallRUP y que son administrados automáticamente a través de motores de workflow que siguen como especificación de procesos de negocio a la definida en el nivel anterior.

5.2 Transformación del Metamodelo SPEM al Metamodelo BPMN

Este trabajo hace una contribución a la mejora en la gestión de las actividades de los procesos de desarrollo de software que están basados en el estándar SPEM. Para esto se propone la transformación de las actividades de los procesos software basado en SPEM a subprocessos BPMN que pueden ser automatizados, previa transformación a XPDL o BPEL4WS, mediante la utilización de un workflow estándar. De esta forma se puede utilizar una herramienta workflow que implemente el estándar de la WfMC (XPDL) o el estándar de OASIS (BPEL4WS) para asistir en la gestión de los procesos de desarrollo de software.

Para poder especificar las reglas de transformación se definió una correspondencia entre metaclasses del metamodelo SPEM y metaclasses del Metamodelo BPMN.

Para la definición de las reglas de transformación de metamodelos se adoptó el lenguaje Relations de QVT. QVT permite hacer transformación de diferentes formas (unidireccionales, bidireccionales, solamente chequeo de correspondencias entre modelos, entre otras). En este caso la transformación solo necesita ser definida en la dirección Metamodelo SPEM hacia Metamodelo BPMN. De esta manera los elementos del metamodelo SPEM están marcados como *checkonly* y los elementos del metamodelo BPMN están marcados como *enforced*, para que de esta forma la ejecución de la transformación cree los elementos del modelo BPMN que se corresponden a los elementos del modelo fuente especificado en SPEM.

5.2.1 Definición de la Transformación de Actividades SPEM a SubProcesos BPMN

En esta sección se presenta relación definida para la transformación de actividades SPEM a SubProcesos BPMN. La transformación entre el metamodelo SPEM y el Metamodelo BPMN se define de la siguiente manera:

Transformation SpemWorkflow(spem: Spem, bpmn: BPMN)

Esta transformación toma un modelo *spem* que es una instancia del metamodelo SPEM y un modelo *bpmn* que es una instancia del metamodelo BPMN.

Para una mejor comprensión de la explicación de la definición de la relación presentada que forman parte de la transformación QVT, los nombres de las metaclasses serán escritos con fuente de letra cursiva.

Debajo se muestra la especificación de la relación **activitySPEMToSubprocessBPMN** que define la correspondencia entre la metaclass *State*, (en particular para los objetos de la metaclass *State* que tienen definida la relación *entry* con un objeto de la metaclass *CallAction* y a su vez, el objeto *CallAction* debe estar relacionada un objeto de la metaclass *Activity* del metamodelo SPEM) y la

metaclase *EmbeddedSubProcess* del metamodelo BPMN. Las transiciones (metaclase *Transition* de SPEM) de entrada y salida del *ActionState*, representadas por los atributos *outgoing* e *incoming* (atributos heredados de la metaclases *StateVertex*) se transforman en conectores de secuencia (*SequenceConnector*) de entrada y salida respectivamente de la metaclase *EmbeddedSubProcess* del metamodelo BPMN. Esta transformación está definida por dos template de creación de objetos, uno para los conectores de entrada y otro para los de salida. Dentro de los templates de los conectores se define el atributo *conditionExpression* que representa la condición bajo la cual se puede ejecutar la transición definida por el conector. La expresión del conector se obtiene a través de la guarda de la transición de SPEM que se corresponde al conector. La correspondencia entre la guarda de SPEM y la expresión de BPMN esta definida en la cláusula **where** de la relación. La cláusula **where** obliga a cumplir la relación *guardToExpression*, entre la guarda de las transiciones y la expresión de los conectores. También dentro de la cláusula **where** de obliga al cumplimiento de la relación *stepSPEMToTaskBPMN* que define la correspondencia entre un *Step* de SPEM y una *Task* BPMN.

