



VII Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento

Memoria Técnica

30 de Enero al 1 de Febrero
Guayaquil - Ecuador

VII Jornadas Iberoamericanas

de Ingeniería de Software e Ingeniería del Conocimiento

Comité Organizador

Mónica Villavicencio Cabezas, Escuela Superior Politécnica del Litoral del Ecuador (chair).

María Verónica Macías Mendoza, Escuela Superior Politécnica del Litoral del Ecuador.

Carlos Monsalve Arteaga, Escuela Superior Politécnica del Litoral del Ecuador.

Grupo Editor de la Revista

Mónica Villavicencio - Directora

Lohana Lema Moreta - Asistente

Colaboradores de edición

Stephanie Flores

Guillermo Pizarro

Diseño de portada

Luis Bajaña

Colaboradores

Emilio Rigazio

David Jurado

Karina Chong

Fátima Cedeño



JISIC'08

VII Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento

Guayaquil – Ecuador
Del 30 de Enero al 1 de Febrero del 2008

Editado y Compilado por:
Escuela Superior Politécnica del Litoral
Facultad de Ingeniería Eléctrica y Computación
Área de Ingeniería en Software VLIR –ESPOL Componente 8



VII Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento

Compilado por:

Escuela Superior Politécnica del Litoral
Facultad de Ingeniería Eléctrica y Computación
Área de Ingeniería en Software VLIR –ESPOL Componente 8

Editado por:

Mónica Villavicencio, Carlos Monsalve, Verónica Macías, Guillermo Pizarro, Lohana Lema y
Stephanie Flores.

Primera Edición: enero 2008

Jornadas de Ingeniería de Software (ISSN 1390-292X) será publicada cada vez que se organice unas Jornadas por el Área de Ingeniería de Software del Componente 8 del Proyecto VLIR-ESPOL.

Comité Permanente

Silvia Teresita Acuña, Universidad Autónoma de Madrid, España
 Manoel Mendonça, Universidad Salvador, Brasil
 Oscar Dieste, Universidad Complutense de Madrid, España
 José Antonio Pow-Sang, Universidad Católica de Perú, Perú

Comité Organizador

Mónica Villavicencio Cabezas, Escuela Superior Politécnica del Litoral del Ecuador (chair)
 María Verónica Macías Mendoza, Escuela Superior Politécnica del Litoral del Ecuador
 Carlos Monsalve Arteaga, Escuela Superior Politécnica del Litoral del Ecuador

Comité de Programa

Aguilar Raúl, Universidad Autónoma de Yucatán - México
 Álvarez Luis Alberto, Universidad Austral - Chile
 Álvarez Marco, Utah State University, EEUU - USA
 Antúnes Pedro, Universidad de Lisboa - Portugal
 Aveledo Marianela, Universidad Simón Bolívar - Venezuela
 Carrizo Dante, Universidad Complutense de Madrid - España
 Chiluzza Katherine, Escuela Superior Politécnica del Litoral - Ecuador
 De Antonio Angélica, Universidad Politécnica de Madrid – España
 De Castro Valeria, Universidad Rey Juan Carlos - España
 Duran Amador, Universidad de Sevilla - España
 Echagüe Juan Vicente, Universidad de la República - Uruguay
 Eterovic Yadrán, Pontificia Universidad Católica de Chile - Chile
 Fernández Mariano, Universidad CEU San Pablo - España
 Ferre Xavier, Universidad Politécnica de Madrid - España
 García Ramon, Instituto Tecnológico de Buenos Aires – Argentina

Comité del Programa (continuación)

García Francisco José, Universidad de Salamanca - España
 Gómez Marta, Universidad CEU San Pablo - España
 Grimán Anna, Universidad Simón Bolívar - Venezuela
 Guerrero Luis, Universidad de Chile - Chile
 Imbert Ricardo, Universidad Politécnica de Madrid - España
 Jino Mario, Universidad Estadual de Campinas - Brasil
 Kong Maynard, Pontificia Universidad Católica del Perú – Perú
 Macías María Verónica, Escuela Superior Politécnica del Litoral - Ecuador
 Macías José Antonio, Universidad Autónoma de Madrid - España
 Muñoz Jaime, Universidad Autónoma de Aguascalientes - México
 Pérez Melvin, CAM Informática - República Dominicana
 Pons Claudia, Universidad Nacional de la Plata – Argentina
 Pow-Sang José Antonio, Pontificia Universidad Católica del Perú - Perú

Rodríguez Gustavo, INAOE - México
Sánchez Segura María Isabel, Universidad Carlos III de Madrid - España
Sierra Enrique, Instituto Tecnológico de Buenos Aires - Argentina
Tirado Francisco, Universidad Complutense de Madrid - España
Triñanes Jorge, Universidad de la República - Uruguay
Tupia Manuel, Pontificia Universidad Católica del Perú - Perú
Vegas Sira, Universidad Politécnica de Madrid - España
Vergilio Silvia Regina, Universidades Federal do Paraná - Brasil
Visconti Marcello, Universidad Técnica Federico Santa María - Chile
Vizcaíno Barceló Aurora, Universidad de Castilla-La Mancha – España

Prólogo

La presente publicación contiene el compendio de 49 artículos presentados y aprobados por el comité del programa de las VII Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del Conocimiento, las mismas que se desarrollaron en Guayaquil, Ecuador del 30 de Enero al 1 de febrero del 2008.

JISIC es un evento que ha logrado posicionarse y mantenerse a nivel internacional, acogiendo a activos investigadores y profesionales interesados en presentar los resultados de sus trabajos de investigación, garantizando así su difusión y promoviendo el conocimiento.

En esta oportunidad 84 artículos de calidad científica fueron recibidos para su evaluación, de los cuales 49 fueron aceptados luego de 3 revisiones. Los autores de los artículos representaron a universidades y empresas de Argentina, Brasil, Colombia, Cuba, Chile, Ecuador, España, México, Perú, Uruguay y Venezuela. Adicionalmente, el evento contó con la presentación de 4 tutoriales y 4 conferencias magistrales.

Es justo aprovechar esta oportunidad para agradecer a quienes colaboraron incansablemente en el desarrollo de JISIC 2008; entre ellos debo de mencionar ante todo a Lohana Lema por su entrega ilimitada, luego a Guillermo Pizarro, Luis Bajaña, Verónica Macías, Stephanie Flores y al grupo de voluntarios de la FIEC y de rama estudiantil de la IEEE. Un agradecimiento muy especial a José Antonio Pow-Sang por su invaluable y desinteresada ayuda. Finalmente, agradezco a todos los autores por habernos escogido y participar de este evento.

