

Argentina



Apoya



XII Conferencia
Iberoamericana
de Ingeniería de Requisitos
y Ambientes de Software



Co
a
del
amb

IDEAS

2009

IDEAS 2009

Memorias

XII Conferencia Iberoamericana de Ingeniería de Requisitos y Ambientes de Software

Editores

Antonio Brogi, João Araújo, Raquel Anaya.

Medellín, Colombia
Abril 13 - 17, 2009

Memorias

**XIII Conferencia Iberoamericana de
Ingeniería de Requisitos
y Ambientes de Software**

.....
PRESIDENCIA DEL COMITÉ ORGANIZADOR
.....

Raquel Anaya
Universidad EAFIT, Colombia

.....
PRESIDENCIA DEL COMITÉ DE PROGRAMA
.....

João Araújo
Universidade Nova de Lisboa, Portugal

Antonio Brogi
Università di Pisa, Italy

Ficha Técnica
Memorias de la XII Conferencia Iberoamericana de Ingeniería
de Requisitos y Ambientes de Software (IDEAS '09)
Editores: Antonio Brogi, João Araújo, Raquel Anaya.
Abril, 2009 - Medellín, Colombia

Copyright © 2009 by IDEAS '09
Prohibida la reproducción total o parcial de esta obra,
por cualquier medio, sin la autorización de sus editores.

ISBN: 978-958-44-5028-9

MIEMBROS DEL COMITÉ CIENTÍFICO

Alejandra Cechich
Alessandro García
Álvaro Arenas
Amador Duran
Antonio Brogi
Antonio Vallecillo
Carla Reis
Carla Silva
Claudia Pons
César Acuña
Coral Calero
Dan Hirsch
Daniel Riesco
Daniela Godoy
Demetrio Ovalle
Elena Navarro
Ernest Teniente
Ernesto Pimentel
Fernanda Alencar
Francisco Pinheiro
Francisco Ruíz
Gaston Mousques
Geneveva Vargas
Guilherme Travassos
Gustavo Rossi
Hernan Melgratti
Isabel Díaz
Isabel Brito
Jaelson Castro
Jaime Muñoz
Jesús García Molina

João Araújo
João Falcão e Cunha
Jonás Montilva
Jorge Trifianes
José Pow-Sang
José Maldonado
Juan Carlos Trujillo
Juan Hernández
Júlio Leite
Luca Cernuzzi
Luis Guerrero
Luis Olsina
Lyrene Silva
Marcello Visconti
Márcio Delamaro
Márcio Barros
María Lencastre
Miguel Katrib
Oscar Dieste
Oscar Pastor
Rafael Calvo
Raquel Anaya
Regina Braga
Renata Guizzardi
Ricardo Falbo
Ruby Casallas
Sandra Fabbri
Silvia Gordillo
Vicente Pelechano
Victor Santander
Xavier Franch

ORGANIZACIÓN LOCAL

Alberto Restrepo
Mónica Henao
Lucas Macías Franco
Isabel Morales

PREFACIO

Bienvenidos a la décimo segunda versión de la Conferencia de Ingeniería de Requisitos y Ambientes Software (IDEAS 2009) que va ser realizado en Medellín Colombia y es organizado por el Departamento de Informática y Sistemas de la Escuela de Ingeniería de la Universidad EAFIT, del 13 al 17 de abril del 2009.

Desde su primera edición en 1.998, IDEAS fue concebido como un espacio para estimular y facilitar el intercambio de conocimiento y experiencias y para orientar las relaciones entre grupos de investigación iberoamericanos que trabajan en diversas áreas de la Ingeniería de Software. IDEAS provee un foro que permite que investigadores, educadores y profesionales presenten y discutan los desarrollos más recientes en ingeniería de software.

El primer evento de IDEAS fue realizado en 1.998 en Torres, Brasil, como un workshop. Desde entonces, el evento se ha realizado de manera exitosa en diversos países de Latinoamérica: San José-Costa Rica (IDEAS'99), Cancún-México (IDEAS'00), Heredia-Costa Rica (IDEAS'01), La Habana-Cuba (IDEAS'02), Asunción-Paraguay (IDEAS'03), Arequipa-Perú (IDEAS'04), Valparaíso-Chile (IDEAS'05), La Plata-Argentina (IDEAS'06), Isla de Margarita-Venezuela (IDEAS'07), Recife-Brasil (IDEAS'08), y Medellín-Colombia (IDEAS'09). Vale la pena destacar que este año se aprueba oficialmente el cambio de nombre del evento de Workshop a Conferencia, teniendo en cuenta su evolución en número de trabajos presentados y participantes inscritos.

La agenda académica de IDEAS'09 cuenta con tres conferencias plenarias, dos meses redondas, cuatro tutoriales y la presentación de los trabajos aceptados. Los tres conferencistas invitados son Jorge Villalobos (Universidad de los Andes, Colombia) quien discutirá las tendencias recientes y retos en el desarrollo de arquitecturas orientadas a servicios, Guilherme Travassos (Universidad Federal de Río de Janeiro, Brasil) quien presentará el estado de la ingeniería de software experimental, y Ernesto Pimentel (Universidad de Málaga, España) quien analizará la aplicación de los métodos formales para coordinar y adaptar servicios y componentes. Los dos paneles estarán orientados a abrir espacios de discusión alrededor de las iniciativas de la academia para responder a las demandas del mercado laboral y el papel de la industria de software latinoamericana en el mercado mundial, respectivamente. La conferencia estará precedida por dos días de tutoriales que estarán orientados a los temas de la ingeniería de requisitos orientada a aspectos, proyectos de desarrollo centrados en la arquitectura, modelado de sistemas multi-agente y calidad de evaluación de aplicaciones Web 2.0, respectivamente.

IDEAS siempre ha recibido artículos en español, portugués e inglés. Para esta edición hemos recibido un total de 82 trabajos de 18 países distintos. Cada trabajo

fue revisado por al menos tres miembros del Comité de Programa. Después de un riguroso proceso de revisión, fueron aceptados 19 artículos completos y 19 artículos cortos.

El trabajo del Comité de Programa y de los revisores adicionales que colaboraron en el proceso de evaluación de artículos es sobresaliente. Todos los autores recibieron comentarios detallados de los evaluadores. Agradecemos a todos los revisores por su excelente trabajo y agradecemos también a todos los autores que enviaron sus trabajos a la conferencia. Agradecemos a la Universidad EAFIT por el patrocinio de IDEAS'09, así mismo al Comité de Organización local que hizo posible la realización de esta conferencia.

Finalmente, extendemos una cordial bienvenida a conferencistas, autores, estudiantes y profesionales que nos acompañaran en IDEAS'09. Esperamos que puedan disfrutar del evento y además tengan la oportunidad de disfrutar de la cultura de Medellín y de la amabilidad de su gente.

Antonio Brogi
João Araújo
Raquel Anaya

Abril 2009

PREFÁCIO

Bem-vindos à 12ª Conferência Ibero-americana em Engenharia de Requisitos e Ambientes de Software (IDEAS 2009) que tem lugar em Medellín, Colômbia, organizada pelo Departamento de Informática e Sistemas, Escola de Engenharia da Universidade EAFIT, de 13 a 17 de Abril de 2009.

