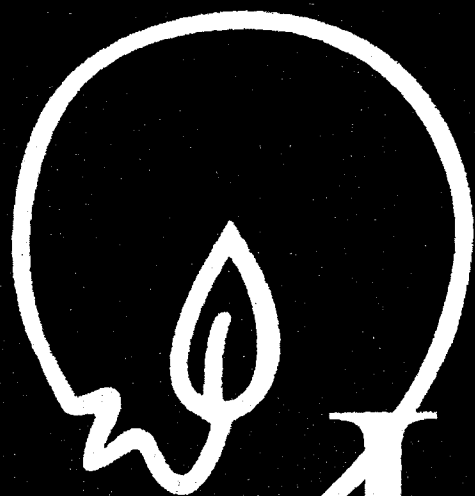


**6º WORKSHOP IBEROAMERICANO**  
de Ingeniería de Requisitos y Ambientes Software



# Ideas

2 0 0 3

30 de abril  
al 2 de mayo  
de 2003

Asunción - Paraguay

**Editores:**

Mario Piattini  
Luca Cernuzzi  
Francisco Ruíz

# Memorias



## **6° Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software**

Asunción, Paraguay  
del 30 de Abril al 2 de Mayo del 2003

# **Memorias**

*Editores*

Mario Piattini  
Luca Cernuzzi  
Francisco Ruíz

*Organización*

Proyecto WEST/CYTED  
Universidad Católica "Nuestra Señora de la Asunción"  
Universidad Autónoma de Asunción

ISBN: 84-96023-05-2

ISBN: 84-96023-05-2

Incluye ponencias en español, portugués e inglés.

A la cabeza de la portada: 6º Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software.

1. Ingeniería de Requisitos. 2. Tecnología de Procesos. 3. Orientación a Objetos. 4. Métodos Formales en Ingeniería del Software. 5. Ambientes y Herramientas de Desarrollo. 6. Calidad y Estimación de Software. 7. Desarrollo Basado en Componentes. 8. Ingeniería del Software para Sistemas Concurrentes y Distribuidos. 9. Bases de Datos. 10. Almacenes de Datos y Minería de Datos. 11. Bibliotecas Digitales. 12. Ingeniería Web.

Esta actividad fue patrocinada por: Proyecto VII.18. WEST (*Web-based Software Technology*, <http://www.dsic.upv.es/~west/>), CYTED (*Programa Iberoamericano de Ciencia Y Tecnología para El Desarrollo*, <http://www.cytel.org/Nueva.asp>), La Universidad Católica "Nuestra Señora de la Asunción", Sede Regional de Asunción, y la Universidad Autónoma de Asunción. Abril del año 2003.



## PRÓLOGO

Este volumen recoge los trabajos presentados en el 6º Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software (IDEAS 2003) celebrado en Asunción del 28 de abril al 2 de mayo de 2003.

Sin duda, IDEAS se ha consolidado como el foro de encuentro y discusión para el intercambio de experiencias y conocimientos entre los principales investigadores de la Ingeniería del Software en el ámbito Iberoamericano.

En esta edición se han recibido un total de 47 trabajos de gran calidad científica, lo que avala, por un lado, la creciente madurez del área y, por otro, el prestigio del Workshop, fruto del buen hacer de las ediciones anteriores. Cada trabajo ha sido evaluado por, al menos, dos miembros del comité de programa, no habiendo prácticamente evaluaciones divergentes. Con el fin de mantener la tradición de no desarrollar sesiones paralelas así como el nivel alcanzado, se decidió aceptar como comunicaciones largas un total de 23, y como comunicaciones cortas otras 12.