Mónica Villavicencio
Presidente del Comité Organizador

INDICE

Sección I: Base de Datos y Minería de Datos

TariyKDD: una herramienta de minería de datos débilmente acoplada con un SGBD <i>Ricardo Timarán Pereira, Andrés O. Calderón Romero, Iván Ramírez Freyre, Fernando Guevara, Juan Carlos Alvarado</i>	3
El Modelo de Negocios Decisional como origen de Especificación de Requisitos en Proyectos de Data Mining: Una Aproximación Metodológica mediante el Framework i* <i>José Gallardo, Óscar Marbán, Claudio Meneses y Aldo Quelopana</i>	13
Un Ambiente de Explotacion de Información basado en la Integración de Agrupamiento, Inducción y Ponderación Bayesiana de Reglas <i>G. Schulz, E. Fernández, H. Merlino, D. Rodríguez, P. Britos, R. García-Martínez</i>	21
Mineria de Datos Aplicada a la Detección de Patrones Delictivos en Argentina <i>F. Valenga, E. Fernández, H. Merlino, D. Rodríguez, C. Procopio, P. Britos y R. García-Martínez</i>	31
Paralelización de Consultas del tipo time-interval en Base de Espacio-Temporales <i>Claudio Gutiérrez-Soto, Gilberto Gutiérrez, Pedro Rodríguez, Pedro Campos</i>	41

Sección II: Calidad, Procesamiento y Administración de Software

Sección II-a:

Incidencia de la planificación y las inspecciones en el desarrollo de proyectos de software: Caso de Estudio realizado en Ecuador <i>Raúl González Carrión, Mónica Villavicencio Cabezas</i>	51
Factores de Éxito o Fracaso para la Mejora de Procesos Software: Caso Real en un Grupo de MiPyMEs <i>César Pardo, Julio Ariel Hurtado Alegria, Francisco J. Pino</i>	59
Una Experiencia de Implantación de COMPETISOFT en una Pequeña Empresa Desarrolladora de Software <i>Jackson Mogrovejo, Abraham Dávila</i>	67
Experiencia de Implementación de Mejora de Procesos en dos PYMEs Desarrolladoras de Software, que poseen certificación ISO 9001:2000 <i>Gonzalo Sánchez, Dianne Vergara, Abraham Dávila</i>	73

Sección II-b:

PROCODI: Lenguaje de Extensibilidad para UML <i>Daniel Alberto Giulianelli, Rocío Andrea Rodríguez, Pablo Martín Vera</i>	81
Modelado del proceso de software con enfoque de negocio. Aplicacion de los estandares BPMN y UML <i>Mabel del V. Sosa, Silvia T. Acuña, Juan de Lara</i>	89
Modelado de Mejora de Procesos de Software en Pequeñas Organizaciones <i>Ing. Pedro E. Colla, Dr. Jorge Marcelo Montagna</i>	97

Sección II-c:

Métricas de Madurez en Conceptualización de Sistemas Expertos <i>M. Pollo-Cattaneo, Fernández, H. Merlino, D. Rodríguez, P. Britos, R. García-Martínez</i>	107
---	-----

Un Análisis Crítico Comparativo de Modelos y Estándares Relacionados con la Adquisición de Software <i>Gloria Piedad Gasca Hurtado, Gonzalo Cuevas Agustín</i>	117
---	-----

Hacia la definición de un modelo para la Gestión de Proyectos en el Desarrollo Global del Software: reflexiones sobre la situación actual <i>Miguel Ángel Blanco, Ismael Caballero, Mario Piattini</i>	125
---	-----

Sección II-d:

PROMETEU - a tool to support documents generation and traceability in the test process <i>Jorge Luiz da Cruz, Mario Jino, Adalberto Nobiato Crespo, Miguel Argollo</i>	133
---	-----

Mapeo de los Procesos de RUP respecto a MoProSoft <i>Katia Cánepa y Abraham Dávila</i>	139
---	-----

Analysis of an Artifact Oriented Test Process Model and of Testing Aspects of ISO/IEC 15504 <i>Paulo M. S. Bueno, Adalberto N. Crespo, Clenio F. Salviano I, Mario Jino</i>	147
--	-----

Sección III: Diseño y Desarrollo de Software

Agilidad y disciplina en el Proceso de Desarrollo de Software para PyMES y Cooperativas en Latinoamérica: CASO VENEZUELA <i>George Di Paula, Dakar Parada, Maria Pérez, Luís Mendoza</i>	157
---	-----

Evolución del proceso de desarrollo de videojuegos en la Iniciativa Académica EDUMÓVIL <i>Gabriel Gerónimo-Castillo, Carlos Alberto Fernández-y-Fernández, Ricardo Ruiz-Rodríguez</i>	163
--	-----

Extensión a WSIL para la búsqueda y descubrimiento de servicios Web con calidad <i>Jesús Cruz-Ahuactzi, Giner Alor-Hernández, Ruben Posada-Gomez, Juan Miguel Gomez</i>	171
--	-----

Un modelo de arquitectura para el aprendizaje y compartición de conocimiento entre sistemas inteligentes autónomos distribuidos <i>Ierache, J., Naiouf, M., García Martínez, R., De Giusti, A.</i>	179
---	-----

Intérprete y Entorno de Desarrollo para el Aprendizaje de Lenguajes de Programación Estructurada <i>Layla Hirsh Martínez</i>	189
---	-----

Especificación Formal de Elementos MoProSoft a partir del Modelo de Referencia de Flujos de Trabajo <i>Leonel Valenzuela Ruiz, Brenda Leticia Flores Rios</i>	197
--	-----

Sección IV: Educación en Ingeniería de Software e Ingeniería del Conocimiento

Residência em Fábrica de Software: Um Caso Real e uma Proposta Genérica para a Normatização de Novos Programas <i>José Augusto Fabri, André Luiz Presende Trindade, Marcelo S. de Paula Pessôa</i>	207
---	-----

Un Modelo de Evaluación Adaptativa del Nivel de Conocimientos en Sistemas Tutoriales Inteligentes <i>Marcela Jiménez, Jovani A. Jiménez, Demetrio A. Ovalle</i>	215
--	-----

Sistema Tutor Inteligente con Tecnología de Agentes: La Elección del Método de Enseñanza	
--	--

<i>Zulma Cataldi, Patricia Calvo, Fernando J. Lage</i>	223
TONGO: Un laboratorio para apoyar la experimentación sobre arquitecturas orientada a servicios <i>Jorge Villalobos, Fabián Contreras</i>	231
Interacción con la Computadora: Modelo de Capacidades de Estudiantes de Nivel Medio de Jujuy <i>Viviana E. Quincoces, Héctor P. Liberatori, Ma. del Pilar Gálvez Díaz, Nilda M. Pérez Otero, Sandra A. Méndez, Adelina García, Cecilia Lasserre, Claudio M. Pérez Ibarra, Beatriz Fiorito</i>	239