Desde a sua primeira edição em 1998, IDEAS foi concebida para estimular e facilitar o intercâmbio de conhecimento e de experiências, além de estreitar as relações entre grupos de pesquisa ibero-americanos trabalhando em diversas áreas da Engenharia de Software. IDEAS proporciona um fórum que tem como objetivo permitir que investigadores, educadores e profissionais apresentem e discutam os mais recentes desenvolvimentos em Engenharia de Software.

O primeiro evento de IDEAS teve lugar em 1998 em Torres, Brasil, como um workshop. Desde então, o evento foi realizado com sucesso em San Jose-Costa Rica (IDEAS'99), Cancun-México (IDEAS'00), Herédia-Costa Rica (IDEAS'01), La Habana-Cuba (IDEAS'02), Asuncion-Paraguai (IDEAS'03), Arequipa-Peru (IDEAS'04), Valparaiso-Chile (IDEAS'05), La Plata-Argentina (IDEAS'06), Isla de Margarita-Venezuela (IDEAS'07), Recife-Brasil (IDEAS'08), e Medellín, Colômbia (IDEAS'09). Vale a pena salientar que este ano o Workshop evoluiu para Conferência, uma vez que o seu tamanho, em termos de submissões e participantes, justifica esta promoção.

IDEAS'09 inclui três palestras convidadas, dois painéis de discussão, quarto tutoriais e as apresentações dos artigos. Os três palestrantes convidados são Jorge Villalobos (Universidade de Los Andes, Colômbia) que discutirá as tendências em arquiteturas orientadas a serviços, Guilherme Travassos (Universidade Federal do Rio de Janeiro, Brasil) que discutirá os desafios em engenharia de software experimental, e Ernesto Pimentel (Universidade de Málaga, Espanha) que discutirá a aplicação de métodos formais para coordenar e adaptar serviços e componentes. Os dois painéis serão voltados ao análise das iniciativas do mundo acadêmico para responder às demandas do mercado de trabalho, e discutir o papel da indústria de software latino-americana no mercado mundial, respectivamente. A conferência será precedida por dois dias de tutoriais, que enfatizarão engenharia de requisitos orientada a aspectos, projetos orientados a arquiteturas, modelação de sistemas multi-agentes, e avaliação de qualidade de aplicações Web 2.0.

IDEAS sempre recebeu artigos em espanhol, português ou inglês, a fim de fomentar a interação entre pesquisadores de diferentes países ibero-americanos. Para esta edição de IDEAS, recebemos um total de 82 submissões de 18 países distintos. Cada artigo foi revisado por pelo menos três membros do Comitê de Programa. Depois de um processo de avaliação rigoroso, 19 artigos foram aceites como artigos completos

e 19 foram aceites como artigos curtos.

O trabalho do Comité de Programa e dos outros avaliadores que colaboraram no processo de avaliação foi de altíssimo nível. Todos os autores receberam comentários detalhados dos avaliadores. Nós gostaríamos de agradecer a todos os avaliadores pelo seu excelente trabalho. Gostaríamos também de agradecer a todos os autores por submeter as suas valiosas contribuições.

Agradecemos a Universidade EAFIT pelo patrocínio de IDEAS'09, assim como o Comité de Organização local que tornou esta conferência possível.

Finalmente, desejamos que todos os conferencistas, autores, estudantes e profissionais que nos acompanharão em IDEAS'09 sejam muito bem-vindos. Esperamos que possam desfrutar do evento e que tenham a oportunidade de conhecer a cultura de Medellín e da amabilidade de sua gente.

Antonio Brogi
João Araújo
Raquel Anaya

Abril 2009

PREFACE

Welcome to the 12th Ibero-american Conference on Requirements Engineering and Software Environments (IDEAS 2009) to be held in Medellín, Colombia, which is organised by the Department of Informatics and Systems, of Eafit University's Engineering School, from April 13 to 17, 2009.

Since its first edition in 1998, IDEAS was conceived as a space to stimulate and facilitate the exchange of knowledge and experiences, and to direct the relations among Ibero-american research groups working in diverse areas of Software Engineering. IDEAS provides a forum that allows that researchers, educators and professionals present and discuss the most recent developments in software engineering.

The first IDEAS event was held in 1998 in Torres, Brazil, as a workshop. Since then, the event has successfully taken place in San Jose - Costa Rica (IDEAS-99), Cancun-Mexico (IDEAS -00), Heredia - Costa Rica (IDEAS-01), La Habana-Cuba (IDEAS-02), Asuncion-Paraguay (IDEAS-03), Arequipa-Peru (IDEAS-04), Valparaiso-Chile (IDEAS-05), La Plata-Argentina (IDEAS-06), Isla Margarita-Venezuela (IDEAS-07), Recife-Brazil (IDEAS-08), and Medellín, Colombia (IDEAS-09). It is worth pointing out that this year a change of the name of the event is going to be approved from "Workshop" to "Conference", having into count its evolution in terms of the number of papers presented, and the number of signed participants.

IDEAS-09 features three plenary sessions, two panels, four tutorials, and the presentations of contributed papers. The three invited speakers are Jorge Villalobos (Los Andes University, Colombia) who will discuss recent trends and challenges in developing service-oriented architectures; Guilherme Travassos (Federal University of Rio de Janeiro, Brazil) who will talk about the current state of experimental software engineering; and Ernesto Pimentel (University of Malaga, Spain) who will analyze the application of formal methods to coordinate and adapt services and components. The two panels will be devoted to open discussion spaces around the initiatives of the academic world to respond to the demand from the labour market, and the role of the Latin-American software industry in the world market, respectively. The conference will be preceded by two days of tutorials, which will be oriented towards requirements engineering topics, particularly to architecture-driven projects, multi-agent systems modelling, and quality evaluation of web 2.0 applications.

IDEAS has always welcomed articles in Portuguese, Spanish, and English. For this edition of IDEAS, we received a total of 82 submissions from 18 different countries. Each paper was reviewed by at least three members of the Program Committee. After a rigorous reviewing process, 19 papers were accepted as full papers and 19 were accepted as short papers.

The work of the Program Committee and that of the extra reviewers who collaborated in the paper evaluation process was outstanding. All authors received detailed comments from the referees. We would like to thank all reviewers for their great job. We would also like to thank all authors for submitting valuable contributions. We would like to thank EAFIT University for sponsoring IDEAS-09, as well as the local Organizing Committee that made it possible to run this conference.

Finally, we cordially welcome all the conference participants, authors, students and professionals that will join IDEAS-09. We do hope that you will enjoy the event and will also have the chance to experience the culture of Medellín and its people's hospitality.

Antonio Brogi
 João Aratijo
 Raquel Anaya

April 2009

TABLA DE CONTENIDO

CHARLAS INVITADAS

Tendencias y retos en el diseño de arquitecturas orientadas a servicios.
Jorge Villalobos

Ingeniería de Software Experimental: Logros y perspectivas
Guilherme Travassos

Integración de software: métodos formales para coordinar y adaptar componentes y servicios
Ernesto Pimentel

SESIÓN 1. MODELADO DEL NEGOCIO

Desarrollo de software orientado a servicios..... 1
Andrea Delgado, Ignacio García, Francisco Ruiz.