Además de las sesiones técnicas, esta edición ha apostado por las conferencias invitadas, que se han agrupado en tres sesiones, la primera dedicada a la tecnología Web que incluye las conferencias: "Modelado Conceptual en la Web (Oscar Pastor)", "La Tecnología .NET en el desarrollo de Software para la Web (Miguel Katrib)", y "Evaluación y Calidad de Sitios WEB" (Luis Olsina); la segunda a la Ingeniería del Software, con las conferencias: "Ingeniería de Requisitos" (Jaelson Castro), "Tecnología J2EE para el Desarrollo de Aplicaciones Distribuidas" (Raúl Monge), y "Procesos y Arquitecturas de Software" (Ernesto Pimentel); y la tercera a tendencias en ambientes Web, con las conferencias invitadas de Ramez Elmasri sobre los temas: "Base de Datos, Comercio Electrónico, Web, y XML: ¿Cómo están relacionados?" y "Extracción de Ontologías y Modelado Conceptual para Información en la Web".

Resulta imprescindible aprovechar estas líneas para agradecer los conferenciantes invitados y a la comunidad científica su apoyo a este Workshop, enviándonos sus trabajos. Asimismo, nuestra gratitud a todos los miembros del Comité de Programa por el esfuerzo realizado. Igualmente debemos reconocer el patrocinio del proyecto WEST-CYTED, de la Universidad Católica "Nuestra Señora de la Asunción", Sede Regional de Asunción y la Universidad Autónoma de Asunción. Por último, queremos agradecer a todos los miembros del Comité Organizador su entrega y dedicación que ha hecho posible la celebración de este Workshop que esperamos resulte del interés de todos los que en él participan.

*Juan de Dios Garbett*

Presidente del Comité Organizador-UAA

*Luca Cernuzzi*

Presidente del Comité Organizador-UCA

*Mario Piattini*

Presidente del Comité de Programa

## Comité de Programa

### *Presidencia:*

Mario Piattini (*U. Castilla-La Mancha, España*)

### *Miembros:*

José Bogarín (*U. Católica de Asunción*)  
Pere Botella (*U. Politécnica de Catalunya*)  
Nieves Brisaboa (*U. de La Coruña, España*)  
Antonio Brogi (*U. de Pisa, Italia*)  
Rodrigo Cardoso (*U. de los Andes, Colombia*)  
Jaelson Castro (*U. de Pernambuco, Brasil*)  
Carmen Costilla (*U. Politécnica de Madrid, España*)  
Luca Cernuzzi (*U. Católica de Asunción, Paraguay*)  
Joao Falcao e Cunha (*U. do Porto, Portugal*)  
Alexander Gelbukh (*IPN, México*)  
Luis Joyanes (*U. Pontificia de Salamanca, España*)  
Carlos Heuser (*UFRGS, Brasil*)  
Julio C. Leite (*PUC-Rio, Brasil*)  
Hanna Oktaba (*UNAM, México*)  
Miguel Katrib (*U. de La Habana, Cuba*)  
Esperanza Marcos (*U. Rey Juan Carlos, España*)  
Mauricio Marín (*U. de Magallanes, Chile*)  
Alfredo Matteo (*U. Central de Venezuela*)  
Raul Monge (*UTFSM, Valparaiso, Chile*)  
Luis Olsina (*U. La Pampa, Argentina*)  
Oscar Pastor (*U. Politécnica de Valencia, España*)  
Rodolfo Pazos (*CENIDET, México*)  
Francisco Pinheiro (*U. de Brasilia, Brasil*)  
Ernesto Pimentel (*U. de Málaga, España*)  
Isidro Ramos (*U. Politécnica de Valencia, España*)  
António Rito da Silva (*U. Técnica de Lisboa*)  
Gustavo Rossi (*U. Nacional La Plata, Argentina*)  
Francisco Ruiz (*U. Castilla-La Mancha, España*)  
Miguel Toró (*U. de Sevilla, España*)  
Ambrosio Toval (*U. de Murcia, España*)  
Antonio Vallecillo (*U. de Málaga, España*)  
Amparo Vilá (*U. de Granada, España*)  
Claudia Werner (*U. Federal de Rio, Brasil*)

## Comité de Organización

### *Presidencia:*

Luca Cernuzzi: *Universidad Católica "Nuestra Señora de la Asunción", Sede Regional Asunción*

Juan de Dios Garbett: *"Universidad Autónoma de Asunción"*

### *Miembros:*

*Universidad Autónoma de Asunción.*

Hugo Correa

Ivey Díaz

Flavio Jodorcovsky

Blanca de Báez

Fernando Garbett

*Universidad Católica "Nuestra Señora de la Asunción", Sede Regional de Asunción.*

Vicente González Ayala

Norma Morales

Magalí González.