Sección V: Herramientas y técnicas de Software

Meta ² Relational: Herramienta para la Gestión de Modelos de Procesos Software <i>Tomás Martínez-Ruiz, Félix García, Mario Piattini</i>	251
(Re)Composición de modelos ER con Idioms <i>Juan Marcelo Flores Soliz, Pablo Azero Alcocer</i>	259
UN-LENCEP: A Controlled Language for Pre-conceptual Schema Specification <i>Carlos Mario Zapata Jaramillo, Alexander Gelbukh, Fernando Arango Isaza</i>	269
Una Arquitectura Flexible para la Administración de Reputación en Sistemas Multiagentes <i>Víctor Daniel Podberzski, Jorge Salvador Ierache, Ramón García Martínez</i>	277
INDIGO: una Propuesta de Planificación en Inteligencia Artificial para la Composición de Servicios Web Semánticos <i>Jaime Alberto Guzmán Luna, Demetrio Arturo Ovalle Carranza</i>	287
Arquitecturas para Gestión de Conversaciones B2B Basadas en Conocimiento <i>José Luis López-Cuadrado, Juan Miguel Gómez, Angel García Crespo, Belén Ruiz Mezcua, Israel González-Carrasco, Giner Alor-Hernández</i>	297
Una propuesta para el tratamiento de brotes epidémicos y de otros fenómenos espacio-temporales <i>Francisco Javier Moreno Arboleda</i>	305
Aplicação de um Checklist de Pré-Teste <i>Odair Jacinto da Silva, Adalberto Nobiato Crespo, Mario Jino</i>	313
Monitoring and Control of an Event-based Middleware <i>Oscar González, Nicolás López, and Rubby Casallas</i>	319

Sección VI: Ingeniería del Conocimiento

Planificación de tareas en un Sistema de Computación Grid, para aplicaciones paralelas y/o secuenciales <i>Francisco Guevara, Andrés Marín, Elisa Heymann</i>	335
Sistema de E-Learning basado en Agentes de Software, Sistemas Tutoriales Inteligentes, Ambientes Colaborativos de Aprendizaje <i>Andrés F. Hoyos P., Jovani A. Jiménez B. y Demetrio A. Ovalle C.</i>	343
Adaptación, Recuperación y Almacenamiento de Contenidos Educativos Digitales para un Sistema Tutorial Inteligente <i>Catalina Salazar Ortiz, Jovani A. Jiménez B., Demetrio A. Ovalle C.</i>	353

Sección VII: Mantenimiento y Reúso de Software

A Portlet-Based Service-Oriented Architecture for a second generation portal <i>Giner Alor-Hernández, Ruben Posada-Gomez, Juan Miguel Gomez, Ana Ma. Chávez-Trejo</i>	363
Robots y Juguetes Autónomos una Oportunidad en el Contexto de las Nuevas Tecnologías en Educación <i>J. Ierache, M. Bruno, M Dittler, N. Mazza</i>	371
Un enfoque ADM para la Reingeniería de Bases de Datos Relacionales hacia Servicios Web <i>García-Rodríguez de Guzmán, I., Polo, M., Piattini, M., Pérez, R. I Alarcos Research Group.</i>	381

Anexo

Un enfoque pragmático para la mejora de procesos software en las PyMEs <i>Hanna Oktaba, Mario Piattini, Francisco J. Pino, Félix García, Tomás Martínez, Claudia Alquicira, Francisco Ruiz</i>	395
Experiences with the use of MERODE in the development of a Web Based Application <i>Karina Chong, Verónica Macías, Monique Snoeck</i>	421
Esquema de Clasificación de Defectos para la mejora del Proceso Software en una Empresa de Telecomunicaciones de Ecuador. <i>Fernando Uyaguay U.</i>	427

Meta²Relational: Herramienta para la Gestión de Modelos de Procesos Software

Tomás Martínez-Ruiz, Félix García, 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

Resumen

En la última década las organizaciones desarrolladoras de software se han concienciado de la importancia que tiene la mejora de sus procesos software para desarrollar productos de calidad. Para llevar a cabo una mejora de sus procesos, la organización debe apoyarse en un Modelo de Referencia de Procesos, un Modelo de Evaluación y un Modelo de Mejora, y en base a estos iniciar el ciclo de mejora. Éste conlleva la realización de continuas valoraciones, para determinar los puntos fuertes y débiles, en las que hay que chequear todas las actividades realizadas y/o productos generados. Dicha tarea puede resultar muy pesada si se realiza a mano, pero a la hora de realizar este trabajo de forma automática es conveniente que toda la información de la empresa esté organizada de forma coherente con la estructura del modelo de procesos elegido. Para facilitar la gestión y de esta estructura (esquema de base de datos) de forma estándar y adaptable a cambios, se presenta la herramienta Meta²Relational. Esta herramienta permite generar el esquema equivalente al metamodelo del Modelo de Referencia de Procesos utilizado basándose en transformaciones QVT. Además permite extraer toda la información de la organización en un archivo XMI con la estructura del metamodelo.

Palabras Claves: Modelo de Referencia de Procesos Software, evaluación de proceso software, Mejora de Procesos Software, COMPETISOFT, metamodelo, transformación, QVT.

Abstract

In the last decade, software development organizations have been become aware of the importance of software process improvement to develop high quality products. To carry out the improvement of their processes, organizations must be supported on a Process Reference Model, an Assessment Model and an Improvement Model, and, based on them to initiate the improvement cycle. This cycle involves continually the carrying out of assessments, in order to determine the strengths and weaknesses, which implies to check the performed activities and/or generated products. This task turns out to be very tedious if is handmade, but when this task should be done automatically, it is convenient for the business information to be represented coherently according to the structure of the selected process model. To facilitate the creation of this structure (database schema), Meta²Relational is presented. This tool allows us to generate the schema equivalent to the metamodelo of the used Process Reference Model based on QVT transformations. Furthermore, the tool facilitates the extraction of all the enterprise information into a XML file structured according to the metamodel.

1. Introducción

La calidad es un factor determinante para cualquiera de los productos que se ofrecen en el mercado, incluidos los productos software. Además, debido a que las soluciones software están presentes en la mayoría de los sistemas utilizados a diario, ésta resulta imprescindible para garantizar que las características del producto satisfacen las necesidades del cliente, y que su funcionamiento está libre de deficiencias [35].

Para asegurar la calidad de un producto software se debe verificar que el proceso que crea dicho producto sea “capaz”, es decir, que este compuesto

por un conjunto de actividades coherentes y ordenadas entre sí, que incluyan las mejores prácticas disponibles. El nivel de madurez de una organización se puede determinar a partir del nivel de capacidad de los procesos que componen ésta, y representa el grado en que se gestiona la calidad dentro de la organización.