Modelado de Negocio Interorganizacional: Una Aproximación para la Trazabilidad entre Objetivos, Modelos Organizacionales y Procesos de Negocio.
José Bocanegra, Joaquín Peña, Antonio Ruiz-Cortés. 15

LIS2BP: Una propuesta para obtener Procesos de Negocio a partir de los Sistemas Heredados.
Alfonso Rodríguez, Angélica Caro. 29

Modelado de Requisitos de Datos para Sistemas de Información basados en Procesos de Negocio.
José Luis de la Vara, Michel H. Fortuna, Juan Sánchez, Cláudia M. L. Werner, Marcos R. S. Borges. 43

SESIÓN 2. DESARROLLO DIRIGIDO POR MODELOS

Product Derivation in a Model-Driven Software Product Line using Decision Models.
Hugo Arboleda, Andrés Romero, Rubby Casallas, Jean-Claude Royer. 57

A two-level formal semantics for the QVT language
Roxana Giandini, Claudia Pons, Gabriela Pérez. 73

Extending Visual Modeling Languages with Timed Behavior Specifications. <i>Jose E. Rivera, Cristina Vicente-Chicote, Antonio Vallecillo.</i>	87	Aplicación del marco metodológico de COMPETISOFT a través de Investigación-Acción y Casos de estudio. <i>Francisco J. Pino, Félix García, Mario Piattini.</i>	167
SESIÓN 3. DESARROLLO DIRIGIDO POR MODELOS (SHORT PAPERS)			
Integración de UML y DSMLs en Entornos de Desarrollo Dirigido por Modelos. <i>Giovanni Giachetti, Beatriz Marín, Oscar Pastor López.</i>	101	SMML: Lenguaje para la Representación de Modelos de Medición del Software. <i>Beatriz Mora, Félix García, Francisco Ruiz, Mario Piattini.</i>	181
Identificación de Defectos en Modelos Conceptuales utilizados en Entornos MDA. <i>Beatriz Marín, Giovanni Giachetti, Oscar Pastor López, Alain Abran.</i>	109	Modelado de Líneas de Procesos mediante SPEM v2.0 (Presentado en sesión 5). <i>Tomas Martínez-Ruiz, Félix García, Mario Piattini.</i>	195
SESIÓN 5. ASPECTOS Y REQUISITOS			
A Service-Oriented Approach for Model Management. <i>Jorge Pérez Medina, Dominique Rieu, Sophie Dupuy-Chessa.</i>	115	Constructing Measurement Repositories in Software Organizations: a real experience. <i>Solange Araujo, Adriano Albuquerque, Arnaldo Belchior, Nabor Mendonça.</i>	209
Uso de Modelos de Anotación para Automatizar el Desarrollo Dirigido por Modelos de Esquemas XML. <i>Ferónica Andrea Bollati, Juan Manuel Vara, Belén Vela, Esperanza Marcos.</i>	121	An Aspect-Oriented Framework for Software Documentation: An Example on Testing. <i>Elisa Y. Nakagawa, Mariela M. F. Sasaki, Jose C. Maldonado.</i>	225
Estrategias para la Definición de una Técnica de Modelado para Arquitecturas de Referencia. <i>Javier Pérez, Juan Bernardo Quintero.</i>	127	Una Ontología de Aspectos para la Ingeniería de Requerimientos. <i>Gladis Errecalde, Claudia Marcos.</i>	239
La influencia de ODM sobre la colaboración entre la Arquitectura Dirigida por Modelos y las Ontologías. <i>Diana Marcela Sánchez Fiquene, José María Cervero, Esperanza Marcos.</i>	133	Derivación de casos de uso con aspectos a partir de modelos organizacionales i*. <i>Karin Andrea Lizana Rojas, Víctor Araya Santander, Fernanda Alencar, Jaelson Castro, Juan Sánchez Díaz.</i>	253
SESIÓN 6. MEJORA DEL PROCESO SOFTWARE (SHORT PAPERS)			
A Domain Specific Language to Generate Web Applications. <i>Juan José Cadavid, Juan Bernardo Quintero, David Esteban Lopez, Jesus Andrés Hincapié.</i>	139	Integrando Proceso y Marco de Medición y Evaluación. <i>Pablo Becker, Hernan Molina and Luis Olsina.</i>	259
Achieving Consistency and Completeness of Business Process Models throughout the Lifecycle. <i>Marta S. Tabares, Fernando Arango.</i>	145	Apoyo Automatizado à Elaboração de Planos de Gerência de Conhecimento para Processos de Software. <i>Jadelly Oliveira and Carla Reis.</i>	267
SESIÓN 4. MEJORA DEL PROCESO SOFTWARE			
Homogenización de marcos en ambientes de mejora de procesos multimarco. <i>César Jesús Pardo Cabvache, Francisco J. Pino, Félix García, Mario Piattini.</i>	151	Estado del Arte de las Pruebas en Líneas de Producto Software. <i>Beatriz Pérez Lamancha, Macario Polo Usaola and Mario Piattini Velthuis.</i>	273
		Um Estudo dos Critérios para Adoção de Metodologias Ágeis. <i>Cleviton Monteiro, Daniel F. Arcoverde, Raoni O. S. Franco and Fabio Q. B. da Silva.</i>	279

Disfunção dos Sistemas de Medição em Organizações de Software..... 285
Gibeon Aquino, Felipe Furtado, Renata Alchorne, Suzana Sampato and Silvio Meira.

MPS.BR – A Experiência de Um Gap Analysis nos Processos..... 291
de Verificação e Validação de uma Organização Brasileira.
Adriano Albuquerque and Lauro Oliveira Neto.

Performance Models to Predict the Productivity..... 297
of Projects: a Practical Application.
Carla Bezerra, Ciro Coelho, Giovano Pires and Adriano Albuquerque.

Utilização de Práticas Genéricas do CMMI para..... 303
apoiar a utilização de Metodologias Ágeis.
Célio Santana, Cristine Gusmão, Ana Rouiller and Alexandre Vasconcelos.

SESIÓN 7. CALIDAD Y COMPONENTES

Análisis de Desajustes Respecto los Requisitos..... 309
en la Selección de Componentes OTS.
Juan Pablo Carvallo and Xavier Franch.

Gestión Sistemática de la Calidad de la Información en los..... 325
Procesos de Selección de Componentes de Software.
Claudia Ayala and Xavier Franch.

SPL-OOWS: Uma extensão do método OOWS..... 339
utilizando linha de produto de software.
Bruno Miguel Nogueira de Souza, Itana M. S. Gimenes and Thelma Elita Colanzi

An Embedded software component..... 353
Quality Maturity Model (EQM2)
Fernando Carvalho, Silvio Meira and Jefferson Silveira.

SESIÓN 8. APLICACIONES

VisAr3D: Uma abordagem baseada em Realidade..... 359
Aumentada para o Ensino de Arquitetura de Software.
Claudia S. Rodrigues and Cláudia M. L. Werner.

Enfoque Integrado para el Procesamiento de..... 374
Flujos de Datos: Un Escenario de Uso.
Mario José Diván and Luis Olsina.

Reutilización de Casos de Uso en el Desarrollo de..... 388
Sistemas Grid seguros.
David G. Rosado, Eduardo Fernandez-Medina and Javier López.