# Indice

## Artículos Extendidos

Ingeniería de Requisitos en Aplicaciones para la Web: Un Estudio Comparativo <i>Escalona, M.J. &amp; Koch, N.</i> .....	2
Desarrollo de un Portal Web para un Departamento Universitario desde Modelado Conceptual <i>Valderas, P., Ruiz, M., Fons, J., Albert, M. &amp; Pelechano, V.</i> .....	15
Um Processo Auto-Documentável de Geração de Ontologias de Domínio para Dados Semi-Estruturados <i>Santi, S.M. &amp; Heuser, C.A.</i> .....	27
The Relevance of User Experience Requirements in Interface: Design – a Study of Internet Banking <i>Patrício, L., Falcão e Cunha, J. &amp; Fisk, R.P.</i> .....	39
Trazabilidad de Requisitos Adaptada a las Necesidades del Proyecto: Un Caso de Estudio Usando Alternativamente RUP y XP <i>Anaya, V. y Letelier, P.</i> .....	50
Metodología Basada en el Conocimiento para el Modelado del Negocio <i>Henao, M. &amp; Anaya, R.</i> .....	61
Integración del Metamodelado y la Medición para la Mejora de los Procesos Software <i>García, F., Ruiz, F., Cruz, J.A. &amp; Piattini, M.</i> .....	73
Modelización de las Competencias Humanas en el Proceso Software <i>Acuña, S.T. &amp; Juristo, N.</i> .....	85
Una Técnica de Descripción Formal para la Generación y Ejecución Automática de Casos de Prueba de Sistemas Orientados a Objetos <i>Jiménez, M.M., Polo, M. &amp; Piattini, M.</i> .....	97
Una Extensión de UML para Representar XML Schemas <i>Vela, B., &amp; Marcos, E.</i> .....	109
JOODE: Plataforma de Servicios Java para el Desarrollo de Aplicaciones Distribuidas y Paralelas <i>Fuentes, T., Katrib, M., Sierra, I. &amp; Del Valle, M.</i> .....	120
A Persistent Object Storage Service on Replicated Architectures <i>Armendariz J.E., Astrain J.J., Córdoba A., Villadangos J. &amp; González de Mendivil J.R.</i> .....	133
Diseño de Bases de Datos Difusas Modeladas con UML <i>Urrutia, A., Varas, M. &amp; Galindo, J.</i> .....	145
Análisis del Impacto del Proceso de Desarrollo en las Características de Calidad del Software <i>Mendoza, L., Pérez, M., Grimán, A. &amp; Ortega, M.</i> .....	156
Modelo de Calidad (MOSCA+) para Evaluar Software de Simulación de Eventos Discretos <i>Rincón, G., Pérez, M., Hernández, S. &amp; Álvarez, M.</i> .....	167
Técnicas Estadísticas para el Análisis de la Calidad de Sitios Web <i>Loranca, M.B. &amp; Olsina, L.</i> .....	178
Conectores: Una Solución para la Composición y Coordinación de Componentes <i>Katrib, M., Pastrana, J.L. &amp; Pimentel, E.</i> .....	190