En la última década, las organizaciones software se han sensibilizado en incrementar su nivel de madurez mediante la mejora de sus procesos software como base para garantizar la calidad de los productos que desarrollan (SPI) [28]. Como soporte a estas iniciativas de mejora, diversos organismos han centrado sus esfuerzos en el desarrollo de

Modelos de Referencia de Procesos, Modelos de Evaluación de Procesos y/o Modelos de Mejora de Procesos. Estas iniciativas tienen como objetivo proporcionar a las organizaciones el conjunto de procesos que agrupan las mejores prácticas en base a las cuales desarrollar un software de calidad, así como los mecanismos necesarios para evaluar la implantación del modelo y mejorar la ejecución de estos procesos dentro de la organización.

Los modelos desarrollados se pueden dividir en dos grandes familias, los orientados a grandes organizaciones, y los diseñados para PyMEs productoras de software [36]. De los primeros destacan CMM [37], CMMI [38-41], SCE [1], CBA-IPI [2], IDEAL [24], propuestos por el Software Engineering Institute (SEI) o los estándares ISO 15504[15-19, 21], la familia ISO 9000 [20] e ISO 12207 [21]. De entre las iniciativas orientadas a PyMEs destacan, MoProSoft [25], EvalProSoft [26], creados ambos por la Secretaría de Economía de México, Agile SPI [11, 12] desarrollado por el proyecto SIMEP-SW financiado por Colciencias y la Universidad del Cauca, o el Marco Metodológico que se está llevando a cabo dentro del proyecto COMPETISOFT [27-30], financiado por CYTED.

Para realizar un ciclo de mejora es necesario realizar continuas evaluaciones de los procesos en que se divide la organización, comprobando todas las tareas ejecutadas y productos generados. Con el objetivo de facilitar las evaluaciones sería conveniente disponer de herramientas que faciliten la estructuración de esta información según el modelo de procesos que sigue la organización, facilitando los cambios y el intercambio estándar de los modelos de referencia y evaluación, así como los resultados de la misma.

Una vez creada la estructura de almacenamiento, se podrá guardar en ella la información de todos los activos de la organización. Estos datos se podrán consultar para conocer el estado de los activos y evitar tener que preguntarlo al usuario encargado de hacer la evaluación, con el consecuente ahorro de tiempo y esfuerzo.

En este artículo se presenta la herramienta Meta²Relational que permite generar la estructura necesaria para almacenar toda la información de la empresa de forma coherente con el modelo de procesos implementado por ésta. La herramienta importa los modelos de procesos en XMI y mediante transformaciones QVT se genera el esquema relacional correspondiente que permite almacenar la información de todo el proceso y su evaluación.

Además de la presente introducción, el artículo presenta en la sección 2 el estado del arte y los trabajos relacionados. En la sección 3 se describen con más detalle las características de la herramienta

desarrollada. La sección 4 muestra el Marco Metodológico de COMPETISOFT y la aplicación de la herramienta en este caso práctico. En la sección 6 se muestran las conclusiones y los trabajos futuros.

2. Trabajos relacionados

Tal y como se ha comentado en la introducción, las organizaciones dedicadas tanto al desarrollo como al mantenimiento de productos software están especialmente motivadas hacia la mejora de sus procesos. Como soporte a las iniciativas de mejora, se han desarrollado diversas herramientas de gestión y valoración de procesos que facilitan el manejo del modelo de procesos para el que han sido diseñadas.

Así pues, se puede destacar para el marco CMM/CMMI la herramienta CMM-Quest, [7] esta permite analizar los procesos de las categorías más importantes de la organización y determinar sus puntos fuertes y débiles Mediante la asignación de valores a las distintas prácticas definidas en el modelo, CMM-Quest genera unos resultados que muestra en forma de diagramas que corresponden con los perfiles de capacidad. También permite crear un informe web con toda la información producida.

Existen otras aplicaciones que ayudan a realizar la valoración usando CMM o CMMI, tales como: CMMI Browser [42], que realiza evaluaciones en el marco CMMI continuo enfocando la evaluación en un conjunto de áreas de procesos y seleccionando el nivel de capacidad deseado; SW-CMM Intern Maturity Toolkit [42], herramienta basada en un libro Excel, y CMMI Self-Assessment Tool [42], ambas permiten realizar auto-evaluaciones dentro de una organización a partir de diferentes cuestionarios.

Para ISO/IEC 15504 (SPICE) se dispone de herramientas como SPICE 1-2-1 [9], que proporciona una funcionalidad parecida a CMMI-Quest, SPICE-Lite [10], es una versión posterior y mejorada de la anterior que aporta una funcionalidad similar. Los resultados de ambas herramientas encajan dentro de los modelos ISO 9001 [14] y BOOTSTRAP [22].

MyBusinessQuest [8] y ErfolgsQuest [6] son también herramientas para evaluar bajo el modelo ISO/IEC 15504, la funcionalidad de ambas es parecida a la de las anteriores pero usando procesos genéricos y centrándose en los niveles 1 a 3 del modelo, respectivamente.

Para el estándar ISO/IEC 9000:2000 [13] está disponible la herramienta Assess&Act que presenta una funcionalidad similar a la de las herramientas vistas para CMMI o ISO 15504. Existen distintas herramientas más para otros tantos modelos de evaluación menos utilizados, por ejemplo, Bootcheck [3] para el modelo Bootstrap [22].

Todas las herramientas citadas anteriormente permiten realizar la evaluación basándose única y exclusivamente en el marco sobre el que se han diseñado. Además estas herramientas no son gratuitas y sólo se puede disponer de ciertas versiones durante un periodo limitado, lo cual no es asumible por una organización con restricciones de presupuesto que quiera iniciar un ciclo de mejora formal.

Con el fin de paliar las necesidades anteriores se ha desarrollado un entorno flexible de evaluación de procesos en PyMES. La herramienta clave en este entorno para la gestión flexible de la capa de almacenamiento y para la interoperabilidad en el intercambio de evaluaciones de procesos es Meta²Relational, que se describe con mayor detalle en la siguiente sección.

3. Meta²Relational

Meta²Relational es una herramienta diseñada para formar parte del Entorno Flexible para la Evaluación de la Capacidad del Software (Figura 1).

Meta²Relational permite al usuario manejar los modelos de referencia y evaluación, y crear en la base de datos el esquema necesario para almacenar la información de la empresa siguiendo la estructura

dictada por el modelo elegido. A partir de la estructura de datos ya creada, EvalModel se encarga de rellenar los datos en el esquema ya creado e iniciar la evaluación según un modelo determinado.

3.1. EvalModel

EvalModel permite realizar evaluaciones atendiendo a diferentes Modelos de Evaluación (Figura 1). Consta de un módulo específico para cada uno de estos modelos y se encarga de evaluar a la organización basándose en los datos de está que se hayan en la base de datos. Para asegurar que los datos están almacenados de forma coherente con el Modelo de Referencia utilizado, el esquema de la base de datos debe crearse utilizando Meta²Relational.