CHARLAS INVITADAS

Modelado de Líneas de Procesos mediante SPEM v2.0

Tomás Martínez-Ruiz, Félix García, and Mario Piattini

Grupo de Investigación Alarcos, Departamento de Tecnologías y Sistemas de Información, Escuela Superior de Informática, Universidad de Castilla-La Mancha Paseo de la Universidad, 4, 13071 Ciudad Real, España
{tomas.martinez,felix.garcia,mario.piattini}@uclm.es

8. Feilkas, M.; *How to represent Models, Languages and Transformations?* Proceedings of the 6th OOPSLA Workshop on Domain-Specific Modeling (DSM'06). (2006). pp.204-213.
9. Fenton, N. y Pfeleger, S. L.; *Software Metrics: A Rigorous & Practical Approach, Second Edition*, (1997).
10. García, F., Bertoa, M. F., Calero, C., Vallecillo, A., Ruiz, F., Piattini, M. y Genero, M.; *Towards a consistent terminology for software measurement*, Information and Software Technology 48 (8). (2006). pp.631-644
11. García, F., Bertoa, M. F. y Vallecillo, A.; *An ontology for software measurement*, Technical Report, UCLM DIAB-04-02-2, (2004).
12. García, F., Piattini, M., Ruiz, F., Canfora, G. y Visaggio, C. A.; *FMESP: Framework for the modeling and evaluation of software processes*, Journal of Systems Architecture - Agile Methodologies for Software Production. 52. (2006). pp.627-639.
13. García, F., Ruiz, F., Calero, C., Bertoa, M. F., Vallecillo, A., Mora, B. y Piattini, M.; *On the Effective Use of Ontologies in Software Measurement*, The Knowledge Engineering Review (in press). 0. (2008). pp.1-24.
14. García, F., Serrano, M., Cruz-Lemus, J., Ruiz, F. y Piattini, M.; *Managing Software Process Measurement: A Metamodel-Based Approach*, Inf Sciences. Vol 177. pp.2570-2586 (2007).
15. Guerra, E., Lara, J. d. y Díaz, P.; *Visual specification of measurements and redesigns for domain specific visual languages*, Journal of Visual Languages and Computing. (2008).
16. ISO/IEC, *ISO 15939: Software Engineering - Software Measurement Process*, (2002).
17. ISO/IEC; *Software and Systems Engineering - Guidelines for the application of ISO/IEC 9001:2000 to Computer Software*, International Standards Organization. (2004).
18. Jouault, F., Allilaire, F., Bézuvin, J., Kurtev, J. y Valduriez, P.; *ATL: a QVT-like Transformation Language*, Companion to the 21th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2006. Portland, Oregon, USA. (2006). pp.719-720.
19. Jouault, F. y Bézuvin, J.; *KM3: a DSL for Metamodel Specification*, Formal Methods for Open Object-Based Distributed Systems, 8th IFIP WG 6.1 International Conference, FMOODS 2006. 4037. Bologna, Italy. (2006). pp.171-185.
20. Kolovos, D. S., Paige, R. F., Kelly, T. y Polack, F. A. C.; *Requirements for Domain-Specific Languages*, First ECOOP Workshop on Domain-Specific Program Development (ECOOP'06). Nantes, France. (2006).
21. Kurtev, J., Bézuvin, J., Jouault, F. y Valduriez, P.; *Model-based DSL Frameworks*, (2006).
22. MacDonell, S. G., Shepperd, M. J. y Sallis, P. J.; *Metrics for Database Systems: An Empirical Study*, IEEE METRICS 1997.
23. McGarry, J., Card, D., Jones, C., Layman, B., Clark, E., Dean, J. y Hall, F.; *Practical Software Measurement. Objective Information for Decision Makers*, (2002).
24. Memik, M., Heering, J. y Sloane, A. M.; *When and how to develop domain-specific languages*, ACM Computing Surveys (CSUR). Volume 37. (2005). pp.316-344.
25. Mora, B., García, F., Ruiz, F., Piattini, M., Boronat, A., Gómez, A., et al. *Software Measurement by using QVT Transformation in an MDA context*, ICEIS 2008. pp.117-124.
26. Mora, B., Ruiz, F., García, F. y Piattini, M.; *SMML: Software Measurement Modeling Language*, Technical Report UCLM-TSI-003, (2008). www.esi.uclm.es:8080/tisi/informes
27. *OMG, OCL 2.0 - OMG Final Adopted Specification*, Object Management Group. (2003).
28. *OMG, Architecture-Driven Modernization (ADM): Software Metrics Meta-Model (SMM)*.
29. Özgür, T.; *Comparison of Microsoft DSL Tools and Eclipse Modeling Frameworks for Domain-Specific Modeling in the context of the Model Driven Development*, 2007.
30. Pelechano, V., Albert, M., Javier, M. y Carlos, C.; *Building Tools for Model Driven Development comparing Microsoft DSL Tools and Eclipse Modeling Plug-ins*, DSDM'06. Sánchez, J., García, J. y Menárguez, M.; *RubyTL: A Practical, Extensible Transformation Language*, ECMDA-FA 2006 4066/2006. (2006). pp.158-172.

Resumen. Recientemente OMG ha propuesto SPEM, una iniciativa para el modelado conjunto de procesos y métodos. Sin embargo, aunque las Líneas de Procesos están en pleno auge, SPEM no considera los mecanismos de variación adecuados para modelarlas. Por ello, en este artículo se proponen nuevos mecanismos para SPEM con los que podrá dar soporte a las Líneas de Procesos Software. Los nuevos mecanismos de variabilidad propuestos aportan a SPEM la capacidad de variación en la que se basan las Líneas de Procesos, mediante la gestión de *puntos de variación* y *variantes* y las relaciones entre ellos. Con la incorporación de estos mecanismos, SPEM soportará el modelado de Líneas de Proceso Software, tal y como se puede ver aplicado al modelo COMPETISOFT. Todo ello facilitará la creación de modelos de procesos adaptados a diferentes contextos organizacionales, propiciando la mejor integración de los procesos dentro de los cuerpos organizacionales.

Palabras clave: SPEM, Línea de Procesos Software, Variabilidad, Punto de Variación, Variante

1 Introducción

La industria dedicada al desarrollo de software está interesada en el modelado de los procesos que se llevan a cabo dentro de las organizaciones como un mecanismo eficaz para conocer y mejorar dichos procesos. En respuesta a esta demanda, OMG ha diseñado SPEM [8], un metamodelo que estandariza el modelado tanto de procesos como métodos.

Sin embargo, los procesos que suceden en empresas distintas son, por definición, distintos. Esto implica que un modelo de procesos genérico como CMMI o ISO/IEC 12207 no se puede implantar de igual forma en cualquier organización [13]. Estos modelos se deben adaptar a las condiciones específicas de cada organización y proyecto, dado que como afirma Humphrey *Al igual que no hay dos proyectos iguales, tampoco hay dos procesos iguales en el mundo* [3]. Como resultado de la adaptación de un proceso a diferentes proyectos y organizaciones específicas en los que se lleva a cabo, se genera un conjunto de procesos muy parecidos pero con pequeñas diferencias entre ellos que los hacen apropiados

para un contexto determinado. Estos procesos conforman una *Línea de Procesos Software* [9], y se basan en la gestión de variaciones utilizando mecanismos específicos que soporten el manejo de los *variantes* y *puntos de variación*, los elementos que permiten la adaptación de procesos.

Debido al auge de organizaciones dedicadas al desarrollo de software, las *Líneas de Procesos* se convertirán en un enfoque eficaz para la creación de los procesos que estas organizaciones demanden, adaptados a sus propios proyectos y cultura organizacional. Ante este auge, es necesario definir un metamodelo que permita el modelado tanto de procesos software como de las *Líneas de Procesos* de las que estos se derivan, a través de la ampliación de SPEM con mecanismos de variabilidad específicamente diseñados para *Líneas de Procesos Software*.

