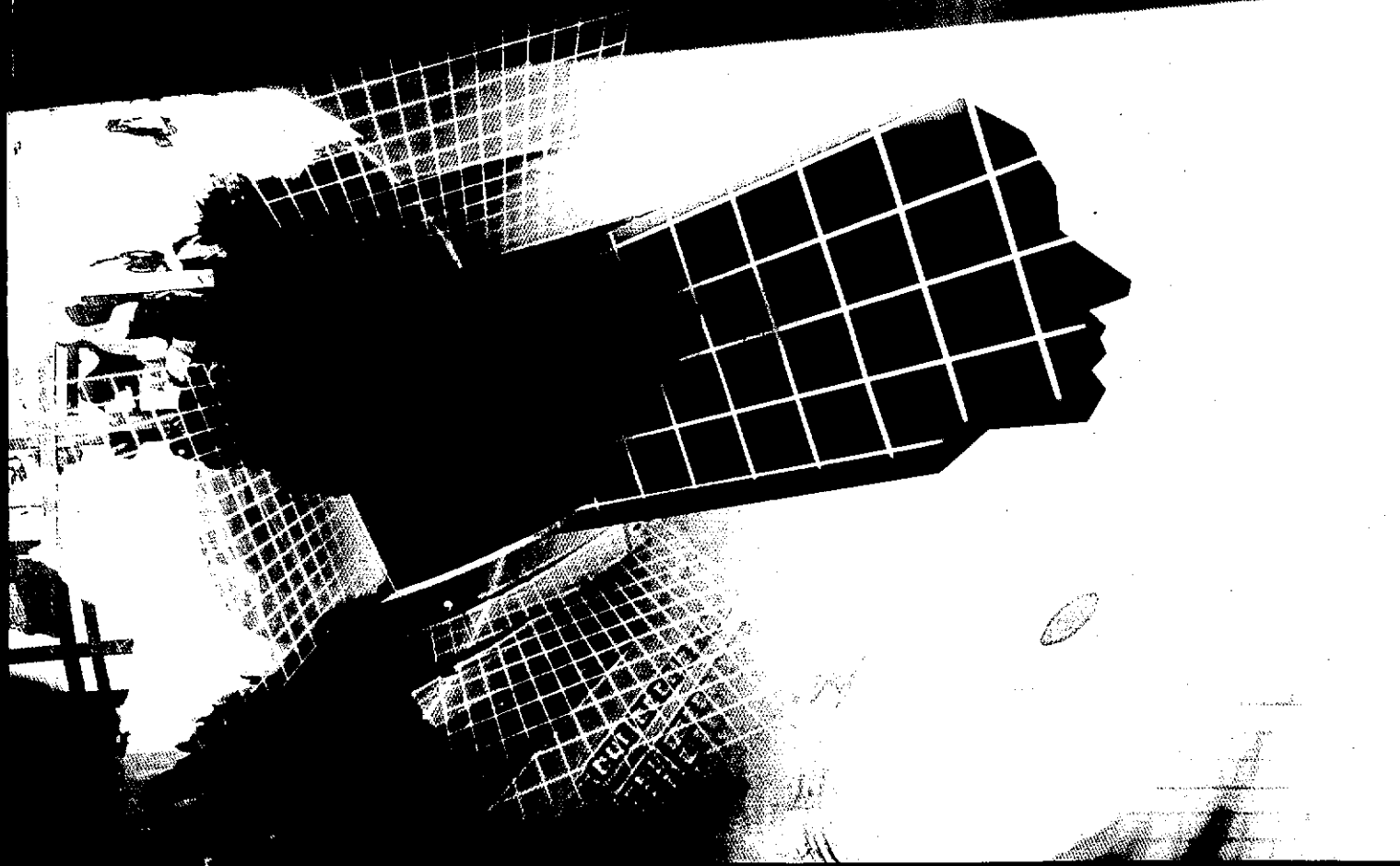




VI Jornadas Iberoamericanas

de Ingeniería del Software
e Ingeniería del Conocimiento

DEL 31 DE ENERO AL 2 DE FEBRERO
LIMA - PERÚ



DEPARTAMENTO
DE **INGENIERÍA**
SECCIÓN INGENIERÍA INFORMÁTICA



90
AÑOS

PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

ISBN 978-9972-2885-1-7



JIISIC'07

VI Jornadas Iberoamericanas de
Ingeniería del Software e Ingeniería del Conocimiento

Lima-Perú
31 de enero al 2 de febrero de 2007

Editado y Compilado por:

Facultad de Ciencias e Ingeniería
Departamento de Ingeniería

Maynard Kong
José Antonio Pow-Sang
Manuel Francisco Tupia
Luis Alberto Flores



90
AÑOS

PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

VI Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento-IIISIC'07

Compilado por:

Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú
Departamento de Ingeniería de la Pontificia Universidad Católica del Perú

Editado por:

Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú
Departamento de Ingeniería de la Pontificia Universidad Católica del Perú
Maynard Kong Wong, José Antonio Pow-Sang Portillo, Manuel Francisco Tupia Anticona y Luis Alberto Flores García.

Primera edición: enero de 2007

Hecho el Depósito Legal en la Biblioteca Nacional del Perú N°2007-00571

ISBN N° 978-9972-2885-1-7

Comité Permanente

Silvia Teresita Acuña, Universidad Autónoma de Madrid, España
 Manoel Mendonça, Universidade Salvador, Brasil
 Oscar Dieste, Universidad Complutense de Madrid, España

Comité Organizador

José Antonio Pow-Sang, Pontificia Universidad Católica del Perú (**chair**)
 Manuel Tupia, Pontificia Universidad Católica del Perú
 Luis Flores, Pontificia Universidad Católica del Perú
 Felipe Solari, Pontificia Universidad Católica del Perú

Comité de Programa

Maynard Kong, Pontificia Universidad Católica del Perú, Perú (**chair**)
 Raul Aguilar, Universidad Autonoma de Yucatán, México
 Idoia Alarcon, Universidad Autónoma de Madrid, España
 Luis Alberto Alvarez, Universidad Austral, Chile
 Marco Alvarez, Utah State University, EEUU
 Pedro Antunes, Universidade de Lisboa, Portugal
 Joao Araujo, Universidade Nova de Lisboa, Portugal
 Marianela Aveledo, Universidad Simon Bolivar, Venezuela
 Pere Botella, Universitat Politècnica de Catalunya, España
 David Camacho, Universidad Autonoma de Madrid, España
 Francisco Camargo, ITESM, México
 Zalatiel Carranza, Universidad de Lima, Perú
 Dante Carrizo, Universidad Complutense de Madrid, España
 Luca Cernuzzi, Univ. Católica Ntra. Señora de la Asunción, Paraguay
 Sergio Coronado, University of Luxembourg, Luxemburgo
 Ernesto Cuadros-Vargas, Universidad Católica San Pablo, Perú
 Angelica de Antonio, Universidad Politècnica de Madrid, España
 Amador Duran, Universidad de Sevilla, España
 Juan Vicente Echagüe, Universidad de la República, Uruguay
 Yadran Eterovic, Pontificia Universidad Católica de Chile, Chile
 Mariano Fernandez, Universidad CEU San Pablo, España
 Xavier Ferre, Universidad Politècnica de Madrid, España
 Ramon Garcia, Instituto Tecnológico de Buenos Aires, Argentina
 Francisco Jose Garcia, Universidad de Salamanca, España
 Luis Guerrero, Universidad de Chile, Chile
 Ricardo Imbert, Universidad Politècnica de Madrid, España
 Mario Jino, Universidade Estadual de Campinas, Brasil
 Nora La Serna, Universidad Nacional Mayor de San Marcos, Perú
 Guillermo Licea, Universidad Autónoma de Baja California, México
 Marta Lopez, Universidad Complutense de Madrid, España
 Jose Antonio Macias, Universidad Autónoma de Madrid, España
 Esperanza Marcos, Universidad Rey Juan Carlos, España
 Victor Hugo Medina, Universidad Distrital Fco. José Caldas, Colombia
 Nelson Medinilla, Universidad Politècnica de Madrid, España
 Ana María Moreno, Universidad Politècnica de Madrid, España
 Jaime Muñoz, Universidad Autónoma de Aguascalientes, México
 Melvin Perez, CAM Informatica, República Dominicana
 Claudia Pons, Universidad Nacional de la Plata, Argentina
 Angel Puerta, Redwhale Software, EEUU
 Isidro Ramos, Universitat Politècnica de Valencia, España
 Gustavo Rodríguez, INAOE, México
 María Isabel Sanchez Segura, Universidad Carlos III de Madrid, España