EvalModel genera un informe completo de la evaluación, que incluye gráficos legibles e ilustrativos. Estos resultados se representan en un formato común, para facilitar la comparación de las evaluaciones entre empresas y evaluaciones realizadas a lo largo del tiempo. La herramienta también ofrece la posibilidad de definir e incorporar nuevos modelos de evaluación.

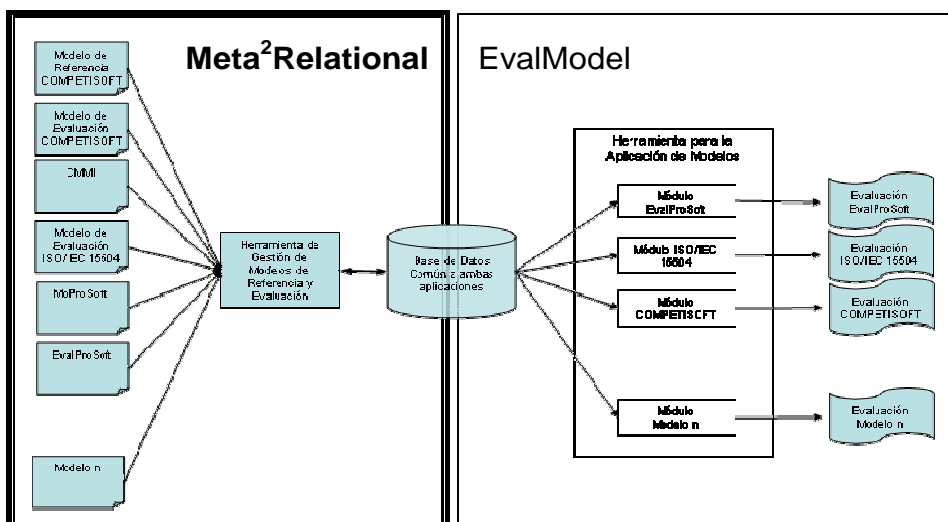


Figura 1. Relación entre las dos partes de Entorno Flexible para la Evaluación de la Capacidad del Software

Meta²Relational debe permitir el manejo de metamodelos, modelos y esquemas y la transformación entre ellos para abstraer al usuario de esta tarea. Este objetivo se descompone en otros, que a su vez componen la funcionalidad de la herramienta:

- El funcionamiento de la herramienta está basado en estándares.

- Transformación de un metamodelo M2 en esquema. Extracción del metamodelo M2 de un esquema, y su representación en XMI. Ambas transformaciones pueden ser personalizadas.
- Instanciación de un modelo concreto (M1) a partir del metamodelo (M2) expresado en el esquema, y de los datos contenidos en la base de datos, y su representación en XMI.

- Posee un interfaz configurable en varios idiomas.
La herramienta se ha creado en Java, utilizando Eclipse 3.2 como entorno de desarrollo.

3.2. Diseño de Meta²Relational

En el diseño de la herramienta se han aplicado los patrones *Observador*, *Facade*, *Experto* y *Buider* [4, 23]. Además se ha realizado un diseño modular, lo que proporciona un aumento de la cohesión y disminución del acoplamiento (Figura 2). Estos módulos (implementados como clases) se han agrupado en paquetes, atendiendo a la similitud de comportamiento.

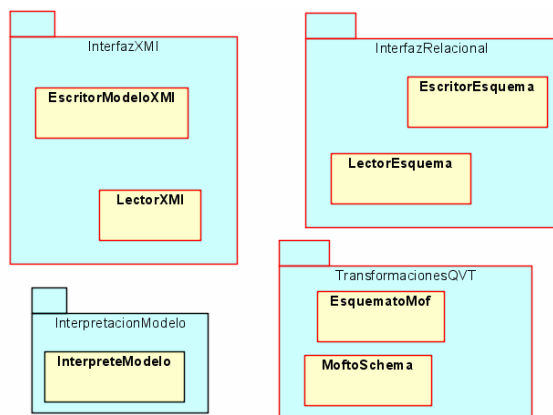


Figura 2. Módulos de la herramienta en la capa de dominio y su agrupación

Los distintos módulos que componen la herramienta realizan una tarea específica (debido a que implementan el patrón experto), lo que simplifica su prueba y facilita su reutilización. De esta manera, la funcionalidad de la herramienta se logra componiendo la aportada por cada uno de los módulos requeridos.

Meta²Relational basa su funcionamiento en un conjunto de estándares: MDA (Model Driven Architecture) [5], es la filosofía en la que se basa; MOF (Meta Object Facility) [32, 33], es el marco sobre el que gestionar los modelos en distintos niveles de abstracción; XMI (XML Model Interchange) [31], permite representar la información de modo uniforme y que en este caso permite a la herramienta compartir datos de forma estándar, promoviendo la interoperabilidad con otras herramientas y entornos; QVT (Query View Translation) [34], es el estándar en el que se han especificado de manera formal las transformaciones entre los dos modelos, MOF y Relacional. Éstas establecen de forma precisa la correspondencia entre los elementos de ambos modelos.

Dado que en la actualidad no existe un motor de QVT lo suficientemente maduro, se optó por implementar provisionalmente las transformaciones

en Java utilizando como pseudocódigo la especificación QVT de dichas transformaciones.

3.3. Interfaz de Meta²Relational

Meta²Relational provee interfaces sencillas que proporcionan ayuda al usuario. La herramienta es configurable mediante un archivo de configuración XML en el que se incluyen los parámetros para su funcionamiento y donde se especifican los idiomas.

Para crear el esquema de la base de datos donde se almacenará la información, la herramienta pide el metamodelo codificado en formato XMI, así como la identificación del servidor SQLServer donde se creará la base de datos una vez realizada la transformación. El metamodelo se puede crear usando la herramienta Rational Rose y exportándolo en formato XMI.

Mediante la selección de la opción *Experto*, la herramienta permite elegir al usuario cuáles de los atributos de cada entidad (clase) formarán parte de la clave primaria de la tabla equivalente, una vez convertidos a columnas.

La transformación para extraer el metamodelo inherente al esquema de una base de datos se realiza a partir del esquema de la misma. Se necesitan los datos del servidor donde se encuentra la base de datos y el nombre de ésta. Mediante un cuadro de dialogo, la herramienta pide la ubicación del archivo XMI a generar.

Marcando la opción *Experto*, se muestra un dialogo que permite especificar cuales de las clases que se generarán deben ser abstractas y permite definir diferentes opciones de las asociaciones detectadas, como son el nombre de ésta, los roles o las cardinalidades de las entidades relacionadas (Figura 3).