Modelos de Interacción para la Coordinación de Componentes <i>Amaro, S., Pimentel, E. &amp; Roldán, A.M.</i> .....	202
Modelando con UML Arquitecturas Software Reflexivas desde la Aproximación Orientada a Aspectos: Un Caso de Estudio <i>Lorenzo, A., Peñarrubia, A., Carsí, J.A. &amp; Ramos, I.</i> .....	214
Aspectos de Diseño Arquitectural y Semántico para un Sistema Web de Catalogación de Métricas <i>Martín, M.A., Molina, H., Papa, F., Fons, J., Pastor, O. &amp; Olsina, L.</i> .....	224
Hacia un Catálogo de Actividades para el Desarrollo de Sitios y Aplicaciones Web <i>Esteban, N. &amp; Olsina, L.</i> .....	237
Construcción de un Modelo Borroso de Predicción del Tiempo de Mantenimiento de Diagramas de Clases UML <i>Genero, M., Romero, F., Olivas, J.A. &amp; Piattini, M.</i> .....	249
Vision: um método para elicitação e análise de requisitos orientado a viewpoints com UML <i>Coutinho, P. &amp; Araújo, J.</i> .....	261
<b>Artículos Breves</b>	
Problem Frames Application on Finite Element Method Simulators <i>Lencastre, M., Castro, J., Santos, F. &amp; Araújo, J.</i> .....	274
Generación Automática de Wrappers para el Acceso y Manipulación de Datos en XML <i>Moltó, G. &amp; Carsí, J.A.</i> .....	280
Extendiendo "UML eXchange Format" (UXF) <i>Beeck, J.C., Montoya, J. &amp; Pari, P.A.</i> .....	286
Arquitectura de un Sistema de Comercio Electrónico para Pequeñas y Medianas Empresas de México <i>Pérez, J., Pazos, R., Baeza, C. &amp; Bravo, M.</i> .....	292
Propuesta Metodológica para la Validación Integral, Ágil y Sistemática de Aplicaciones Web <i>Perallos, A. &amp; Cortázar, R.</i> .....	298
Acerca del Diseño de Aplicaciones "E-Governance" <i>González, M., Cernuzzi, L. &amp; Pastor, O.</i> .....	304
Aplicación de una Estructura Cooperativa a un Asistente para Consultas de Base de Datos <i>Chávez, S.</i> .....	310
Optimización de Consultas en Bases de Datos Relacionales <i>Pazos, R.A., Martínez, &amp; J.A.González, J.J.</i> .....	316
Ssquirel: un lenguaje de consulta para bases de datos semiestructurados <i>García, E.A., López, A. &amp; López, S.</i> .....	322
• Ingeniería de Requisitos de Calidad <i>Chirinos, L., Losavio, F., &amp; Matteo, A.</i> .....	328
Recovery in Multi-Threaded Distributed Systems <i>Hernández, C., Pérez, &amp; F. Peña, J.M.</i> .....	334

## Integración del Metamodelado y la Medición para la Mejora de los Procesos Software

Félix García, Francisco Ruiz, José Antonio Cruz, Mario Piattini  
Grupo Alarcos

Escuela Superior de Informática  
Paseo de la Universidad, 4  
13071 Ciudad Real (España)

{Felix.Garcia, Francisco.Ruiz, Mario.Piattini}@uclm.es, jacruz@proyectos.inf-cr.uclm.es

### Resumen

*La mejora del proceso software se ha convertido en uno de los objetivos estratégicos fundamentales en las organizaciones a la hora de promover la mejora de la calidad de sus productos. Para poder implantar un plan efectivo de mejora es muy importante establecer un buen marco de trabajo que permita que cada organización comprenda y defina de forma efectiva los procesos que lleva a cabo, y que le permita realizar adecuadamente la evaluación de sus procesos en base a la medición de los mismos. En este artículo se describe un marco de trabajo en el que se integra el modelado y la medición de los procesos como elementos clave para promover la mejora de los mismos. El marco de trabajo está basado en la arquitectura conceptual del estándar MOF para la definición de metadatos y sus relaciones en cuatro niveles de abstracción. Con el fin de dar soporte a la medición integrada se ha desarrollado GenMETRIC, herramienta extensible de cálculo, gestión y visualización de métricas software.*