Comité de Programa (continuación)

René Santaolaya Salgado, CENIDET, México
Miguel Angel Serrano, CIMAT, México
Almudena Sierra, Universidad Rey Juan Carlos, España
Enrique Sierra, Instituto Tecnológico de Buenos Aires, Argentina
Francisco Tirado, Universidad Complutense de Madrid, España
Ambrosio Toval, Universidad de Murcia, España
Jorge Triñanes, Universidad de la República, Uruguay
Raimundo Vega, Universidad Austral, Chile
Sira Vegas, Universidad Politécnica de Madrid, España
Silvia Regina Vergilio, Universidade Federal do Paraná, Brasil
Monica Villavicencio, Escuela Superior Politécnica del Litoral, Ecuador
Marcello Visconti, Universidad Técnica Federico Santa María, Chile
Aurora Vizcaino Barceló, Universidad de Castilla-La Mancha, España

Colaboradores en el Proceso de Revisión

Abel Gómez
Alejandro Hossian
Alex Bustos
Aurora Pozo
César J. Acuña
Enrique Fernandez
Fernando Molina
Fuensanta Medina Domínguez
Jaime Navón
Jennifer Pérez
Joaquín Nicolás
José Ángel Olivas
Jose Arturo Mora Soto
Jose Carsí
José María Cavero
Luis Flores
Manuel Tupia
Maria Alejandra Ochoa
Marisa Cogliati
Miguel Ángel Martínez Aguilar
Nelly Condori-Fernandez
Norberto Millo
Paola Britos
Percy Pari Salas
Sonia Pamplona

Prólogo

Este volumen contiene los trabajos aceptados y presentados en las VI Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento (JIISIC'07) celebradas en Lima, Perú, del 31 de enero al 2 de febrero de 2007. Desde su edición inicial en 2001, las JIISIC han demostrado ser el foro de reunión más importante, a nivel Iberoamericano, de investigadores y profesionales interesados en ambas disciplinas.

El evento actual es la continuación de la labor iniciada en las JIISIC'01, celebrada en Buenos Aires (Argentina), JIISIC'02 en Salvador de Bahía (Brasil), JIISIC'03 en Valdivia (Chile), JIISIC'04 en Madrid (España) y JIISIC'06 en Puebla (México).

En la presente convocatoria se han recibido 88 artículos de calidad científica para su evaluación. Cada trabajo ha sido evaluado por al menos 2 revisores y se ha contemplado la resolución de divergencias, que por cierto han sido muy pocas. Finalmente fueron aceptados 54 artículos de autores procedentes de Argentina, Brasil, Colombia, Corea del Sur, Cuba, Chile, Ecuador, España, Estados Unidos de América, México, Perú y Uruguay. Además de las sesiones técnicas, se aceptaron cuatro tutoriales.

Es preciso indicar que todo esto no hubiera sido posible sin la colaboración de muchas personas. Por ello queremos agradecer especialmente a los miembros del Comité de Programa por su excelente y desinteresada labor, necesaria para renovar la calidad y prestigio ganado. También queremos destacar el enorme esfuerzo de Manuel Tupia, Luis Flores y Felipe Solari, miembros del Comité Organizador, sin cuyo trabajo no hubieran podido celebrarse estas Jornadas. Nuestro agradecimiento al Ing. Eduardo Ismodes, decano de la Facultad de Ciencias e Ingeniería, y al Ing. Kurt Paulsen, jefe del Departamento de Ingeniería, por el gran apoyo que nos han brindado. Por último, pero no al final, expresamos nuestro sincero agradecimiento a todos los autores que aportaron sus contribuciones al evento.

Maynard Kong
Presidente del Comité de Programa

José Antonio Pow-Sang
Presidente del Comité Organizador

ÍNDICE

ARTÍCULOS	1
Sesión 1a: Bases de datos y Minería de Datos	
Un Modelo de Proceso para Educación de Requisitos en Proyectos de Data Mining <i>José Gallardo Arancibia, Óscar Marbán Gallego, Claudio Meneses Villegas</i>	3
Optimizing Lies in State Oriented Domains based on Genetic Algorithms <i>A. Zylberberg, E. Calot, J. Iervache, H. Merlino, P. Britos, R. Garcia-Martinez</i>	11
Extension del Lenguaje SQL con Nuevas Primitivas SQL para el Descubrimiento de Reglas de Clasificación <i>Ricardo Timarán Pereira</i>	19
Sesión 1b: Pruebas de Software, Validación y Verificación. Prop. de Inteligencia Artificial a IS	
Certificación de Propiedades Usando Distintos Probadores de Teoremas: Un Caso de Estudio <i>J. Santiago Jorge, Victor M. Gulias, Laura M. Castro</i>	27
GraspKM en la Recuperación de la Estructura de Software <i>Erick Vicente, Manuel Tupia, Luis Rivera</i>	35
Testing Exploratorio en la Práctica <i>Beatriz Pérez, Amparo Pittier, Mariana Travieso, Mónica Wodzislowski</i>	43
Sesión 1c: Ingeniería de Requerimientos	
Una Propuesta para la Elicitación de Requerimientos de Seguridad Basada en Preguntas <i>Vianca Vega Z., Gloria Gasca H., Edmundo Tovar C., José Carrillo V.</i>	51
Um Processo de Engenharia de Requisitos Baseado em Reutilização de Ontologias e Padrões de Análise <i>Ricardo de Almeida Falbo, Aline Freitas Martins, Bruno Marques Segrini, Gleison Baiôco, Rodrigo Dal Moro, Julio Cesar Nardi</i>	59
Elicitación de Requisitos Empleando UN-Lencep y Esquemas Preconceptuales <i>Carlos Mario Zapata J., Fernando Arango I.</i>	69
Sesión 2a: Ingeniería del Conocimiento, Bases de Datos y Minería de Datos	
Onto-DOM: A Question-Answering Ontology-Based Strategy for Heterogeneous Knowledge Sources <i>Mariel Alejandra Ale, Cristian Gerarduzzi, Omar Chionti, Maria Rosa Galli</i>	79
Knowledge Engineering for a Fuzzy Power Plant Process Controller <i>Youngchul Bae, MalRey Lee, Sang Doo Shin, Thomas Gatton, Yigon Kim</i>	87
Un Acercamiento a los Modelos Multidimensionales Espacio Temporales <i>Francisco Javier Moreno Arboleda, Fernando Arango Isaza</i>	93

Sesión 2b: Ontologías, Metodologías, Patrones y Frameworks

Asynchronous Merging of Software Ontologies: An Experience <i>Nicolas Anquetil, Aurora Vizcaíno, Francisco Ruiz, Kathia Oliveira, Mario Piattini</i>	99
Hacia una Metodología Orientada al Conocimiento para la Educación de Requisitos en Ingeniería del Software <i>Alejandro Hossian, Enrique Sierra, Ramón García-Martínez, María Alejandra Ochoa, Paola Britos</i>	107
Casos de (Re)Uso: Uma Abordagem para Reuso de Software Interativo Dirigida por Casos de Uso e Padrões Concretos de Interação, <i>Augusto Abelin Moreira, Marcelo Soares Pimenta</i>	115

Sesión 2c: Ingeniería de Requerimientos, Arquitecturas y Diseño de Software

Modelado de Aplicaciones con Procesos Concurrentes y Distribuidos <i>Daniel A. Giulianelli, Rocio A. Rodriguez, Pablo M. Vera</i>	123
Requisitos No Funcionales: Evaluando el Impacto de Decisiones <i>Marcela Quispe-Cruz, Nelly Condori-Fernández</i>	133
Atributos Contextuales Relevantes para la Selección de Técnicas de Educación de Requisitos <i>Dante Carrizo, Oscar Dieste</i>	143