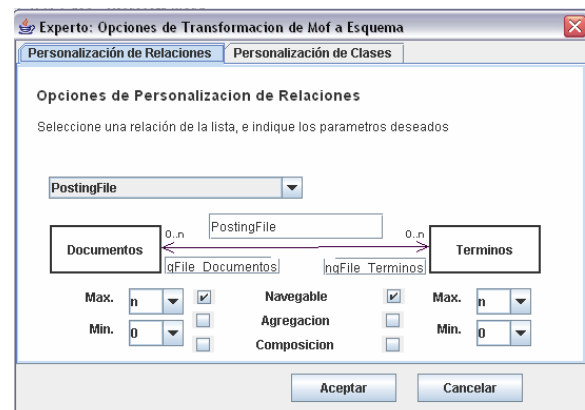


Figura 3. Pantalla para personalizar asociaciones en la transformación Esquema-Metamodelo

A la hora de crear un archivo XMI con la información contenida en la base de datos y de

acuerdo con el metamodelo de ésta, aparece una pantalla en la que se piden las referencias de la base de datos (servidor y nombre). También se puede indicar el archivo XMI que contiene el metamodelo que corresponde con es esquema de la base. Si dicho archivo no se especifica, el metamodelo se genera a partir del esquema.

Esta funcionalidad se extiende para interpretar y generar un archivo XMI con los resultados de evaluación de EvalModel, únicamente indicándole la referencia de la base de datos, dado que su metamodelo es conocido Meta²Relational. Asimismo, la herramienta permite añadir, eliminar y/o modificar la lista de servidores SQLServer a los que se conecta.

Los menús de que dispone la herramienta permiten al usuario cambiar el idioma, modificar el archivo de configuración, ver el log o lanzar las herramientas Rational Rose o el administrador de SQLServer, para ver los diagramas generados.

4. Caso de Estudio: COMPETISOFT

Para realizar el caso de estudio de la herramienta se ha utilizado el modelo de referencia de procesos definido en COMPETISOFT.

COMPETISOFT [27-30] es un proyecto iberoamericano en el que se está definiendo un Marco de Mejora de Procesos Software orientado a pequeñas organizaciones iberoamericanas. Este marco incluye los exitosos resultados y experiencias obtenidos por iniciativas semejantes en distintos países iberoamericanos, e incorpora las mejores prácticas contenidas en los modelos orientados para grandes organizaciones. Además, COMPETISOFT cuenta con la experiencia de investigadores, PyMÉS y organizaciones gubernamentales de los países involucrados en el proyecto (Figura 4).

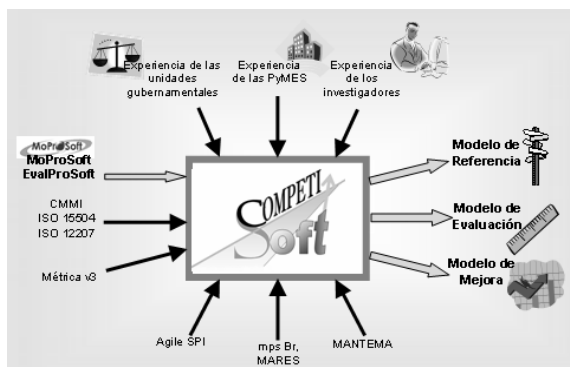


Figura 4. Enfoque general del Proyecto COMPETISOFT

Los procesos definidos por el modelo de procesos de COMPETISOFT se agrupan dentro de tres categorías (Figura 5). La primera es Alta Dirección,

que proporciona los lineamientos para el funcionamiento del resto de categorías y procesos y se retroalimenta con la información generada por estos, su único proceso es Gestión de Negocio.

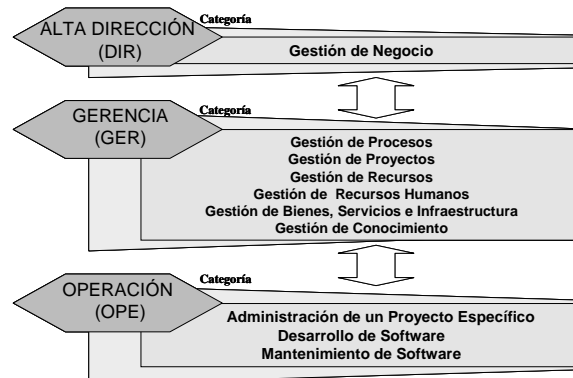


Figura 5. Categorías y procesos del Modelo de Procesos de COMPETISOFT

Gerencia es la segunda categoría, incluye las prácticas de gestión de procesos, proyectos y recursos y proporciona a la categoría de Operación todos los elementos necesarios para su funcionamiento, agrupa los procesos de Gestión de Proyectos, Procesos y Recursos, así como los de Gestión de Recursos Humanos y Ambiente de Trabajo, Gestión de Bienes, Servicios e Infraestructura y Gestión de Conocimiento.

La tercera categoría, Operación, aborda las prácticas de los proyectos de desarrollo y mantenimiento de software. Estas categorías se adaptan a los niveles de una organización.

Los procesos definidos dentro del modelo de procesos de COMPETISOFT se adaptan al patrón de procesos mostrado en la Figura 6. Los procesos de evaluación y mejora definidos en los modelos correspondientes del marco también se ajustan a este patrón, del que se pueden destacar las siguientes características:

- Todos los procesos tienen un identificador único,
- Los procesos identifican aquellos otros con los que se relacionan,
- Se definen indicadores a partir de los cuales comprobar el grado de consecución de los objetivos,
- Los procesos se dividen en actividades, y estas a su vez en Tareas. Cada tarea está marcada con un color que es sinónimo del nivel de capacidad a partir del cual se debe tener en cuenta ésta,
- Tanto el propósito del proceso como las entradas/salidas de cada actividad están claramente identificadas,
- Todos los productos se definen en función de su correspondiente plantilla,

- Todos los procesos tienen **Guías de Ajuste**, estas permiten adaptar los procesos genéricos a las necesidades de la organización.

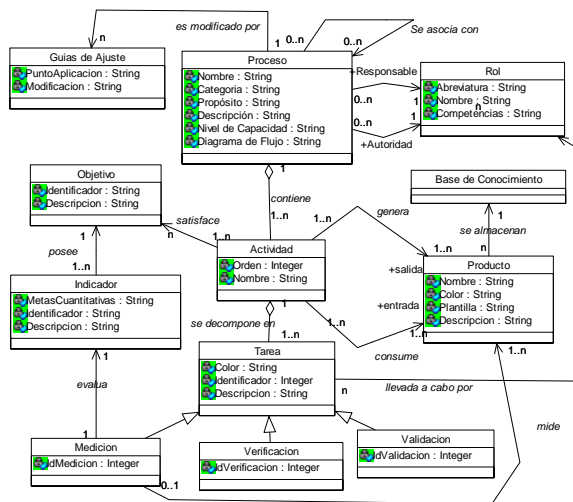


Figura 6. Patrón de Procesos de COMPETISOFT

4.1. Aplicación de COMPETISOFT

Para validar el funcionamiento de la herramienta, se ha ejecutado usando como metamodelo de entrada el correspondiente con el patrón de procesos el Modelo de Referencia de Procesos de COMPETISOFT descrito anteriormente (Figura 6).