En este paper se definen los mecanismos de variabilidad necesarios para dotar a SPEM de la capacidad de modelado de *Líneas de Proceso Software*. En la Secc. 2 se analizan distintas iniciativas de variabilidad en Líneas de Producto. La Secc. 3 muestra un resumen de SPEM y en la Secc. 4 se explican los nuevos mecanismos propuestos, ilustrados con un ejemplo. Finalmente la Secc. 5 contiene las conclusiones y trabajo futuro.

2 Estado del Arte en Variabilidad

Tras una revisión en la literatura, no hemos encontrado ninguna iniciativa en la que se definan técnicas de gestión de variabilidad en *Líneas de Procesos Software*. Sin embargo sí que se han encontrado diversas iniciativas de gestión de variabilidad en Líneas de Producto. La mayoría de estos mecanismos de variabilidad se basan en *puntos de variación* y *variantes*. Un *punto de variación* es el lugar donde ocurre la variabilidad [2], mientras que las *variantes* son los elementos concretos que se colocan en los puntos de variación y que implementan esa variabilidad de una forma determinada.

En [12], tras un análisis de los diferentes enfoques mediante los que se puede modelar la variabilidad en una Línea de Productos, se determina que la utilización de variantes y puntos de variación permite, entre otras ventajas, la posibilidad de añadir nuevas implementaciones de la variabilidad, mediante la adición de variantes. Además, según [1], es posible especificar dependencias y restricciones entre estos elementos. En [11] además se definen dependencias entre los puntos de variación y tres niveles de abstracción mediante los que modelar la variabilidad.

La metodología *Pulse* define los elementos básicos de una Línea de Productos Software: puntos de variación, variantes, relaciones entre estos, restricciones y un mecanismo de generación de productos. Además se incluye un Modelo de Decisión a un nivel más alto para definir las variantes del sistema [10].

En [14] se define una extensión a UML para modelar puntos de variación y variantes, además, se especifica que la elección de una variante afecta a la arquitectura del software, y se distingue a aquellas variantes que afectan de forma transversal a todo el sistema.

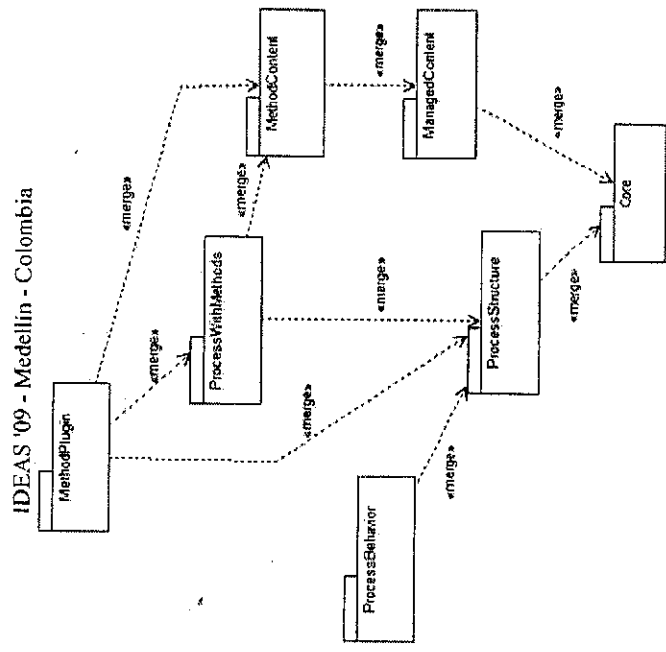


Fig. 1. Estructura del estándar SPEM v2

Partiendo de las ideas de estas propuestas, se pueden diseñar mecanismos de variabilidad para Líneas de Procesos Software. Estos mecanismos se deberían añadir a la definición de procesos que hace SPEM para permitirle la especificación de Líneas de Procesos.

3 Software Process Engineering Metamodel

SPEM (Software Process Engineering Metamodel) [8] es una iniciativa de OMG (Object Management Group) para el modelado de procesos y métodos que se basa en otros modelos de OMG como MOF (Meta Object Facility) o UML (Unified Modeling Language). Este metamodelo está siendo ampliamente usado para modelar procesos y goza de gran aceptación en la industria. En la versión 2.0 del estándar, se aportan nuevas capacidades sobre la versión anterior. En la Fig. 1 se muestran los paquetes del modelo y sus dependencias. Además, SPEM define una notación gráfica para sus elementos con la que representar diagramas de procesos y métodos.

3.1 Variabilidad en SPEM

SPEM v2.0 incluye la definición de variabilidad dentro de su paquete *MethodPlugin*, pero orientada a la composición de procesos utilizando métodos.

La clase abstracta *VariabilityElement* permite añadir el concepto de variabilidad a la definición de procesos y métodos aportada en los paquetes *ProcessStructure* y *MethodContent*, respectivamente. Dado que de ella heredan las clases *Activity*, *MethodContentElement* y *Section*. Éstas y sus especificaciones adquieren capacidad para "variar". La enumeración *VariabilityType* define el tipo de variabilidad que se da entre las dos instancias de *VariabilityElement*. Además de *na* (por defecto), incluye los siguientes valores:

Contributes Es una relación de variabilidad que permite la adición de las propiedades de un elemento a otro base, sin alterar el contenido de este último. El elemento base puede tener más de un contribuyente.

Replaces Es un mecanismo de variabilidad que permite el reemplazo del elemento base por otro, sin alterar sus propiedades. Un elemento base sólo puede tener una relación *replaces*. Ésta relación es transitiva.

Extends Esta relación es un mecanismo de herencia entre elementos. Es similar a la de herencia en UML. La relación *extends* es transitiva y tanto *contributes* como *replaces* tienen mayor prioridad.

Extends-replaces Combina los efectos de las dos relaciones anteriores. Mientras que *replaces* sustituye todas las propiedades del elemento base, esta relación sólo reemplaza aquellos valores que han sido redefinidos en el elemento reemplazante.

3.2 Limitaciones de SPEM para Modelar Variabilidad en Líneas de Proceso Software

Como resultado del análisis a SPEM realizado, se han detectado algunas limitaciones en su uso para modelar Líneas de Proceso Software:

1. La variabilidad propuesta se orienta más hacia la variación de Métodos que de Procesos. No en vano, de la clase *VariabilityElement* heredan *methodContent*, superclase de todas las clases que permiten el diseño de métodos, y únicamente la clase *Activity*, incluida en el modelado de procesos. Dado que el proceso es la entidad que se ejecuta, sería conveniente efectuar las variaciones/adaptaciones necesarias para su ejecución en el contexto de un proyecto en el proceso, sin implicar al método del que éste deriva. Aunque dentro de los procesos de SPEM se pueden variar las actividades (clase *Activity*), las variaciones para adaptar estos procesos a los diferentes contextos no se pueden limitar exclusivamente a las actividades, si no que deben involucrar también el resto de elementos que se relacionan con ellas.
2. Puede resultar difícil la reutilización de los *plugins* de SPEM v2.0 con los mecanismos de variabilidad definidos. En el paquete *MethodPlugin* de SPEM v2.0 se definen los elementos *MethodPlugin* como paquetes extensibles que representan un contenedor físico para paquetes de contenidos y procesos [8]. Esto tiene el inconveniente de que estos incluyen todas las modificaciones que hay que realizar a un proceso para adaptarlo a un contexto, Por tanto, ese conjunto de modificaciones será difícilmente aplicable a otro proceso diferente.