Sesión 3a: Arquitecturas y Diseño de Software

Evaluación de Arquitecturas de Software con ATAM (Architecture Tradeoff Analysis Method): Un Caso de estudio <i>Andrea Delgado, Alberto Castro, Martín Germán</i>	151
Transformación de Vistas Arquitectónicas Orientada por Modelos <i>Rogelio Limón Cordero, Isidro Ramos Salavert, Arturo Aragon Sorroza</i>	161
Analyzing and Designing Software Architecture Views driven by their Relationships <i>Rogelio Limón Cordero, Isidro Ramos Salavert, Maricela Morales Hernández, Jorge Zuráte Perez</i>	171

Sesión 3b: Métodos de Diseño, Modelado de Dominio y Meta-Modelado

Aplicando MDA al Diseño de un Datawarehouse Temporal <i>Carlos Neil, Claudia Pons</i>	181
Estrategias de Detección de "Feature Envy" en Aplicaciones Java <i>Carlos Angarita Márquez, Silvia Takahashi Rodriguez</i>	191
Un Caso Práctico en MDA para Construir Aplicaciones JEE5 y .NET <i>Andres Yie, Juan Bohórquez, Rubby Casallas</i>	201

Sesión 3c: Calidad en el Software

Evolução de um Processo Ágil de Desenvolvimento baseado em framework, <i>Franciene Duarte Gomes, Maria Istela Cognin</i>	211
Desarrollo de un Código de Métricas para Pequeñas Empresas Ecuatorianas Desarrolladoras de Software <i>Raúl González Carrión, Henry Hernandez Rendón, Mónica Villavicencio Cabezas</i>	221

A Organização de uma Máquina de Processo e a Melhoria do Processo de Produção de Software em um Ambiente de Fábrica <i>José A. Fabri, André L.P. Trindade, Alexandre L'Erário, Marcelo S. de P. Pessoa</i>	229
---	-----

Sesión 4a: Modelado de Procesos

A Minimal OCL-based Profile for Model Transformation <i>Roxana Giandini, Gabriela Pérez, Claudia Pons</i>	237
Extensión MDA (Model Driven Architecture) para Proceso Basado en RUP (Rational Unified Process), <i>Andrea Delgado, Natacha Carballal, Catalina Rapetti</i>	247
Organización de Conocimientos en Procesos de Ingeniería de Software por Medio de Modelado de Procesos: una Adaptación de SPEM <i>Oscar M. Rodríguez-Elias, Ana I. Martínez-García, Aurora Vizcaino, Jesús Favela, Mario Piattini</i>	257

Sesión 4b: Ing. del Software basada en Componentes, Usabilidad e Interacción Persona-Computadora

Un modelo de Componentes para el Diseño y Ejecución de Procesos de Colaboración basado en ThinkLets <i>Víctor Alberto Hermida, Carlos Hernán Tobar, Julio Ariel Hurtado, César A. Collazos</i>	267
Monitoreo del Desempeño de los Factores de Seguridad de una Transacción Web a través de la Interfaz de Usuario <i>R. Mendoza González, J. Muñoz Arteaga, F. J. Álvarez Rodríguez, M. Vargas Martín</i>	275

Sesión 4c: Métricas e Ingeniería del Software Empírica

Experimento Exploratorio para la Validación de Medidas para Modelos de Procesos de Negocio <i>Elvira Rolón, Félix García, Francisco Ruiz, Mario Piattini</i>	283
Estudio Experimental en Equipos de Desarrollo de Software sobre las Relaciones entre Personalidad, Satisfacción y Calidad del Producto <i>Marta Gómez, Silvia T. Acuña, Ramón Rico</i>	293
Estimación basada en Escenarios Principales, <i>José Cao, Enrique Fernández, Hernán Merlino, Alejandro Hossain, Enrique Sierra, Eduardo Diez, Paola Britos, Ramón García-Martínez</i>	301

Sesión 5a: Modelos de Calidad

RevisionCASE, Herramienta para Gestionar Revisiones a Proyectos de Software Empleando Razonamiento Basado en Casos <i>Martha Delgado Dapena, Sofía Álvarez Cardenas, Josué Carralero Enaga, Javier Travieso Arencibia, Iren Lorenzo Fonseca, Alejandro Rosete Suárez</i>	309
Modelo Liviano de Calidad para la Mejora de Procesos de Desarrollo Software <i>Carmen J. Sánchez, María E. Solís, Francisco J. Pino, Julio A. Hurtado</i>	315
Diseño y Desarrollo de un Entorno Integrado para Simuladores de Entrenamiento de Procesos Industriales <i>Pedro A. Corcuera</i>	325

Sesión 5b: Métricas e Ingeniería del Software Empírica

Avaliando a Relação entre Tamanho-Complexidade e Número de Defeitos de Software em Nível de Módulo <i>Waldo Luis de Lucca, Plínio R. S. Vilela, Mario Jino</i>	333
---	-----

<i>Empirically Evaluating the Usefulness of Software Visualization Techniques in Program Comprehension Activities</i> <i>Glanco de F. Carneiro, Angelo C. Araujo Orrico, Manoel G. de Mendonça Neto</i>	341
--	-----

Sesion 5c: Modelado y Mejora de Procesos

Un método de Evaluación Ágil del Proceso Software: Agile SPI - Process Assessment Method <i>Julio Ariel Hurtado, César Pardo, Luis Fernández, Juan Carlos Vidal</i>	349
MUM - Proceso de Desarrollo de Software Modularizado, Unificado y Medible <i>Beatriz Pérez, Lucía Pedrana, Marcelo Bellini</i>	359
Enfoque de Metamodelado y Multiformalismo Aplicado al Proceso Software usando ATOM3 <i>Mabel del V. Sosa, Silvia T. Acuña, Juan de Lara</i>	367
<i>O Papel do CMMI na Configuração de um Meta-Processo de Produção de Software com Características Fabris: Um Estudo de Caso</i> <i>José Augusto Fabri, André Luiz Presende Trindade, Márcio Silveira, Marcelo S. de Paula Pessoa</i>	375

Sesión 6a: Modelos de Calidad

Utilización de un Método ad hoc para el Mejoramiento de Procesos con MoProSoft <i>Verónica Martínez, Yessica Gómez, Hanna Oktaba, Angélica Urrutia, Rodolfo Villarroel</i>	385
Perfil UML 2.0 para Aplicaciones de Monitoreo Ambiental <i>Adriana B. Urciuolo, Rodolfo J. Iturraspe, Ezequiel Moyano</i>	393
Una Abstracción Posible del Toyotismo Subtensa en un Modelo Concurrente de Ciclo de Vida de Software <i>Alejandro Estayno, Marcelo Estayno, Alicia Mon</i>	403

Sesión 6b: Aplicaciones Industriales y Cómputo Móvil

Aplicación de la Tecnología Bluetooth Orientada a la Integración de Servicios de Internet en Dispositivos Móviles <i>Juan Guillermo Torres Hurtado, Álvaro Bernal Noreña</i>	411
Modelo Multiagente en Sistemas de Misión Crítica Aplicado al Control de Tráfico Aéreo Bajo el Concepto de Free Flight <i>Victor Battista, Jorge Ierache, Paola Britos, Darío Rodríguez, Ramón García-Martínez</i>	419
El Problema Cinemático en Manipuladores Robóticos Industriales un Abordaje de Solución mediante Redes Neuronales Artificiales <i>Alejandro Hossian, Enrique Sierra, Enrique Fernández, Paola Britos, Ramón García-Martínez</i>	427

Sesión 6c: Educación en Ing. de Software e Ing. del Conocimiento, Informática Educativa

Estudio, Implantación y Resultados de la Adaptación Espacio Europeo de Educación Superior en las Asignaturas de Programación de la Titulación de Informática de la Universidad de Málaga <i>Jose Luis Pastrana, Maria Victoria Belmonte, Carlos Cotta, Antonio Fernández, Enrique Soler, Maria Inmaculada Yague</i>	435
Ontologías en el Desarrollo de Entornos Virtuales para Entrenamiento <i>Raúl A. Aguilar, Angélica de Antonio, Fidel Rojas-Toledo</i>	445

Sesión 6d: Mejora de Procesos

Experiencia en Team Software Process (TSP) y Mejoras de Estimación, Calidad y Productividad de los Equipos en la Gestión del Software 451
Gonzalo Cuevas, José Calvo Manzano, Tomas San Felu, Sussy Bayona

Aprendizaje por Refuerzos en Problemas de Planeamiento con Restricciones 459
Pedro E. Colla, Ernesto Martínez

Tutoriales

Uso de Esquemas Preconceptuales para la Generación Automática de Diagramas de Clases, Comunicación y Máquina de Estados 469
Carlos Mario Zapata J.