### 1. Introducción.

La investigación del proceso software ha adquirido un gran auge en los últimos tiempos debido fundamentalmente a la creciente complejidad de los sistemas software. Las aplicaciones software son productos muy complejos y por lo tanto, difíciles de desarrollar y mantener. Cualquier organización dedicada a la producción y mantenimiento del software, a la hora de desarrollar su actividad sigue un proceso determinado. El proceso software es por tanto un proceso, pero con características especiales derivadas de la especial complejidad de los productos software obtenidos. De hecho, la calidad final del producto software está muy directamente relacionada con la forma en que se desarrolla y mantiene, y ello ha motivado que las organizaciones dedicadas al desarrollo y mantenimiento del software se preocupen cada vez más de la mejora de los procesos a la hora de promover una mejora en la calidad de sus productos. Con tal fin, han surgido diversas iniciativas que establecen un marco de referencia para que las organizaciones mejoren sus procesos. Entre estas iniciativas cabe destacar CMM[25], CMMi [26], la norma ISO 15504 [10] y dada su importancia, la mejora ha sido incorporada a la nueva familia de normas ISO 9000:2000 [12] [13], que promueven la adopción de un enfoque basado en procesos cuando se desarrolla, implementa y mejora un sistema de gestión de la calidad.

De entre las iniciativas de mejora enumeradas anteriormente, CMMi (Capability Maturity Model Integration) destaca por su especial importancia. El objetivo fundamental que se persigue con CMMi es la integración de la mejora de los procesos y sistemas software. CMMi utiliza tres modelos tradicionales que son CMM-sw (CMM para el software), el modelo de Ingeniería de Sistemas EIA/IS 731, y el modelo de desarrollo integrado del producto y proceso y los incluye en un modelo integrado de madurez de la capacidad. Dentro del contexto de CMMi, estos modelos proporcionan un conjunto común de requerimientos de proceso que conducen a la utilización de las mejores prácticas y conocimientos dentro de una organización dedicada al desarrollo y/o mantenimiento del software. Para alcanzar los objetivos de cada nivel de madurez, la organización debe entender, controlar y mejorar de forma continua sus procesos. Como consecuencia de tener

procesos efectivos y eficientes en una organización se obtienen productos de alta calidad que satisfacen plenamente las necesidades del cliente y de la organización.

Con el fin de obtener productos software de calidad es necesario llevar a cabo una buena gestión de los procesos software. La gestión exitosa del proceso software supone que los productos y servicios producidos por los procesos son conformes a los requisitos tanto de los clientes externos como internos a la organización, y a sus objetivos de negocio. Para poder llevar a cabo dicha gestión es necesario asumir cuatro responsabilidades clave que son [5]: Definición, Medición, Control y Mejora del proceso. Teniendo en cuenta estas responsabilidades, para poder promover la mejora de los procesos es muy importante considerar la gestión integrada de los siguientes aspectos:

- **Modelado de Procesos.** Dada la especial complejidad de los procesos software, derivada de la gran diversidad de elementos que hay que considerar a la hora de gestionar los mismos, es necesario realizar un proceso de definición de los procesos software de forma efectiva.
- **Evaluación de Procesos.** Para poder promover la mejora de los procesos software, es muy importante previamente establecer un marco de evaluación (con el fin de conocer sus puntos fuertes y débiles). Para ello es necesario proporcionar un marco efectivo para la medición de los procesos y productos software en una organización.

La integración de estas dos áreas es un factor fundamental para que una organización alcance un alto grado de madurez en sus procesos tal como identifica CMMi. Por ello, es imprescindible comprender bien los procesos, para lo cual es necesario una definición de los mismos, y mejorarlos, para lo cual es importante realizar previamente un proceso de medición efectiva.

Desde el punto de vista del modelado de procesos, a la hora de gestionarlos es necesario conocer en primer lugar cuáles son los elementos involucrados. El proceso software se puede definir como el conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, empaquetar y mantener un producto software [3]. Existe una gran cantidad de elementos relacionados con el proceso software que se pueden agrupar en las categorías [6]: Tecnología de Desarrollo Software, relacionado con el soporte tecnológico, en forma de herramientas, infraestructuras y entornos; Métodos y Técnicas de Desarrollo Software, que constituyen líneas guía sobre cómo se deben hacer las cosas: uso de la tecnología y realización de las actividades; Comportamiento Organizacional, relacionado con los recursos humanos que llevan a cabo los procesos software y que tienen que estar coordinados y deben gestionarse desde una eficiente estructura organizacional; y Economía y Marketing, relacionado con la gestión de proyectos, debido a que el producto software final debe cumplir con unos plazos y costes determinados y debe de satisfacer las necesidades del cliente al que va destinado.