3. Estos mecanismos de variabilidad no permiten la delimitación de la parte común a todos los procesos de la Línea de Procesos Software, denominada *core process*, ni acotar el rango de posibles variaciones aplicables a los procesos de la línea.

Aunque la parte común a un conjunto de procesos se pudiera considerar como un *plug-in* base, que se personalice a través de la extensión mediante otros *plug-ins*, el mecanismo de variabilidad *replaces* que se pudiera ubicar en cualquier *plug-in* tiene capacidad de sustituir elementos del *plug-in* base, alterando de esta forma el *core process*.

4. SPEM v2.0 reutiliza la notación de UML, no incluye notación específica para representar la variabilidad de procesos, lo que hace difícil entender algunos diagramas.

En resumen, debido a las limitaciones enumeradas anteriormente, se cree que los mecanismos de variabilidad definidos en SPEM v2.0 no son los apropiados para modelar variabilidad en Líneas de Procesos Software.

4 Nuevos Mecanismos de Variabilidad

Nuestra propuesta se basa en los mecanismos para la gestión de variabilidad definidos por las Líneas de Productos Software. Se pretende, a partir de dichos mecanismos, desarrollar otros que permitan modelar la variabilidad en *Líneas de Procesos Software*. A continuación se enumeran los elementos que, a nuestro juicio, serán necesarios.

Puntos de variación y variantes Al igual que se ha identificado para las Líneas de Productos Software, los puntos de variación se corresponderán con los puntos dentro del modelo de procesos donde ocurrirá la variación (adaptado de [10]). Serán elementos "vacíos" del modelo que se ocuparán con variantes para así permitir la generación de un nuevo proceso adaptado al contexto en el que se va a desarrollar. Las variantes se corresponderán con distintos elementos (*VActivity*, *VWorkProductDefinition*, etc.) que se pueden insertar en unos determinados puntos de variación, que adaptan el proceso al contexto. Debido a que en un proceso intervienen distintos tipos de elementos, también existirán diferentes tipos de Variantes y Puntos de Variación.

Dependencias La variabilidad en una entidad tan compleja como un proceso, no se puede ver como un hecho puntual, si no que afectará a distintas partes de su estructura. En este sentido, distintos elementos de la línea podrán depender de otros.

Restricciones Las dependencias pueden implicar restricciones sobre los elementos de la Línea de Procesos, de cara a garantizar la consistencia del modelo generado.

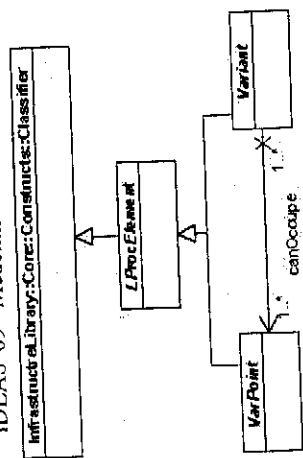


Fig. 2. Metamodelo de las Variantes y Puntos de Variación genéricos

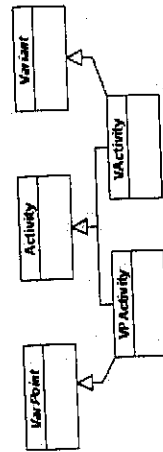


Fig. 3. Creación de Variantes y Puntos de Variación concretos en base a los genéricos

4.1 Nuevos Mecanismos de Variabilidad Añadidos a SPEM v2.0

Al igual que en UML se pueden definir perfiles que extienden su metamodelo aportándole nuevas capacidades para dominios específicos [4], el metamodelo de SPEM se puede extender para que dé soporte a los mecanismos de variabilidad definidos en el punto anterior. En este sentido, se pueden añadir diferentes extensiones al paquete *MethodPlugin* del estándar. Para garantizar la consistencia y correcta utilización de los nuevos elementos, se utilizan restricciones OCL [7]. En primer lugar, habrá que diseñar un *LProcElement*, éste elemento es una abstracción tanto de los puntos de variación (*VarPoints*), como de las variantes (*Variant*), tal y como puede verse en la Fig. 2. A su vez, *LProcElement* es una especificación abstracta de *Classifier*, del paquete de UML.

Las variantes deben conocer a los puntos de variación que puede ocupar, esto se representa mediante la relación *canOccupe*. Estas clases también serán abstractas debido a que pueden existir diferentes variantes y puntos de variación de distinta naturaleza. En la Fig. 3 se puede ver la relación de estas clases abstractas con los elementos concretos descritos en SPEM.

Los modelos de procesos contendrán las distintas especificaciones de *VarPoint*, que serán "ocupadas" por las especificaciones de *Variant* correspondientes. La relación de *Occupación (Occupation)*, que hereda de la relación de *Generalización de UML*, puede ser de varios tipos (*Opcional, Obligatorio o Alternativo*) (Fig. 4). Esta relación se denota con una flecha con un círculo en la punta trasera y un triángulo relleno en la punta delantera (Fig. 5), para diferenciarlo del resto de relaciones que se dan en SPEM.

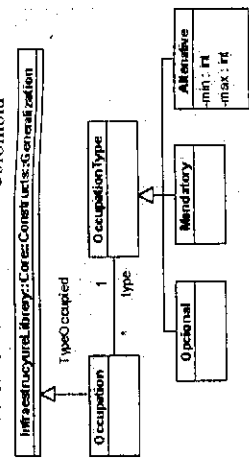


Fig. 4. Metamodelo de la Relación de *Occupación*

Fig. 5. Notación de la relación de *Occupación*

En la Tabla 1 se especifican las restricciones que obligan a que los elementos entre los que se establece la relación *Occupación* sean compatibles para garantizar que la variante encaje en el punto de variación. Por una parte ambos (*Variante* y *punto de Variación*) tienen que ser especificaciones del mismo tipo concreto, y por otra parte, también hay que asegurar que la variante ha sido diseñada para ocupar este punto de variación, es decir, el punto de variación en concreto se encuentra en el conjunto de puntos de variación que puede ocupar ("canOccupe") la variante. De forma implícita, se especifica que la relación de *Occupación* se debe establecer entre un punto de variación y una variante. De igual manera, los elementos relacionados a través de la relación de *Occupación*

Finalmente, entre los elementos también se podrán establecer dependencias, denominadas *VariabilityDependencies*, que pueden ser de varios tipos dependiendo de los elementos entre los que se defina (Fig. 6). Éstas son una especificación de las dependencias de UML, que sólo se pueden dar entre elementos propios de las *Líneas de Procesos Software*, según la restricción de la Tabla 2.

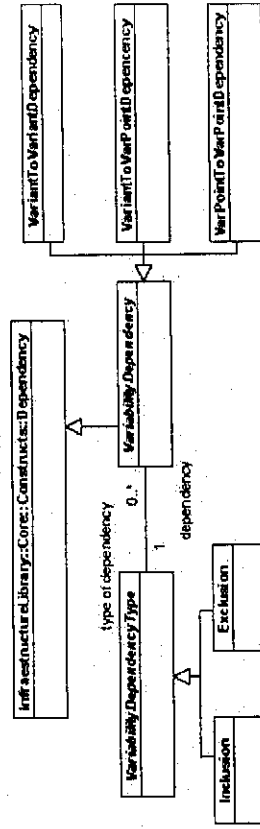


Fig. 6. Dependencias de Variabilidad y sus relaciones

```

constraint Occupation inv:
    (self.general.OcclsTypeOf (VPRoleDefinition) implies
    self.specific.OcclsTypeOf (VRoleDefinition)
    or self.general.OcclsTypeOf (VPAActivity) implies
    self.specific.OcclsTypeOf(VAActivity)
    or self.general.OcclsTypeOf (VPTaskDefinition) implies
    self.specific.OcclsTypeOf(VTaskDefinition)
    or self.general.OcclsTypeOf (VPWorkProduct) implies
    self.specific.OcclsTypeOf(VWorkProduct))
    and self.specific.canOccupe->includes(self.general)
    and
    self.specific.canOccupe->includes(self.general)
    
```

Table 2. Restricciones de las dependencias de Variabilidad

```

constraint VariabilityDependency inv:
    self.supplier.OcclsKindOf (LPProcElement) and
    self.client.OcclsKindOf (LPProcElement)
    
```

Para facilitar la utilización del modelo, se proponen nuevos iconos mediante los cuales representar gráficamente los elementos especificados. En la Tabla 3 se muestran estos iconos junto a los iconos con que se relaciona.