Como Organizar um Processo Fabril de Produção de Software 473
José Augusto Fabri, Marcelo S. de Paula Pessoa

El Uso de la Incertidumbre como Herramienta en la Ingeniería de Software 477
Nelson Medinilla Martínez

Aplicación de Técnicas de Aprendizaje Cooperativo en la Enseñanza del Desarrollo de Software 481
Pedro Campos, Luis Alberto Flores, José Antonio Pow-Sang, Claudia Zapata

Organización de conocimientos en procesos de ingeniería de software por medio de modelado de procesos: una adaptación de SPEM

Oscar M. Rodríguez-Elias^{1,2}, Ana I. Martínez García², Aurora Vizcaíno³, Jesús Favela²,
Mario Piattini³

¹UABC-Facultad de Ciencias, Ensenada, B.C., México

omrodriguez@ieee.org

²CICSE-Departamento de Ciencias de la Computación, Ensenada, B.C., México

{martinea | favela}@cicse.mx

³Universidad de Castilla-La Mancha, Escuela de Informática, Ciudad Real, España

{aurora.vizcaino | mario.piattini}@uclm.es

Resumen

Este trabajo presenta una adaptación del Software Process Engineering Metamodel (SPEM), enfocada en apoyar en el estudio de procesos de ingeniería de software desde una perspectiva del conocimiento involucrado en los mismos. El objetivo es que los modelos de procesos desarrollados, puedan ser de utilidad para el estudio de necesidades de conocimiento por parte de los encargados del proceso. Lo que a su vez puede servir para la definición de estrategias, métodos, sistemas, etc., enfocados en dar solución a las necesidades de conocimiento por medio de la aplicación de técnicas de administración del conocimiento, que estén conectadas con las necesidades reales de quienes desempeñan las actividades dentro de los procesos. La propuesta es ejemplificada por medio de un caso de estudio donde se catalogan algunos de los temas de conocimiento que son relevantes dentro de una parte de un proceso de mantenimiento de software.

Palabras clave: Modelado de procesos, ingeniería de software, administración de conocimiento, SPEM.

1. Introducción

La ingeniería de software (IS) es conocida por ser una actividad que involucra distintos tipos de conocimiento, de ahí la necesidad creciente en distintas organizaciones de desarrollo de software para definir métodos y técnicas que les ayuden a manejar el conocimiento que poseen de una mejor manera [1].

La introducción de sistemas de apoyo a la administración del conocimiento (AC) en grupos de IS ha aportado diversos beneficios [2]. Sin embargo, los sistemas que comúnmente son implantados en estas organizaciones poseen características que evitan su uso [3]. Particularmente, este tipo de problemas se deriva de que los sistemas de AC comunes están desconectados de los procesos de trabajo de las organizaciones, y por lo tanto, no están enfocados en resolver sus necesidades de conocimiento reales [4].

Antes de desarrollar o implantar sistemas de AC en un grupo específico, es necesario hacer un estudio de las necesidades de conocimiento del mismo [5], haciendo un análisis de los procesos de trabajo desde un enfoque centrado en cómo fluye y se maneja el conocimiento a través de los procesos realizados por el grupo en cuestión [6]. Un paso importante en este análisis, es el modelado de los procesos [7].

Este trabajo presenta una adaptación hecha al Software Process Engineering Metamodel (SPEM) [8], con el fin de usarlo como apoyo en el estudio de necesidades de conocimiento en grupos de IS. El resto de este artículo se estructura de la siguiente manera: la sección 2 presenta una introducción al trabajo previo en el uso del modelado de procesos para estudiar flujos de conocimiento, y en particular, a su aplicación en procesos de IS. A continuación, en la sección 3 se describen las adaptaciones realizadas a SPEM. Posteriormente, en la sección 4 se da un ejemplo de la aplicación de las extensiones por medio de un caso de estudio. La sección 5 presenta las principales lecciones aprendidas en el caso de estudio. Finalmente, en la sección 6 se resumen las conclusiones de este trabajo.

2. Modelado de procesos con enfoque en el flujo del conocimiento

Para que los sistemas de AC sean realmente de utilidad, es importante que estos ayuden a que las personas encargadas de los procesos de las organizaciones puedan contar con el conocimiento que requieren para llevar a cabo sus actividades [5, 9]. De aquí que, el flujo del conocimiento a través de los procesos de trabajo se vuelve un aspecto central de los sistemas de AC [10]. En este sentido, un primer paso para brindar apoyo por medio de la AC en una organización, es entender cómo fluye el conocimiento a través de sus procesos de trabajo [11]; para posteriormente identificar las necesidades de conocimiento reales y utilizarlas como base para proponer alternativas de solución.

De lo anterior se puede observar cómo el uso de técnicas de análisis de procesos puede ser de gran beneficio al momento de definir estrategias de AC que estén integradas al trabajo real de las organizaciones, y que consideren las necesidades de conocimiento de las mismas [6]. Un medio útil para la identificación de estas necesidades es el modelado de procesos con un enfoque al análisis del flujo del conocimiento [12-14].

El modelado de procesos puede apoyar en la identificación del conocimiento que entra y sale de las actividades de un proceso, las fuentes donde es obtenido o almacenado, y los flujos del conocimiento a través de actividades, personas, u otro tipo de fuentes [15, 16]. Además, a través del modelado es posible detectar los problemas que puedan estar afectando al flujo del conocimiento, facilitando, a su vez, la definición de estrategias para mejorarlo [12, 14].

Existen una gran variedad de técnicas de modelado de procesos, cada una de las cuales se enfoca en aspectos específicos del proceso, e incluso en tipos de procesos particulares [7, 17]. Es importante que la técnica empleada para el estudio de un proceso desde la perspectiva del flujo del conocimiento, cuente con primitivas que faciliten la representación del conocimiento usado, creado, etc. durante las actividades [13]. Sin embargo, dado que la mayoría de las técnicas han sido definidas con el propósito de la reingeniería de procesos de negocios, por lo general no cuentan con los elementos que permitan representar el conocimiento involucrado en el proceso desde la perspectiva del flujo del conocimiento [13, 18].

Debido a lo anterior se han propuesto algunas técnicas para modelado de procesos, con un enfoque al conocimiento involucrado en el mismo, que han sido empleadas en diversos campos [18-21]. Si bien algunas de ellas han sido usadas en organizaciones de

desarrollo de software, estas no han sido definidas específicamente para procesos de IS.

2.1 Modelado de procesos de ingeniería de software

Curtis et al. [7] mencionan que mucha de la investigación en modelado de procesos ha sido realizada en organizaciones de desarrollo de software debido a que la comunidad de IS está acostumbrada al modelado formal. Incluso varios de los lenguajes que han sido usados para modelado de procesos se han derivado de lenguajes usados para análisis, diseño e incluso programación de sistemas de software [7, 17].

Por otra parte, existen distintos trabajos enfocados en identificar el conocimiento requerido por los encargados del desarrollo y mantenimiento de software [22-25]. Sin embargo, pocos trabajos han aplicado técnicas de ingeniería de procesos para identificar requerimientos de conocimiento en grupos específicos. No obstante que para aplicar la AC en un grupo en particular, es importante identificar las necesidades y el contexto de dicho grupo [9].

La técnica de gráfica rica [26], es una de las pocas de uso extendido que ha sido usada para analizar flujos de conocimiento en procesos de IS [12, 14]. Esta técnica es muy útil para obtener una visión general del proceso, dado que por su generalidad, es posible modificarla según las necesidades. Sin embargo, la falta de formalidad en los modelos tiene el inconveniente de que se dificulta la identificación de algunos detalles de bajo nivel que podrían ser importantes, es por eso que se recomienda usarla en etapas tempranas del modelado, y complementarla posteriormente con técnicas más formales [26, 27].