Dada la diversidad de elementos que hay que tener en cuenta cuando se habla de un proceso software, la definición y gestión de los procesos software en una organización no es una tarea trivial. Uno de los elementos básicos para la gestión exitosa de los procesos software es su correcta definición, basada en la noción de modelo de proceso software, sobre el que han surgido diversas iniciativas. El modelado de los procesos software se ha convertido en una solución muy aceptada para tratar su inherente complejidad. En la literatura se pueden encontrar diversos lenguajes y formalismos de modelado, conocidos como "Lenguajes de Modelado de Procesos" (LMP), que tienen como objetivo representar de una forma precisa y no ambigua, los diferentes elementos relacionados con un proceso software. En general, en un proceso software se pueden identificar los siguientes elementos, que son conceptos generales (aunque con notaciones y términos distintos) en los diferentes LMPs [3]: Actividad, Producto, Recurso y Organizaciones y Roles.

Ante la diversidad de propuestas de modelado de procesos existentes, se hace necesario proporcionar un metamodelo de procesos que sea la referencia común y que incluya todos los aspectos necesarios para definir con la mayor semántica posible la forma mediante la que se desarrolla y mantiene el software. Con tal fin, el Object Management Group recientemente ha propuesto el metamodelo SPem (Software Process Engineering Metamodel Specification) [20], que constituye un lenguaje para la creación de modelos de procesos concretos en una organización. Este lenguaje queda encuadrado en el marco conceptual de la arquitectura de cuatro



niveles de OMG basada en el estándar MOF [19], y que permite gestionar de forma efectiva los diferentes conceptos relacionados con los procesos software en diferentes niveles de abstracción.

El otro aspecto clave a considerar es la importancia de incorporar los aspectos de medición con la definición de un metamodelo de la medición que constituya la referencia común en las organizaciones que quieren mejorar sus procesos. En esta línea cabe destacar el reciente trabajo de Olsina et al. [17] en el que se propone un modelo conceptual de las métricas donde se representan todos los elementos y relaciones necesarias para definir y explotar información de distintas métricas de atributos para diferentes dominios software.

En este artículo se propone un marco conceptual basado en la arquitectura de cuatro niveles de OMG para la evaluación y mejora del proceso software mediante la integración del modelado del proceso y de la medida. En primer lugar se da una visión general de la arquitectura. En el apartado 3 se describe el metamodelo SPEM para la definición de modelos de procesos. En el siguiente apartado se describe un metamodelo genérico de la medida, que ha sido definido e incorporado a la arquitectura conceptual, con el fin de establecer la referencia necesaria a la medición integrada en una organización. En el apartado 5 se describe GenMetric, herramienta extensible que se ha desarrollado para dar soporte al proceso de medición integrada en una organización. Finalmente se presentan las conclusiones y los trabajos relacionados.

2. Marco Conceptual para la Mejora Integrada del Proceso Software.

Para que una organización pueda realizar una gestión integrada de sus procesos software es muy importante que establezca una base rigurosa para:

- La definición de sus modelos de procesos con una terminología única y con una semántica precisa y bien definida.
- La gestión integrada de la medición en la organización mediante un metamodelo de la medida que sea el marco de referencia para la creación de modelos concretos de medida (medida de la base de datos, medidas de los productos de trabajo resultado del análisis o diseño, medida de los modelos de procesos, etc...).