4.2 Ejemplo de Aplicación

Para ilustrar la propuesta en este apartado se muestra un caso de estudio realizado sobre el modelo de procesos de COMPETISOFT [5, 6]. En este proyecto iberoamericano se está definiendo un Marco de Mejora de Procesos Software

Table 3. Nuevos iconos de Variantes y Puntos de variación

Elemento	Actividad	Producto de Trabajo	Rol	Tarea
Base				
Punto de Variación				
Variante				

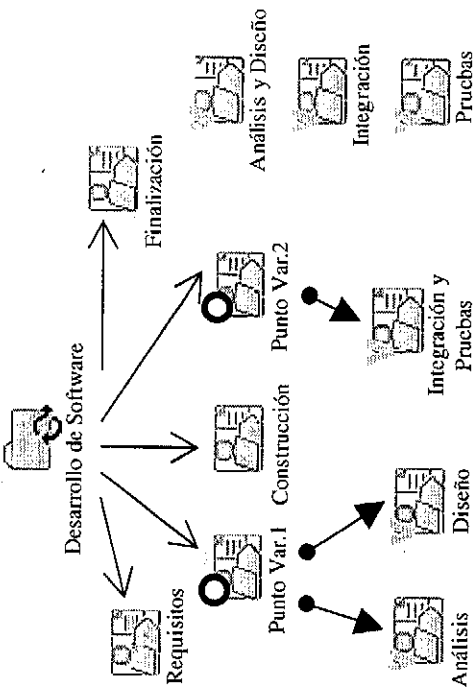


Fig. 7. Proceso de Desarrollo de COMPETISOFT, modelado en SPEM

para pequeñas organizaciones formado por diversas propuestas de mejora de procesos. Pretende ser utilizado, con las adaptaciones correspondientes, en los distintos países de Iberoamérica.

En el patrón de procesos del modelo se incluye un apartado denominado "Guías de Ajuste" que determinan posibles modificaciones a los procesos que no afectan a sus objetivos. Estas guías de ajuste constituyen un primer esbozo de introducción de variabilidad en el proceso. La extensión a SPEM propuesta en este artículo permite modelar las variaciones incluidas en las guías de ajuste de COMPETISOFT, de manera que se genera una Línea de Procesos Software que permite diseñar procesos adaptados a las necesidades de cada organización y proyecto. A continuación se enumeran las variaciones concretas tenidas en cuenta en el modelo de procesos de COMPETISOFT, y cómo se modelan utilizando los mecanismos de variabilidad propuestos para SPEM.

1. Aunque en COMPETISOFT se ha considerado las actividades correspondientes al *Análisis* y al *Diseño* por separado, en determinadas organizaciones puede resultar conveniente fusionarlas, denominando a la actividad resultante *Análisis y Diseño*. En este caso, el proceso tendrá un punto de variación de tipo actividad, que será realizado una variante que contenga la actividad conjunta o por dos de ellas con sendas actividades (Fig. 7). Un ejemplo similar al anterior ocurre con las actividades relativas a la *Integración y Pruebas*. Como se puede ver en la Fig. 7, el proceso de Desarrollo tiene asociadas tres actividades, *Requisitos*, *Construcción* y *Finalización*, y dos puntos de variación, *Punto Var.1* y *Punto Var.2*. Asimismo se puede ver que los puntos de variación son ocupados por variantes de tipo actividad, el *Punto Var.1* es ocupado por las variantes *Análisis y Diseño*, mientras que el *Punto Var.2* se ocupa con la variante *Integración y Pruebas*. De esta forma, el proceso de De-

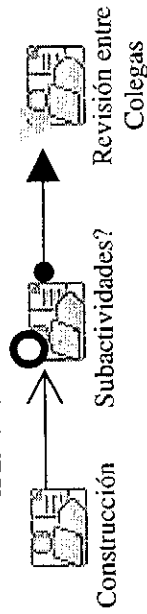


Fig. 8. Actividad de *Revisión entre colegas* optativa para la actividad de *Construcción*

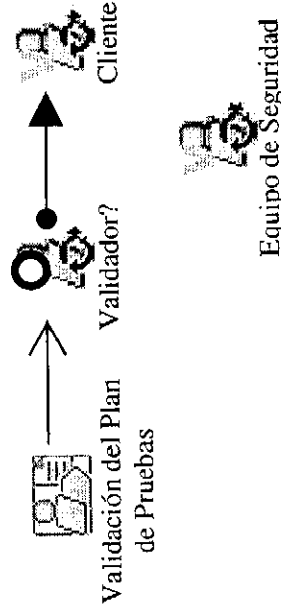


Fig. 9. Variación en el modelado de los roles encargados de realizar la *Validación del Plan de Pruebas*

sarrollo resultante de esta configuración tendrá seis actividades: *Requisitos, Análisis, Diseño, Construcción Integración y Pruebas, y Finalización*

2. En la actividad de *Construcción* se puede añadir la (sub)actividad de *Revisión entre Colegas* para verificar el código de los componentes. Esta actividad se encapsulará dentro de una variante, que podrá ocupar el punto de variación colocado en *Construcción* (Fig. 8).

En este caso se puede ver que a la actividad de *Construcción* se le ha añadido un punto de variación de tipo actividad mediante el cual se le pueden agregar subactividades, y que ha sido ocupado por la variante correspondiente con la *Revisión entre colegas*.

3. El *Plan de Pruebas del Sistema* se puede validar con el *Cliente* y con el *Equipo de Seguridad*. En este caso dicha (sub)actividad tendrá un punto de variación de tipo rol, que podrá ser ocupado con las variantes correspondientes a los dos roles comentados (Fig. 9).

Según se puede ver en la Fig. 9, el rol encargado de realizar la actividad de *Validación del Plan de Pruebas* debe ser un punto de variación de tipo rol. Este punto de variación podrá ser ocupado por las variantes de tipo rol correspondientes con el *Cliente*, o con el *Equipo de Seguridad*.

4. En la actividad de *Definición de Pruebas Unitarias* se puede tener en cuenta un estándar *IEEE*. Este documento será una variante de entrada a la tarea comentada. En la Fig. 10 se puede observar la actividad de *Definición de Pruebas Unitarias*, que tiene relacionado un punto de variación de tipo actividad que podrá ser ocupado por las variantes correspondientes con el estándar *IEEE 1008 1987* o con *otro estándar*.