top relation activitySPEMToSubprocessBPMN

```
{
  nameActivity : String;
  nameTransitionIn : String;
  nameTransitionOut : String;
  nameRole : String;

ckeckonly domain spem actionState: ActionState
  {
    entry = callAction : CallAction
    {
      operation = activity : Activity
      {
        name = nameActivity,
        peformer = performe : ProcessRole{name = nameRole},
        assistente = assistente : ProcessRole {}
      }
    }
    incoming = transitionInSpem : Transition
    {
      guard = guardTransitionIn : Guard{},
      name = nameTransitionIn
    },
    outgoing = transicionOutSpem : Transition
    {
      guard = guardTransitionOut : Guard{},
      name = nameTransitionOut
    }
  }
} //fin ckeckonly domain spem actionState: ActionState
```



```

enforce domain bpmn subprocess: SubProcess
{
// El id se forma concatenando el string "id" con el nombre de la actividad.
  id = "id"+ nameActivity,
  name = nameActivity,

// Para definir el template para los conectores BPMN de entrada se utiliza la
// información de las transiciones de entrada al step de SPEM.
  incomingConnectors = sequenceConnectorInBPMN : SequenceConnector
  {
    id = "id"+nameTransitionIn,
    name = nameTransitionIn,
    conditionExpression = conditionExpressionIn : Expression{},
    quantity = 1
  },

// En la definición del template para los conectores BPMN de salida se
// utiliza la información de las transiciones de entrada al step de SPEM.
  outgoingConnectors = sequenceConnectorOutBPMN : SequenceConnector
  {
    id = "id"+nameTransitionOut,
    name = nameTransitionOut,
    conditionExpression = conditionExpressionOut : Expression{},
    quantity = 1
  },

// Los lanes son creados con la información del Performer que llevan a cabo
// la actividad de SPEM.
// Además en este template se crea el pool al que pertenece el lane, el pool se
// crea con un id constante para asegurar la creación de un único pool en la
// transformación.
  lanes = lane : Lane{} //fin de Lane
}; // fin de domain bpmn subprocess: SubProcess
where
{
// Realiza el mapping entre los pasos de la actividad y las tareas del
// subProceso
stepSPEMToTaskBPMN(activity, subProcess);

// Obliga el cumplimiento de la relación
// ProcessRoleToOrganizationRoleBPMN que define la correspondencia
// entre un ProcessRole de SPEM y OrganizationRole de BPMN.
  processRoleToOrganizationRoleBPMN(processRole, participantBPMN);

// Exige el cumplimiento de la relación guardToExpression que define la
// correspondencia entre un Guard de SPEM y una Expression de BPMN.
  guardToExpression(guardTransitionIn, condicionExpressionIn);
  guardToExpression (guardTransitionOut, condicionExpressionOut);

// Define la creación del lane a partir e la información del performe.
  processRoleToLane(performe,lane);
}

```

6 CONCLUSIONES

Este trabajo se centra en hacer una contribución a la optimización de los procesos de desarrollo de software, considerando al proceso de desarrollo de software como un proceso de negocio particular, y con esto, pueden ser automatizados en todo o en parte a través de un motor de workflow. Para lograr esta automatización de los procesos de desarrollo de software, se definió una transformación de actividades SPEM a subprocesos de un workflow para poder lograr la automatización total o parcial de su gestión. Teniendo en cuenta esto, las actividades del proceso de desarrollo de software se transforman en una especificación de subprocesos BPMN y así poder utilizar un motor workflow que siga el estándar de la WfMC u Oasis (dependiendo del lenguaje de implementación elegido), y de esta forma, automatizar total o parcialmente la gestión de las actividades de los procesos de desarrollo de software especificados con el estándar de la OMG denominado SPEM.

La relación presentada en este trabajo representa una parte central de la transformación entre los metamodelos SPEM y BPMN, debido a que define la relación entre las actividades del proceso de desarrollo de software y los subprocesos BPMN.

Usar como objetivo de la transformación el metamodelo BPMN, tiene la ventaja que los modelos obtenidos son independientes de la plataforma de automatización, esto es, independiente del motor workflow que se utilice para la administración de los proyectos de desarrollo de software basados en alguna metodología especificada en SPEM.

Un caso práctico de la aplicación de la transformación, es su aplicación a la especificación en SPEM de las actividades del SmallRUP, esto da como resultado una especificación de subprocesos workflow en BPMN. Luego, aplicándole a la especificación BPMN el mapping a BPEL4WS (o XPDL), se obtiene una especificación de procesos que puede ser tomada por cualquier motor workflow que interprete una especificación BPEL4WS (o XPDL), y así, poder administrar automáticamente, por medio de un motor workflow, las actividades de los proyectos de desarrollo de software que utilicen como metodología de desarrollo a SmallRup.