Con el fin de integrar estos dos aspectos tan importantes en un proceso software se he definido una arquitectura conceptual de cuatro niveles de abstracción. Esta arquitectura está basada en el estándar MOF (Meta Object Facility) para metamodelado basado en la tecnología de objetos [19] propuesto por el Object Management Group (OMG). El objetivo de MOF es la especificación y gestión de metadatos en diferentes niveles de abstracción. MOF describe un lenguaje abstracto de modelado (basado en el núcleo de UML). En la siguiente tabla se muestra la arquitectura conceptual del estándar MOF y su aplicación al marco de trabajo propuesto para la mejora del proceso software:

Tabla 1. Niveles conceptuales en MOF y su aplicación para la mejora integrada

Nivel	MOF	Aplicación
M3	Modelo MOF (Meta-Metamodelo)	Modelo MOF
M2	Meta-modelo	Software Process Engineering Metamodel (SPEM) Metamodelo Genérico de la Medida (ISO 15939)
M1	Modelo	Modelos Concretos de Procesos Modelos Concretos de Medida
M0	Datos	Instancias de Modelos de Procesos (proyectos concretos del mundo real) Instancias de Modelos de Medida (resultados de la aplicación del modelo de medida)

En el nivel inferior de la arquitectura, M0, se encuentran los resultados de:

- La aplicación de un modelo de procesos, como por ejemplo, un modelo para la evaluación y mejora [7], o un modelo para el mantenimiento [23] a un proyecto software concreto. A este nivel de la arquitectura se registrarían los resultados de la ejecución de un modelo de procesos concreto.
- La aplicación de un proceso de medición. En este nivel se registran los valores obtenidos tras la aplicación de un modelo de medición concreto, como por ejemplo, los valores de la medición de una base de datos relacional, o los valores de la medición de diagramas de clases UML.

Los datos gestionados en el nivel M0 son instancias de los datos representados en el nivel inmediatamente superior M1. En este nivel, según la arquitectura conceptual propuesta se incluirían modelos concretos para la definición del proceso software, y modelos concretos para su medición. Desde el punto de vista de la definición, en este nivel la organización incluiría sus modelos de procesos, como por ejemplo, el modelo de proceso de desarrollo, de mantenimiento, de evaluación y mejora seguido en la misma, etc... Desde el punto de vista de la medición, en este nivel se incluirían los modelos concretos de medida usados en la organización. Por ejemplo, se podrían incluir modelos concretos para la medición de bases de datos relacionales [2], objeto-relacionales [22], activas [4], etc... y modelos concretos para medir artefactos software como los diagramas de clases UML [8], diagramas de transición de estados [9], etc... Además también en la organización a este nivel se podría disponer de modelos de medición de los propios modelos de procesos definidos.

Todos los modelos definidos en el nivel M1 son instancias de los conceptos representados en M2. Por lo tanto en el nivel de abstracción M2 de la arquitectura conceptual, se deben incluir metamodelos genéricos para la creación de modelos concretos. En nuestro marco de trabajo ha sido necesario la inclusión de dos metamodelos genéricos:

- Metamodelo Genérico de Procesos Software, a partir del cual se puedan derivar modelos concretos de procesos. Para ello se usa SPEM (ver apartado 3) como metamodelo de procesos. Este metamodelo contiene los constructores necesarios para definir cualquier modelo de proceso software concreto
- Metamodelo Genérico de la Medición, a partir del cual sea posible definir modelos concretos de medida. En el apartado 4 se describe en detalle este metamodelo.

En el último nivel conceptual de la arquitectura, M3, todos los conceptos del metamodelo de procesos y del metamodelo de la medición son representados usando el lenguaje abstracto MOF, que está compuesto básicamente por dos constructores: Clase-MOF y Asociación-MOF (éstos son los principales elementos para nosotros, aunque existen otros como: paquete, tipos de datos, etc...). De esta forma, todos los conceptos del nivel M2 son instancias de Clase-MOF o Asociación-MOF. Por ejemplo los conceptos de SPEM tales como "Actividad", "Producto de Trabajo" y conceptos del metamodelo de medición como "Métrica", "Indicador", "Unidad de Medida" son instancias de Clase-MOF, y las relaciones "Actividad precede Actividad", "Producto de Trabajo precede Producto de Trabajo" o "Métrica tiene Unidad de Medida", son instancias de Asociación-MOF.