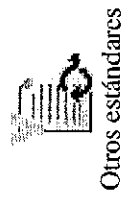
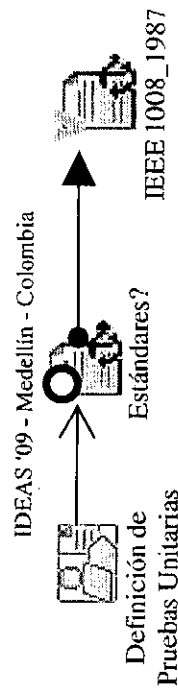


Fig. 10. Variabilidad en el uso de estándares de la actividad *Definición de Pruebas Unitarias*

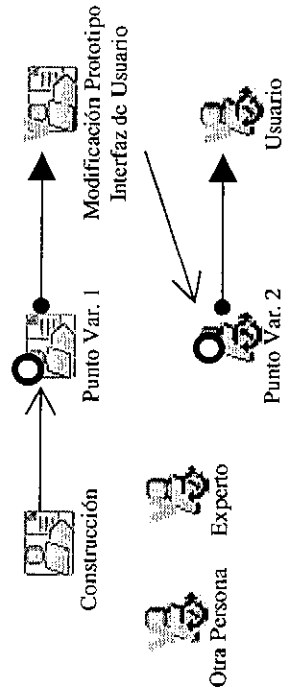


Fig. 11. Doble variabilidad con las subactividades de *Modificación del Prototipo de la Interfaz de Usuario* y con el rol encargado de realizarla

5. En la actividad de *Construcción*, se puede agregar una (sub)actividad de *Modificación del Prototipo de la Interfaz de Usuario*, en la que debe participar el *Usuario*, un *Experto* o individuos con un perfil similar. En este caso, la actividad de *construcción* tendrá un punto de variación donde se puede insertar la variante correspondiente a la (sub)actividad comentada. Esta variante también tendrá un punto de variación que podrá ser ocupado por cualquiera de los roles descritos antes.

La Fig. 11 muestra la actividad de *Construcción* en la que se incluye un punto de variación, que podrá ocuparse por la variante con la subactividad de *Modificación del Prototipo de la Interfaz de Usuario*. Como se puede ver en la Fig. 11, la actividad de *Modificación del Prototipo de la Interfaz de Usuario* tiene un punto de variación de tipo rol, que puede ocuparse por un *usuario*, un *experto* u *otra persona*. En este caso, el primer punto de variación se ocupa con la variante correspondiente a la subactividad, y el segundo, es ocupado con la variante correspondiente con el *Usuario*.

Como se ha podido ver en este ejemplo, con los nuevos mecanismos de variabilidad definidos, el proceso de desarrollo de software de *COMPETISOFT* puede ser adaptado a diferentes contextos. Mediante esta adaptación, se han

generado diversos procesos casi idénticos que forman parte de una Líneas de Procesos Software.

5 Conclusiones y Trabajo Futuro

Como se ha motivado en el presente trabajo, las Líneas de Procesos Software son una poderosa herramienta para permitir la adaptación de procesos a las condiciones específicas de la organización y los proyectos que ésta lleve a cabo en cada momento.

La inclusión de variabilidad en los mecanismos de SPEM enfocados a la creación de procesos permitirá el desarrollo de líneas de procesos software adaptables, y a partir de éstas será posible definir procesos en los que se tenga en cuenta las necesidades del entorno en el que se van a desarrollar. De esta forma, SPEM dará soporte al modelado de procesos específicos interpretados según las requerimientos de cada uno de los proyectos que se lleven a cabo en la organización.

Las ventajas que ofrece esta propuesta de inclusión de mecanismos de variabilidad dentro de SPEM se basan en que: (i) permitiendo la especificación de forma clara de las partes del proceso que pueden variar y acotando la variación para asegurar que el resto del proceso no es modificado; (ii) al definir variantes muy concretas, son fácilmente reutilizables; (iii) se incluye una notación gráfica para representar la variabilidad en diagramas, tal y como se ha visto en los ejemplos basados en el modelo de procesos de COMPETISOFT.

El trabajo futuro consistirá en refinar estos mecanismos, así como validarlos aplicando casos de estudio en COMPETISOFT. Asimismo se pretende implementar una herramienta, basada en el plugin EPF composer de SPEM, que incluya los mecanismos de variabilidad descritos. De igual forma, puede ser interesante integrar técnicas derivadas de la *Programación Orientada a Aspectos*, para la gestión de variaciones que afecten de manera transversal a todos los procesos de un modelo.

Agradecimientos. Este trabajo forma parte de la investigación sobre Líneas de Procesos Software patrocinada por Sistemas Técnicos de Loterías del Estado S.A. en el marco del convenio sobre Innovación del Entorno Metodológico de Desarrollo y Mantenimiento de Software, el Programa de Formación de Profesorado Universitario (FPU) del Ministerio de Ciencia e Innovación (MICINN) y de los proyectos COMPETISOFT 506AC0287 del programa CYTED, ES-FINGE TIN2006-15175-C05-05 del Ministerio de Ciencia y Tecnología (España) e INGENIO de la Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia, PAC08-0154-9262.

Referencias

1. Asikainen, T., Männistö, T., Soiminen, T.: Kumbang: A domain ontology for modeling variability in software product families. *Advanced Engineering Informatics* 21(1), 23-40 (2007)

2. Claus, M.: Generic modeling using UML extensions for variability. In: *Workshop on domain specific visual languages*. Tampa Bay, Florida (2001)
3. Humphrey, W.: *Managing the Software Process*. Addison-Wesley (1989)
4. Korherr, B., List, B.: A UML 2 profile for variability models and their dependency to business processes. In: *15 th International Workshop on Database and Expert Systems Applications*, pp. 829-834. Regensburg, Germany (2007)
5. Oktaba, H., Garcia, F., Piattini, M., Pino, F., Alquicira, C., Ruiz, F.: Software process improvement in small latin-american organizations: Competisoft project. *IEEE Computer* 40(10), pp. 21-28 (2007)
6. Oktaba, H., Piattini, M., Pino, F., Garcia, F., Alquicira, C., Ruiz, F., Martincz, T.: Competisoft: A improvement strategy for small latin-american software organizations. In: H. Oktaba, M. Piattini (eds.) *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies*. Idea Group Inc. (2008)
7. OMG: The object constraint language specification- version 2.0. *Tech. rep.*, Object Management Group (2004)
8. OMG: Software process engineering metamodel specification. *Tech. Rep. ptc/07-03-03*, Object Management Group (2007)
9. Rombach, D.: *Integrated software process and product lines*. In: M. Li, B. Boehm, L. Osterweil (eds.) *International Software Process Workshop 2005*, LNCS 3840, pp. 83-90. Springer-Verlag, Teddington, UK (2005)
10. Schmid, K., John, I.: A customizable approach to full lifecycle variability management. *Science of Computer Programming* 53(3), 259-284 (2004)
11. Sinnema, M., Deelstra, S.: Industrial validation of covamof. *Journal of Systems and Software* 49(1), 717-739 (2007)
12. Webber, D., Gornaa, H.: Modeling variability in software product lines with the variation point model. *Science of Computer Programming* 53(3), 305-331 (2004)
13. Yoon, I.C., Min, S.Y., Bae, D.H.: Tailoring and verifying software process. In: *Proceedings of the Eighth Asia-Pacific Software Engineering Conference (APSEC-01)*, pp. 202-209 (2001)
14. Zhang, H., Jarzabek, S.: XVCL: a mechanism for handling variants in software product lines. *Science of Computer Programming* 53(3), 381-407 (2004)