El beneficio de esta automatización también se aprecia teniendo en cuenta el dinamismo de los cambios en los procesos de desarrollo de software, con lo cual, cualquier cambio en la especificación de las actividades de un proceso de desarrollo de software puede ser propagado a la especificación Workflow de dichas actividades y así adaptar rápidamente la especificación del workflow para la automatización de las actividades de los procesos de desarrollo de software.

Esta transformación optimiza la construcción del software debido a que se dispone de un sistema automatizado (motor workflow) que administrará los recursos y organizará a un equipo de ingenieros de software en el transcurso del desarrollo de un proyecto en particular. El proceso de desarrollo adopta todas las ventajas propias de un proceso de negocio.

REFERENCIAS

- [1] Daniel K.C. Chan, Karl R.P.H. Leung, "Software Development as a Workflow Process," apsec, p. 282-291, Fourth Asia-Pacific Software Engineering and International Computer Science Conference (APSEC'97 / ICSC'97), IEEE 1997.
- [2] N. Debnath, D. Riesco, G. Montejano, et al, "Supporting the SPEM with a UML Extended Workflow Metamodel", ACS/IEEE International Conference on Computer Systems and Applications (AICCSA'06). Conference to be held in Dubai/Sharjah during March 8-11, 2006, www.ieee.org.
- [3] Anthony Barnes, Jonathan Gray, "COTS, Workflow, and Software Process Management: An Exploration of Software Engineering Tool Development" aswec, p. 221-232, 2000 Australian Software Engineering Conference, IEEE 2000.
- [4] Object Management Group, "Software Process Engineering Metamodel Specification"; An Adopted Specification of the Object Management Group, Inc; Version 1.1 formal/05-01-06; January 2005, <http://www.omg.org/docs/formal/05-01-06.pdf>, último acceso Marzo 2006.

- [5] Object Management Group “Business Process Modeling Notation (BPMN) Specification”. Final Adopted Specification dtc/06-02-01, http://www.bpmn.org/Documents/OMG_Final_Adopted_BPMN_1-0_Spec_06-02-01.pdf, último acceso Mayo 2007.
- [6] Object Management Group, “Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification” Final Adopted Specification ptc/05-11-01, <http://www.omg.org/docs/ptc/05-11-01.pdf>, último acceso Junio 2007
- [7] Object Management Group “Meta Object Facility (MOF) Core Specification” OMG Available Specification. Version 2.0. formal/06-01-01, <http://www.omg.org/docs/formal/06-01-01.pdf>, último acceso Diciembre 2006.
- [8] Rob Allen, Open Image Systems Inc., United Kingdom Chair, WfMC External Relations Committee; “The Workflow Handbook 2001”; Workflow Management Coalition; October 2001.
- [9] Workflow Management Coalition, Workflow Standard – Workflow Process Definition Interface -XML Process Definition Language, Workflow Management Coalition , WfMC-TC-1025, 2002, http://www.wfmc.org/standards/docs/TC-025_10_xpdl_102502.pdf , último acceso Octubre 2006.
- [10] BEA, IBM, Microsoft, SAP and Siebel, “Business Process Execution Language for Web Services Version 1.1” , S. Thatte, et al., May 2003, <ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf>, último acceso, Mayo 2007.
- [11] OFBiz Workflow Engine, <http://incubator.apache.org/ofbiz/docs/workflow.html>, último acceso Abril 2007.
- [12] Open Business Engine, <http://obe.sourceforge.net/> , último acceso Marzo 2007.
- [13] IBM, “WebSphere Process Server”, <http://www-306.ibm.com/software/integration/wps/>, último acceso Junio 2007
- [14] Oracle, “BPEL Process Manager” <http://www.oracle.com/technology/products/ias/bpel/index.html>, último acceso Abril 2007.
- [15] Object Management Group, BPMN Documents “BPMNModel UML Documentation”. Draft Specification, <http://www.bpmn.org/Documents/BPMNMetaModel.zip> , último acceso Octubre 2006.
- [16] Gary Pollice “Using the RUP for small projects: Expanding upon Extreme Programming”, A Rational Software White Paper – 04/08/15, <ftp://ftp.software.ibm.com/software/rational/web/whitepapers/2003/tp183.pdf>, último acceso Mayo 2007.