Otros trabajos como [20, 21, 28] han empleado sus propios enfoques de modelado de procesos, e incluso proponen herramientas para apoyar en el modelado de los mismos. Sin embargo, ninguna de estas ha sido desarrollada para procesos de IS, además, su utilización requiere que se tengan disponibles las herramientas de modelado que proponen. Otros enfoques usados son los derivados de la ingeniería del conocimiento [29]; el problema de estas técnicas es que no necesariamente son de ayuda para identificar posibilidades de mejoras en los procesos [13].

Si bien el uso de técnicas genéricas de modelado de procesos puede ser de ayuda para estudiar flujos de conocimiento, es recomendable proveer elementos de modelado que permitan representar aspectos particulares del tipo de proceso estudiado, debido a que la representación explícita de estos elementos facilita su análisis [17]. En este sentido, para estudiar flujos de conocimiento en grupos de desarrollo y mantenimiento

de software, es recomendable usar una técnica de modelado de procesos que esté enfocada a procesos de IS, y que a su vez cuente con primitivas que permitan la representación explícita del conocimiento involucrado en el proceso.

2.2 SPEM

SPEM es un metamodelo diseñado para describir procesos y sus componentes, siguiendo un enfoque de modelado orientado a objetos con base en UML [8]. SPEM ha sido desarrollado como un perfil de UML; es decir que extiende los mecanismos de UML de una forma estandarizada, con el propósito de modelar procesos de desarrollo de software.

Existen varias ventajas con respecto al uso de SPEM como lenguaje de modelado de procesos de IS. Primero, SPEM ha sido desarrollado por el Object Management Group (OMG), lo que le da el potencial de convertirse en un lenguaje estándar para el modelado de procesos de IS.

En segundo lugar, UML es el lenguaje de modelado más extendido en las organizaciones de desarrollo de software, lo que puede facilitar la asimilación de SPEM como lenguaje de modelado de procesos en este tipo de organizaciones. Esto es importante si consideramos que un estudio de procesos requiere de diversas iteraciones, lo que implica que se requiere retroalimentación de los participantes del proceso para ir mejorando los modelos en cada iteración [30]. Por lo tanto, usar un lenguaje con el que puedan estar familiarizados podría facilitar esta labor.

Finalmente, para modelar procesos con SPEM es posible usar cualquier herramienta que permita el uso de perfiles de UML [31]. Dado el extendido uso de UML, existen muchas herramientas disponibles con esta característica. Incluso, interfaces ampliamente usadas como plataformas de desarrollo de software, comienzan a proveer facilidades para modelar procesos con base en SPEM, tal es el caso del proyecto Eclipse Process Framework, (<http://www.eclipse.org/epf/>).

Sin embargo, SPEM no provee elementos para representar el conocimiento involucrado en los procesos. Con el fin de usarlo para estudiar flujos de conocimiento en procesos de IS, fue necesario adaptarlo añadiendo estos elementos como parte del lenguaje. A continuación se describen las adaptaciones que, siguiendo este enfoque, se hicieron a SPEM.

3. Adaptación de SPEM para estudiar flujos de conocimiento

La adaptación que se hizo de SPEM está basada en tres aspectos principales:

- (i) Ilustrar el conocimiento, y sus fuentes, involucrado (usado, creado, y/o modificado) en las actividades del proceso.
- (ii) Ilustrar flujos de conocimiento entre actividades, y cómo las fuentes que lo contienen son usadas o modificadas a través de éstas.
- (iii) Ilustrar las transferencias de conocimiento entre roles u otros tipos de fuentes.

Siguiendo estas tres premisas, se definieron una serie de conceptos que se agregaron como parte del metamodelo de SPEM, y se propuso un conjunto de elementos para complementar el lenguaje de modelado. Estos elementos se describen a continuación.

3.1 El conocimiento como producto del trabajo

SPEM basa su notación en tres elementos básicos: productos del trabajo (*work products*), definiciones de trabajo (*work definitions*), y roles (*roles*). Las definiciones de trabajo son operaciones que describen el trabajo realizado por los roles del proceso. Estas son usadas para estructurar el proceso, por ejemplo, describiendo su ciclo de vida, sus fases, iteraciones, o actividades. Los productos del trabajo es todo aquello usado o generado durante el proceso.

Con base en lo anterior, si consideramos al conocimiento como un recurso que es usado, generado o modificado durante las actividades del proceso, entonces podemos definirlo como un tipo de producto del trabajo. La figura 1 muestra los elementos principales que se agregaron a SPEM para permitir la representación del conocimiento en los modelos.

En particular, se definió un tipo de producto de trabajo para referirse al conocimiento involucrado en las actividades. Este conocimiento puede ser conceptos específicos, como temas o fuentes de conocimiento; o paquetes de conocimiento, que son grupos de temas de conocimiento relacionados, por ejemplo, pueden usarse para agrupar el conocimiento de un rol determinado, de un documento, etc. Otro uso que se puede dar a estos paquetes de conocimiento, es para agrupar temas específicos en temas más generales, en áreas, o en categorías, con el fin de usarlos para estructurar y clasificar el conocimiento, como se ejemplifica más adelante en este trabajo.

3.2 Relaciones

A la par de los conceptos para definir el conocimiento involucrado en las actividades, se definió un conjunto de relaciones para definir transferencias de conocimiento, así como el conocimiento que puede estar contenido en fuentes determinadas.

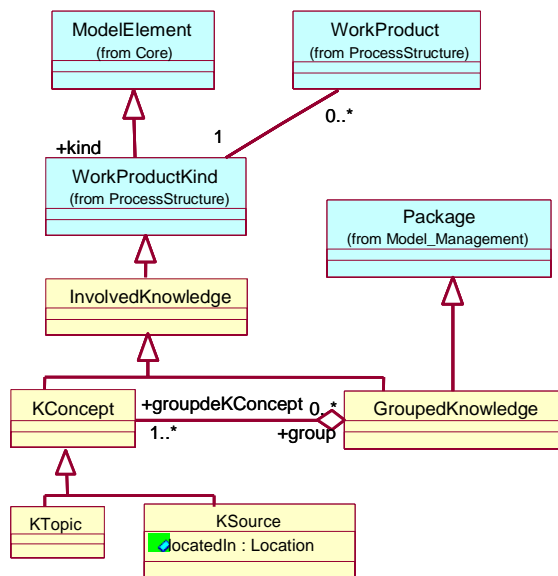


Fig. 1. Integración de los conceptos de conocimiento al metamodelo SPEM.

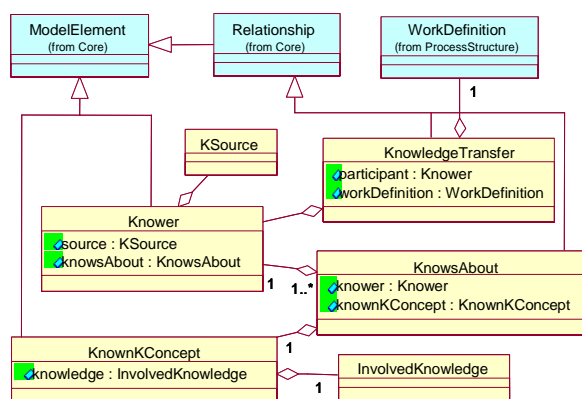


Fig. 2. Relaciones KnowledgeTransfer y KnowsAbout.

La figura 2 ilustra las relaciones que han sido definidas, y cómo estas se relacionan al metamodelo de SPEM. La relación *KnowledgeTransfer* es usada para representar transferencias de conocimiento entre fuentes, por ejemplo entre dos o más roles, entre un rol y un documento, etc. Esta relación tiene dos propiedades principales, las fuentes involucradas en la transferencia, y la definición de trabajo donde ésta se da. Los conocimientos transferidos u obtenidos por una fuente pueden ser temas específicos, o grupos de estos temas, y se especifican con una relación tipo *KnowsAbout*. La relación *KnowsAbout* define el conocimiento contenido en una determinada fuente.

3.3 Notación

Para integrar los elementos antes mencionados, en los diagramas generados con SPEM, se definieron una serie de iconos mostrados en la Figura 3. Estos iconos son usados en conjunto con los ya definidos por SPEM. Los temas de conocimiento o habilidades específicas son representados con elementos tipo *KTopic*. El elemento *GroupedKnowledge* es usado para representar grupos de temas relacionados, a su vez, es también útil para clasificar estos temas en paquetes. Finalmente, las transferencias de conocimiento son representadas por medio del icono *KnowledgeTransfer*.