El resultado que se ha obtenido se puede ver en la Figura 7.

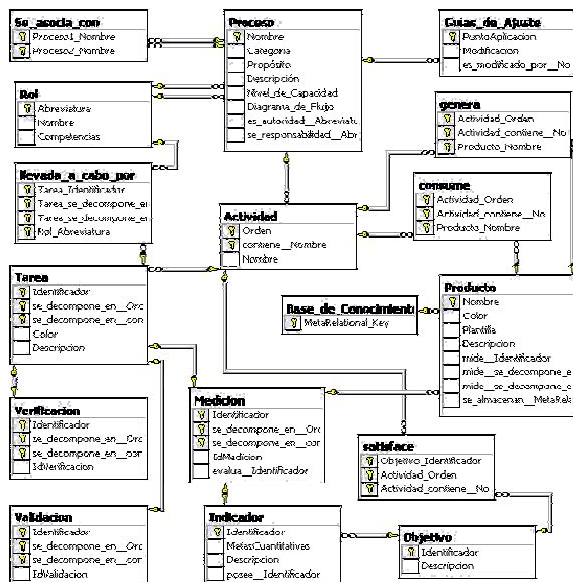


Figura 7. Esquema correspondiente al metamodelo del modelo de procesos de COMPETISOFT

La herramienta asimismo es capaz de realizar la transformación inversa, así pues, tomando como esquema el que se muestra en la Figura 7, generará el

metamodelo correspondiente, y que es equivalente al mostrado en la Figura 6. Este se puede ver en la Figura 8.

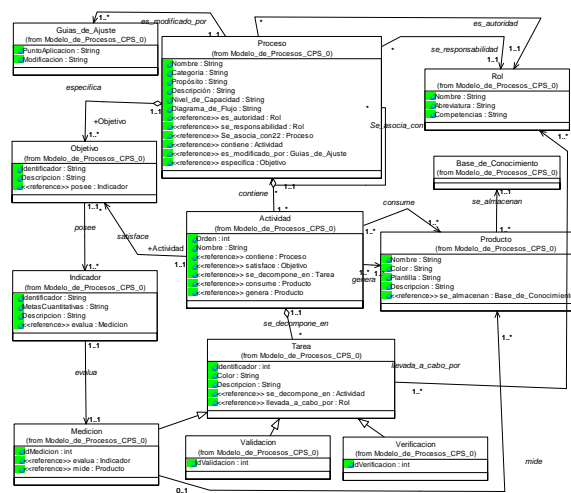


Figura 8. Metamodelo de COMPETISOFT generado a partir de su esquema

Debido a la pérdida de semántica en las transformaciones entre los dominios UML y Relacional, Meta²Relational incorpora la posibilidad de personalización, mediante la opción experto. Mediante la personalización de los diagramas resultantes, se garantiza que la aplicación de las dos transformaciones definidas es idempotente.

Una vez introducidos los datos dentro de la base de datos, se usó la opción de extraer la información almacenada en forma de fichero XMI (junto con el fichero de XMI schema XSD asociado). En la Figura 9 se muestra parte del archivo XMI generado.

5. Conclusiones y trabajo futuro

Este artículo ha presentado Meta²Relational, una herramienta para realizar transformaciones entre metamodelos y esquemas, basada en estándares como XMI, MOF o MDA, que permite crear la estructura necesaria por toda organización para almacenar la información de sus activos de manera coherente al modelo de referencia de procesos utilizado, con el propósito de facilitar las evaluaciones.

Además, dada su modularidad y que se ha implementado en Java, los distintos módulos de que se compone son fácilmente reutilizables. Al estar Meta²Relational basada en el uso de estándares de metamodelado, la herramienta puede ser acoplada fácilmente a otras, incluso no sólo herramientas específicas de valoración de procesos software, sino cualquier herramienta que requiera la definición dinámica y sencilla de una estructura adaptable donde almacenar la información que generen.

```

<Proceso Categoria="Gerencia"
  Descripción="El proceso de
  Gestión de negocio..."
...
  <Actividad Nombre="Planificación"
    Orden="1" consume="" genera=""
    satisfice=
      "@Objetivo_Identificador@1"
    xmi.id="@Actividad_Orden@1.0">
    <Tarea Color="amarillo"
      descripcion="Articular,
      documentar..."
      identificador="1"
      llevada_a_cabo_por=
        "@Rol_Abreviatura@GD"
      xmi.id=
        "@Tarea_Identificador@1.0"
    />
    <Tarea Color="amarillo"
      descripcion="Entender la
      situación"
      identificador="2"
      llevada_a_cabo_por=
        "@Rol_Abreviatura@RGN"
      xmi.id=
        "@Tarea_Identificador@2.0"
    />
    <Tarea Color="azul"
      descripcion="Validar
      el plan..."
      identificador="3"
      llevada_a_cabo_por=
        "@Rol_Abreviatura@RGN"
      xmi.id=
        "@Tarea_Identificador@11.0"
    />
  </Actividad>
  <Actividad nombre="Preparación
  para la Realización"
  orden="2"
... />
</Proceso>

```

Figura 9. Archivo XMI generado

Asimismo dado el auge que tiene la transmisión de información por la web codificada en XML, su capacidad para interpretar y exportar el contenido de una base de datos en formato XMI también es muy interesante. Esta funcionalidad permite la creación de un archivo XML que pueda ser usado como “paquete” a la hora de distribuir la información contenida en una base de datos, o parte de la misma, sin tener que enviar toda ella.

Como base sobre la que estructurar y diseñar las transformaciones entre los modelos MOF y el que representa una base de datos, se diseñaron las transformaciones QVT necesarias para llevar a cabo la transformación de modelos.

Una propuesta de trabajo futuro será integrar un motor de QVT y ejecutar en él las transformaciones definidas. La integración se realizaría de forma sencilla sustituyendo los módulos en que se han implementado las transformaciones por otros que implementen un patrón *Adapter* [23] y comuniquen la herramienta con el motor QVT.

Otra extensión a realizar consistirá en el diseño e implementación de un módulo con capacidad para

dibujar los gráficos necesarios y así permitir la visualización de los diagramas de forma autónoma, sin depender de otras herramientas como Rational Rose.

6. 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 y del proyecto COMPETISOFT 506AC0287 (Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria de Software de Iberoamérica) del programa CYTED (Ciencia y Tecnología para el Desarrollo).