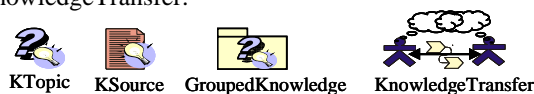


Fig. 3. Notación usada para representar los conceptos de conocimiento en diagramas de SPEM.

Con excepción del icono *KnowledgeTransfer*, el resto es usado en los diagramas predefinidos en SPEM. Para la representación de transferencias de conocimiento hemos definido diagramas adicionales. Para representar el proceso de manera general, así como las principales fuentes, o paquetes de conocimiento que intervienen, empleamos los casos de uso. Los diagramas de actividad nos ayudan a identificar detalles del conocimiento involucrado en actividades específicas, así como el conocimiento que cada rol puede requerir al desempeñar dichas actividades. Los diagramas de clases permiten representar el conocimiento que puede ser obtenido de cada fuente, así como dependencias y relaciones entre tipos de conocimiento y/o fuentes. Los diagramas de paquetes son usados para organizar y clasificar el conocimiento o habilidades en paquetes de conocimiento relacionados, por ejemplo, el conocimiento contenido en una fuente determinada. Finalmente, el análisis de transferencias de conocimiento es realizado por medio de diagramas de transferencia, que son un tipo de diagrama que se ha definido con base en la relación *KnowledgeTransfer*. Este tipo de diagramas permiten ilustrar las fuentes participando en la transferencia, la actividad o flujo de trabajo donde se da la transferencia, el conocimiento que es transferido, así como el conocimiento que cada fuente aporta, y el que obtienen al participar en dicha transferencia. Si se requiere especificar la transferencia con mayor detalle, por ejemplo, protocolos de discusión entre roles, o secuencia de llenado de contenido en un documento, etc., se hace uso de los diagramas de secuencia, o de diagramas de estados.

A continuación ilustramos el uso de los elementos de modelado por medio de un caso de estudio. En particular, el caso de estudio ejemplifica cómo el uso de los elementos de modelado puede ayudar a organizar y clasificar los temas de conocimiento involucrados en las actividades llevadas a cabo por el grupo estudiado.

4. Caso de estudio

Con el fin de aplicar estrategias de AC en una organización, uno de los primeros pasos es identificar las necesidades reales de conocimiento de las personas involucradas en los procesos de la misma [5, 6]. Esto puede llevarse a cabo por medio de una “auditoría de conocimiento” que nos ayude a identificar qué conocimiento es requerido por los participantes del proceso, qué fuentes de conocimiento existen, y qué conocimiento puede ser obtenido de dichas fuentes [32]. Para facilitar el análisis y manejo de esta información, una técnica muy útil es la definición de taxonomías de conocimientos y fuentes, las cuales nos permitirán clasificar estos elementos de una manera que se facilite la estructuración y generación de la base de conocimientos de la organización [32].

En el caso de estudio presentado en esta sección, se ejemplifica cómo el uso de los elementos de modelado descritos, permiten identificar el conocimiento y sus fuentes. A su vez se ilustra cómo desde el modelado de los procesos es posible comenzar a definir una clasificación de dichos elementos, con el fin de que ésta sirva para definir las taxonomías de conocimientos y fuentes que posteriormente puedan ser usadas para estructurar y generar la base de conocimientos del proceso.

Para este ejemplo se ha tomado información obtenida en un grupo dedicado al mantenimiento del software de un centro de investigación en México (ver [14]).

4.1 Diagramas de actividad y paquetes de conocimientos

La figura 4 muestra un ejemplo del uso de los diagramas de actividad para representar el conocimiento que es usado y generado en las actividades. El diagrama describe parte de las actividades que realiza un ingeniero de software durante el proceso de mantenimiento estudiado. Como se puede observar, el conocimiento que cada fuente

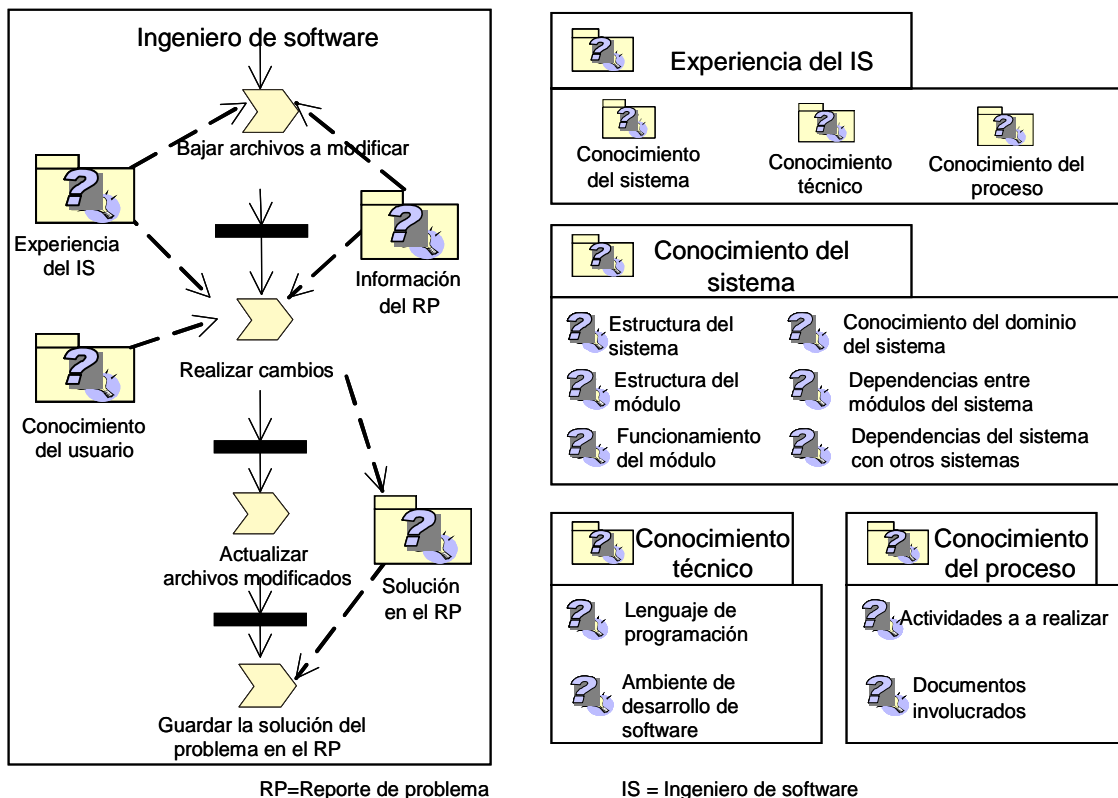


Fig. 4. Ejemplo de un diagrama de actividad que muestra el conocimiento involucrado en las actividades.

aporta u obtiene es agrupado en un paquete que representa el conocimiento de esa fuente. Posteriormente, estos paquetes de conocimiento son extendidos para definir el conocimiento específico que agrupan. Este conocimiento puede ser temas concretos, o paquetes que agrupan temas relacionados. De esta manera, al ir modelando el proceso es posible comenzar la clasificación del conocimiento involucrado, a la vez que se comienza la identificación de las fuentes que pueden contener el conocimiento, ya sea porque son las fuentes de donde es obtenido, o donde es almacenado.

A partir de modelos genéricos como el mostrado en la figura 4, se inició la identificación de temas y fuentes de conocimiento específicos. Como ejemplo veamos el caso del conocimiento sobre un sistema específico, digamos sistema A, que es un sistema para control de los alumnos inscritos en los posgrados que se imparten en el centro de investigación.

El conocimiento del dominio del sistema A, comprende los procesos escolares, tales como procesos de inscripciones, control de becas a estudiantes, etc. Este tipo de información está contenida en diversas fuentes, entre las que destacan los reglamentos y normas escolares, así como el personal de la dirección a cargo del control de estudiantes. Al identificar los distintos procesos a los que da soporte el sistema, se facilitó la identificación de las fuentes específicas que pueden servir para obtener información sobre dichos procesos. De esta manera se comenzó la estructuración y clasificación de dichas fuentes como parte de la base de conocimientos para apoyar el mantenimiento del sistema de control de estudiantes.

Por medio de este análisis, fue posible también identificar la estrecha relación que guardan las fuentes de conocimiento sobre el dominio del sistema A, con la evolución del mismo sistema. Esto se debe a que los

principales cambios en el sistema A se derivan de cambios en los reglamentos escolares, y es en base a estos reglamentos, y en la información proporcionada por el personal a cargo de los procesos escolares, que se definen los cambios que deben hacerse al sistema A, y la manera de hacerlos.

Con relación a la estructura del sistema A, se detectó un documento que describe las relaciones entre los distintos módulos del sistema y los archivos fuente que corresponden con dichos módulos. Esta fuente de conocimiento resulta de gran ayuda a la hora de hacer modificaciones al sistema, sobre todo si éstas requieren ser hechas en módulos del sistema que no son totalmente conocidos por la persona encargada de realizar los cambios.

Con respecto al funcionamiento de los distintos módulos del sistema A, se detectaron documentos que describen la manera en que deben operar. Dichos documentos han sido generados para apoyar a los usuarios, pero también resultaron ser una útil fuente de información para los encargados de modificar el sistema, dado que por medio de dichos documentos pueden darse una idea de cómo debe funcionar cada módulo. Las relaciones entre fuentes y el conocimiento que puede ser obtenido de ellas se modeló por medio de diagramas de clases, como se describe a continuación.

4.2 Diagramas de clases: relaciones entre el conocimiento y sus fuentes

La figura 5 presenta un ejemplo del uso de diagramas de clases para describir las relaciones entre fuentes y conocimiento. Este tipo de diagrama es usado para describir el conocimiento que cada fuente contiene, la forma en que están agrupadas las fuentes,

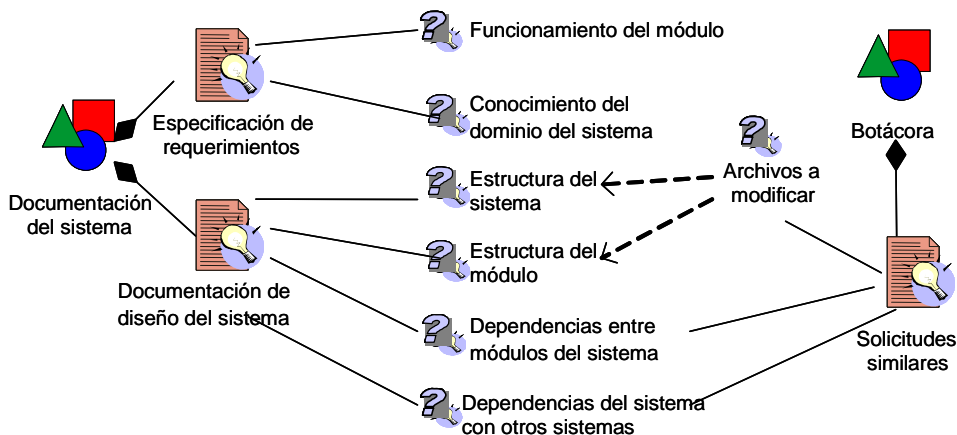


Fig. 5. Ejemplo de un diagrama de clases que ilustra el conocimiento que puede ser obtenido de cada fuente.

así como relaciones y dependencias entre tipos de conocimiento. Por ejemplo, en la figura 5 se muestra que las solicitudes de mantenimiento similares a la que se está atendiendo pueden ser usadas para obtener información sobre los archivos que requerirán ser modificados. A su vez, para esto también es necesario conocer la estructura del sistema y del módulo específico que se requiere modificar, información que puede ser obtenida del documento de diseño del sistema.

Por medio de las extensiones propuestas a SPEM, también es posible crear diagramas que se enfocan en identificar transferencias de conocimientos entre actividades, roles y otro tipo de fuentes. El objetivo del modelado de estos flujos de conocimiento es identificar los problemas que pudieran estar afectando que el conocimiento fluya adecuadamente a través del proceso. Ejemplos de lo anterior son la identificación de conocimiento que es generado en ciertas actividades y no es guardado, aun cuando sea importante para realizar otras actividades. Otro caso es la identificación de fuentes importantes con respecto al conocimiento que contienen, y que no están siendo usadas adecuadamente durante el proceso.

Por ejemplo, en el grupo estudiado identificamos que los requerimientos relacionados con las solicitudes de modificaciones a los sistemas, no están siendo capturados adecuadamente, ya que existe un mecanismo estándar para esto. De manera que se dificulta saber con posterioridad qué fue lo que generó un cambio determinado.

La identificación de este tipo de situaciones es importante debido a que pueden contribuir a proponer alternativas para mejorar los flujos de conocimiento durante el proceso. Si consideramos que los procesos de desarrollo y mantenimiento de software utilizan gran cantidad de conocimiento, entonces lograr esquemas que permitan que el conocimiento fluya de forma más rápida y eficiente, puede contribuir en gran medida a la mejora de dichos procesos.

5. Lecciones aprendidas

Las siguientes son dos de las principales lecciones aprendidas en la realización de este trabajo:

Identificación de problemas relacionados con el conocimiento. La aplicación de SPEM, y en particular el uso de las extensiones definidas para representar explícitamente el conocimiento involucrado en los procesos modelados, permitieron identificar problemas en los flujos de conocimiento que actualmente se dan dentro del proceso. Por ejemplo, se identificó información y conocimiento que no estaba siendo almacenado por el grupo, aun cuando resultaba de gran

relevancia para otras actividades, o para otros roles dentro del proceso. Esto fue posible gracias a la representación explícita del conocimiento y fuentes involucradas en los procesos. De no haber contado con la representación explícita del conocimiento en los diagramas de SPEM, parte de estos problemas difícilmente hubieran sido identificados.

Estructuración y clasificación de temas y fuentes de conocimientos. La posibilidad de representar, tanto temas específicos como conjuntos de los mismos, facilitó la definición de una estructura de clasificación de conocimientos y fuentes, lo cual ayudó a estructurar un mapa de conocimientos del proceso estudiado [33].

Los dos puntos mencionados son importantes debido a que antes de realizar la mejora de cualquier proceso, es necesario identificar las debilidades del mismo. Por otra parte, un primer paso en la definición de sistemas de AC, es definir taxonomías que permitan estructurar, clasificar y organizar el conocimiento de la organización y las fuentes que lo contienen [32].

Como resultado de este trabajo, el grupo donde se realizó el estudio actualmente se encuentra tomando medidas para resolver parte de los problemas observados, con el fin de mejorar su proceso de desarrollo y mantenimiento de software.

6. Conclusiones y comentarios finales

Con base en nuestra experiencia en este trabajo, podemos constatar lo que otros autores han afirmado con respecto a la necesidad de proveer lenguajes de modelado de procesos que estén orientados a representar el conocimiento involucrado en los procesos [13, 18]. Consideramos que SPEM es una buena alternativa como lenguaje para analizar los procesos existentes en grupos de IS. Debido a la alta dependencia del conocimiento que existe en este tipo de procesos, un lenguaje de modelado que considere el conocimiento involucrado podría ser de gran ayuda.

Ya que SPEM no cuenta con primitivas para representar explícitamente el conocimiento involucrado en procesos de IS, vimos la necesidad de dotarlo con esta alternativa con el fin de que sirva para analizar y definir procesos de IS considerando el conocimiento involucrado en las actividades que constituyen a los procesos. El trabajo presentado en este artículo es un esfuerzo en este sentido. La experiencia obtenida al aplicarlo a un caso real, nos ha dado evidencia de que puede ser de gran ayuda para apoyar en la definición de estrategias de AC en grupos encargados del desarrollo y mantenimiento de software.

En particular, el uso de las extensiones definidas para representar el conocimiento involucrado en el

proceso, fue de gran ayuda para identificar problemas relacionados con el conocimiento, y de esta manera comenzar a tomar medidas para abordarlos. Así también, el uso de las extensiones permitió definir modelos generales del proceso de mantenimiento del grupo estudiado junto con las principales áreas o temas generales de conocimiento involucrados. Esto facilitó la identificación de temas o áreas de conocimiento específicos relacionados con cada uno de los sistemas que son mantenidos por el grupo. Por ejemplo, los temas de conocimiento relacionados con el dominio de cada uno de dichos sistemas.