7. Bibliografía

- [1]. Byrnes, P. and M. Philips, *Software Capability Evaluation Version 3.0. Method description*, in ESC-TR-96-002, CMU/SEI-96-TR-002, Editor. 2006.
- [2]. Dunaway, D. and S. Masters, *CMM-Based Appraisal for Internal Process Improvement (CBA-IPI). Version 1.2. Method description*. 2001.
- [3]. European Software Institute, *Bootcheck*.
- [4]. Gamma, E., et al., *Patrones de Diseño. Elementos de Software orientado a objetos reusable*. 2003, Madrid: Pearson Educación. 384.
- [5]. García, I., et al., *Análisis de Propuestas de MDA*. 2005: Ciudad Real. p. 80.
- [6]. HM&S GmbH, *Erfolgs Quest*.
- [7]. HM&S IT-Consulting GmbH, *CMM-Quest*.
- [8]. HM&S IT-Consulting GmbH, *My Business Quest*.
- [9]. HM&S IT-Consulting GmbH, *SPICE 1-2-1*.
- [10]. HM&S IT-Consulting GmbH, *SPICE Lite*.
- [11]. Hurtado, J., F. Pino, and J. Vidal, *Software Process Improvement Integral Model: Agile SPI. Technical Report SIMEP-SW-O&A-RT-6-V1.0*. 2005. 2006, Universidad del Cauca - Colciencias.: Popayán, Colombia.
- [12]. Hurtado, J., et al., *Agile SPI: Software Process Agile Improvement, A Colombia Approach to Software Process Improvement in Small Software Organizations*, in *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies*, M. Piattini, Oktaba, H., Editor. 2007, Idea Group Inc.: USA. p. In press.
- [13]. ISO, *Sistemas de Gestión de la Calidad. Fundamentos y Vocabulario*. 2000, AENOR: Madrid.
- [14]. ISO, *Sistemas de Gestión de la Calidad. Requisitos*. 2000, AENOR: Madrid.

- [15]. ISO, *ISO/IEC 12207:2002. Information technology - Software life cycle processes*. 2002, International Organization for Standardization: Geneva.
- [16]. ISO, *ISO/IEC 15504-2:2003/Cor.1:2004(E). Information technology - Process assessment - Part 2: Performing an assessment*. 2004, International Organization for Standardization: Geneva.
- [17]. ISO, *ISO/IEC 15504-3:2003/Cor.1:2004(E). Information technology - Process assessment - Part 3: Guidance on Performing an Assessment*. 2004, International Organization for Standardization: Geneva.
- [18]. ISO, *ISO/IEC 15504-4:2003/Cor.1:2004(E). Information technology - Process assessment - Part 4: Guidance on use for process improvement and process capability determination*. 2004, International Organization for Standardization: Geneva.
- [19]. ISO, *ISO/IEC 15504-5:2006(E). Information technology - Process assessment - Part 5: An exemplar Process Assessment Model*. 2006, International Organization for Standardization: Geneva.
- [20]. ISO_9001, *ISO 9001:2000. Quality management systems -Requirements*. 2000, International Organization for Standardization: Geneva.
- [21]. ISO_12207, *ISO/IEC 12207:2002/FDAM 2. Information technology - Software life cycle processes*. 2004, International Organization for Standardization: Geneva.
- [22]. Kuvaja, P., et al., *Software Process Assessment and Improvement: The BOOTSTRAP Approach*. 1994, Oxford, UK.: Blackwell Business Publishers.
- [23]. Larman, C., *UML y Patrones : introducción al análisis y diseño orientado a objetos*. 1999: Prentice Hall.
- [24]. McFeeley, R., *IDEAL: A Users Guide for Software Process Improvement, Handbook CMU/SEI-96-HB-001*. 1996, Software Engineering Institute, Carnegie Mellon University: Pittsburgh, USA.
- [25]. Oktaba, H., et al., *Modelo de Procesos para la Industria de Software - MoproSoft - Versión 1.3, Agosto de 2005. NMX-059/01-NYCE-2005*. 2005, Organismo nacional de normalización y evaluación de la conformidad - NYCE: Ciudad de México.
- [26]. Oktaba, H., et al., *Método de Evaluación de procesos para la industria de software - EvalProSoft - Versión 1.1, Marzo de 2004. NMX-I-006/(01 al 04)-NYCE-2004*. 2004, Organismo nacional de normalización y evaluación de la conformidad - NYCE: Ciudad de México.
- [27]. Oktaba, H., et al., *Software Process Improvement in Small Latin-American Organizations: COMPETISOFT Project*. IEEE Computer, 2007: p. In press.
- [28]. Oktaba, H., et al., *COMPETISOFT: A Improvement Strategy for Small Latin-American Software Organizations*, in *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies*, H. Oktaba and M. Piattini, Editors. 2007, Idea Group Inc.
- [29]. Oktaba, H., et al., *Mejora de Procesos Software para Pequeñas Organizaciones*, in *Fábricas de Software: Experiencias, Tecnología y Organización*, M. Piattini and J. Garzas, Editors. 2007, Ra-Ma.
- [30]. Oktaba, H., et al., *Un Enfoque Pragmático para la Mejora de Procesos Software en las PyMEs*, in *COURE: Circulo de Usuarios de Oracle*. 2007: Madrid. p. 1-27.
- [31]. OMG, *OMG XML Metadata Interchange (XMI)*. 2001, Object Management Group.
- [32]. OMG, *Meta Object Facility (MOF) Specification version 1.4*. 2002, Object Management Group.
- [33]. OMG, *Meta Object Facility (MOF) Specification version 2.0*. 2004, Object Management Group.
- [34]. OMG, *Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification*. 2005, Object Management Group.
- [35]. Piattini, M., F. Garcia, and I. Caballero, *Calidad de Sistemas Informáticos*. 2006, Madrid: Ra-Ma. 388.
- [36]. Pino, F., F. Garcia, and M. Piattini, *Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas*. Revista Española de Innovación, Calidad e Ingeniería del Software (REICIS), 2006. 2(1): p. 6-23.
- [37]. SEI, *The Capability Maturity Model: Guidelines for Improving Software Processes*. 1995, Software Engineering Institute.
- [38]. SEI, *Standard CMMI® Appraisal Method for Process Improvement (SCAMPI), Version 1.1: Method Definition Document (CMU/SEI-2001-HB-001)*. 2001, Software Engineering Institute (SEI): Pittsburgh.
- [39]. SEI, *Capability Maturity Model® Integration (CMMISM), Version 1.1 CMMISM for Systems Engineering, Software Engineering, Integrated Product and Process Development and Supplier Sourcing*. 2002.
- [40]. SEI, *Capability Maturity Model Integration (CMMI), Version 1.1 CMMI (CMMI-SE/SW/PPD/SS, V1.1) Staged Representation*. 2004.
- [41]. SEI, *CMMI for Development, Improving processes for better products*. 2006, CMMI Product Team.
- [42]. SPIPartners, *Supporting Tools for Software Process Improvement*.