Como trabajo futuro, buscaremos integrar las adaptaciones hechas a SPEM en un ambiente de modelado de procesos. De forma tal que los modelos sirvan también para definir la estructura de la base de conocimientos del proceso. Así mismo, seguiremos aplicando el enfoque de modelado para estudiar otros procesos y otros grupos de desarrollo, con el fin de continuar nuestra evaluación de las adaptaciones a SPEM que han sido definidas, y extenderlas o mejorarlas en caso de ser necesario.

7. Agradecimientos

Este trabajo fue apoyado en parte por CONACYT bajo el proyecto C01-40799 y la beca 164739 proporcionada al primer autor, en México; y por los proyectos MECENAS (PBI06-0024) y ENIGMAS (PIB-05-058), de la Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, en España. Este trabajo también forma parte de la red CALIPSO (TIN2005-24055-E), apoyada por el Ministerio de Ciencia y Tecnología en España.

8. Referencias

- [1] A. Aurum, R. Jeffery, C. Wohlin, and M. Handzic, "Managing Software Engineering Knowledge." Springer, Berlin, Germany, 2003.
- [2] T. Dingsøyr and R. Conradi, "A survey of case studies of the use of knowledge management in software engineering," *International Journal of Software Engineering and Knowledge Engineering*, vol. 12, num. 4, 2002, pp. 391-414.
- [3] K. C. Desouza, "Barriers to Effective Use of Knowledge Management Systems in Software Engineering," *Comm. of the ACM*, 46(1), 2003, pp. 99-101.
- [4] T. A. Stewart, "The Case Against Knowledge Management," *Business 2.0*, vol. 3, February, 2002, pp. 80.
- [5] K. Wiig, *People-Focused Knowledge Management: How Effective Decision Making Leads to Corporate Success*. Elsevier, Amsterdam, 2004.
- [6] R. Maier and U. Remus, "Defining Process-oriented Knowledge Management Strategies," *Knowledge and Process Management*, vol. 9, num. 2, 2002, pp. 103-118.
- [7] B. Curtis, M. I. Kellner, and J. Over, "Process Modeling," *Comm. of the ACM*, vol. 35, num. 4, 1992, pp. 75-90.
- [8] "Software Process Engineering Metamodel Specification (SPEM)," vol. 2004. Object Management Group, 2002.
- [9] K. Dalkir, *Knowledge Management in Theory and Practice*. Elsevier, Amsterdam, 2005.
- [10] U. M. Borghoff and R. Pareschi, "Information Technology for Knowledge Management," *Journal of Universal Computer Science*, vol. 3, num. 8, 1997, pp. 835-842.
- [11] M. E. Nissen, "An Extended Model of Knowledge-Flow Dynamics," *Communications of the Association for Information Systems*, vol. 8, 2002, pp. 251-266.
- [12] B. H. Hansen and K. Kautz, "Knowledge Mapping: A Technique for Identifying Knowledge Flows in Software Organizations", *Proc. of the European Conference on Software Process Improvement (EuroSPI 2004)*, Trondheim, Norway, 2004, pp. 126-137.
- [13] P. Bera, D. Nevo, and Y. Wand, "Unraveling Knowledge Requirements through Business Process Analysis," *Comm. of the AIS*, vol. 16, 2005, pp. 814-830.
- [14] O. M. Rodríguez, A. I. Martínez, A. Vizcaíno, J. Favela, and M. Piattini, "Identifying Knowledge Management Needs in Software Maintenance Groups: A qualitative approach", *Proc. of the Fifth Mexican International Conference on Computer Science (ENC'2004)*, Colima, México, 2004, pp. 72-79.
- [15] M. S. Abdullah, I. Benest, A. Evans, and C. Kimble, "Knowledge Modelling Techniques for Developing Knowledge Management Systems", *Proc. of the European Conference on Knowledge Management*, Dublin, Ireland, 2002, pp. 15-25.
- [16] T. H. Davenport and L. Prusak, *Working Knowledge: How Organizations Manage What they Know*. Harvard Business School Press, Boston, Massachusetts, USA, 2000.
- [17] R. Conradi and L. Jaccheri, "Process Modelling Languages," *LNCS 1500*, Springer, Berlin, 1999, pp. 27-52.
- [18] G. Papavassiliou and G. Mentzas, "Knowledge modelling in weakly-structured business processes," *Journal of Knowledge Management*, vol. 7, num. 2, 2003, pp. 18-33.
- [19] S. Kim, H. Hwang, and E. Suh, "A Process-based Approach to Knowledge Flow Analysis: A Case Study of a manufacturing Firm," *Knowledge and Process Management*, vol. 10, num. 4, 2003, pp. 260-276.
- [20] M. Strohmaier and K. Tochtermann, "B-KIDE: A Framework and a Tool for Business Process-Oriented Knowledge Infrastructure Development," *Journal of Knowledge and Process Management*, vol. 12, num. 3, 2005, pp. 171-189.

- [21] R. Woitsch and D. Karagiannis, "Process-oriented Knowledge Management Systems based on KM-Services: The PROMOTE Approach," *International Journal of Intelligent Systems in Accounting, Finance & Management*, vol. 11, 2002, pp. 253-267.
- [22] T. C. Lethbridge, "What Knowledge Is Important to a Software Professional?" *IEEE Computer*, vol. 33, num. 5, 2000, pp. 44-50.
- [23] P. N. Robillard, "The Role of Knowledge in Software Development," *Comm. of the ACM*, vol. 42, num. 1, 1999, pp. 87-92.
- [24] K. M. Oliveira, N. Anquetil, D. M.G, M. Ramal, and R. Meneses, "Knowledge for Software Maintenance." *Proc. of the Fifteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'03)*, San Francisco, CA, 2003, pp. 61-68.
- [25] M. G. B. Dias, N. Anquetil, and K. M. d. Oliveira, "Organizing the Knowledge Used in Software Maintenance," *Journal of Universal Computer Science*, vol. 9, num. 7, 2003, pp. 641-658.
- [26] A. Monk and S. Howard, "The Rich Picture: A Tool for Reasoning About Work Context," *Interactions*, vol. 5, num. 2, 1998, pp. 21-30.
- [27] P. Checkland and J. Scholes, *Soft System Methodology in Action*. John Wiley and Sons, 1999.
- [28] M. E. Nissen and R. E. Levitt, "Agent-Based Modeling of Knowledge Flows: Illustration from the Domain of Information Systems Design", *Proc. of the Hawaii International Conference on System Science (HICSS 2004)*, Big Island, Hi, USA, 2004, pp.
- [29] H. Zhuge, "Knowledge flow management for distributed team software development," *Knowledge-Based Systems*, vol. 15, num. 8, 2002, pp. 465-471.
- [30] D. G. Wastell, P. White, and P. Kawalek, "A Methodology for Business Process Redesign: Experiences and Issues," *Journal of Strategic Information Systems*, vol. 3, num. 1, 1994, pp. 23-40.
- [31] J. Bézivin and E. Breton, "Applying the Basic Principles of Model Engineering to the Field of Process Engineering," *UPGRADE*, vol. V, num. 5, 2004, pp. 27-33.
- [32] M. Rao, "Knowledge Management Tools and Techniques: Practitioners and Experts Evaluate KM Solutions." Elsevier, Amsterdam, 2005, pp. 438.
- [33] O. M. Rodríguez-Elias, A. I. Martínez-García, A. Vizcaíno, J. Favela, and M. Piattini, "Constructing a Knowledge Map for a Software Maintenance Organization", *Proc. of the Poster Session of the 21st IEEE International Conference on Software Maintenance (ICSM 2005)*, Budapest, Hungary, 2005, pp. 51-54.
- [34] J. P. Soto, O. M. Rodríguez, A. Vizcaíno, M. Piattini, and A. I. Martínez-García, "Localización de fuentes del conocimiento en el proceso del mantenimiento del software," *Memorias del Campus Multidisciplinar en Percepción e Inteligencia (CMPI-2006)*, vol. I. Universidad de Castilla-La Mancha, Albacete, España, 2006, pp. 118-123.