Con esta arquitectura se hace posible la gestión integrada de la mejora del proceso software al integrarse de forma sistemática la definición de procesos y su medición. Como soporte a esta arquitectura conceptual se ha desarrollado la herramienta MANTIS-Metamod [7], que permite la definición de los metamodelos (en base a los constructores del lenguaje MOF) y de los modelos (en base a los constructores de sus metamodelos) necesarios. Para la gestión del almacenamiento e intercambio de los metadatos de la arquitectura conceptual, se usa un gestor de repositorio [24] que implementa el estándar XMI (XML Meta-data Interchange) [21] para promover la portabilidad de los modelos y metamodelos definidos.

3. Metamodelo SPEM (Software Process Engineering Metamodel).

SPEM (Software Process Engineering Metamodel) es una especificación de OMG en fase de finalización, y en la que se describe un metamodelo genérico para la descripción de procesos software concretos. Este metamodelo está basado en UML [18], y por tanto, en los principios de orientación a objetos, y sirve como plantilla para la creación de modelos de procesos concretos, como podrían ser: el Proceso Unificado de

Desarrollo de Rational (RUP), el modelo de evaluación y mejora de CMMi, etc.... En la especificación se define SPEM como un metamodelo de procesos software, y como un perfil (*profile*) UML.

Como metamodelo, SPEM especifica el conjunto mínimo de elementos necesarios para describir cualquier proceso concreto de desarrollo software, sin incluir constructores para áreas o disciplinas específicas, de forma que en SPEM se describe un metamodelo genérico. El objetivo fundamental de esta especificación es tratar de homogeneizar la diversidad terminológica existente en los lenguajes de modelado de procesos software, en los que los mismos conceptos se tratan con nombres diferentes.

El modelo conceptual de SPEM está basado en la idea de que un proceso de desarrollo software consiste en la colaboración entre entidades abstractas y activas denominadas "roles de proceso" (*process roles*) que realizan operaciones denominadas "actividades" (*activities*) sobre entidades tangibles denominadas "productos de trabajo" (*workproducts*). La especificación SPEM está compuesta por un conjunto de paquetes en los que se describen cada uno de sus elementos. Todos estos paquetes se construyen a partir del paquete *SPEM\_Foundation*, que es un subconjunto de UML 1.4 [18], y el paquete de extensiones de SPEM (*SPEM\_Extensions\_Package*), que añade los constructores y la semántica necesaria para la ingeniería del proceso software. SPEM se estructura básicamente en 5 paquetes que son:

- **Elementos Básicos.** En este paquete se describen los elementos básicos necesarios para la descripción de procesos.
- **Dependencias (*Dependencies*).** Este paquete contiene las siguientes dependencias definidas en SPEM: *Categoriza (*categorizes*)*, que es una relación que va de un paquete determinado a un elemento de proceso correspondiente a otro paquete, y proporciona un medio para asociar a los elementos del proceso múltiples categorías; *Impacta (*impacts*)*, que relaciona productos de trabajo, e indica que la modificación de un producto de trabajo podría invalidar otro; *Importa (*import*)*, que representa que los contenidos del paquete destino se añaden al espacio de nombres (*Namespace*) del paquete origen; *Precede (*precedes*)*, que es una relación entre actividades, o entre definiciones de trabajo (*WorkDefinitions*), e indica dependencias "comienzo-comienzo", "fin-comienzo" o "fin-fin" entre estos elementos; *Refiere a (*refers to*)*, que relaciona elementos de proceso e indica si están incluidos en el mismo componente de proceso y *Traza (*trace*)*, que es una relación entre "Definiciones de Trabajo" y "Elementos de Información", usada para traza de requerimientos y cambios entre modelos.
- **Estructura del Proceso (*Process Structure*).** En este paquete se describen los principales elementos estructurales a partir de los cuales se construye la descripción de un proceso. En la Figura 1 se representan los componentes de este paquete. Los principales elementos de este paquete son: "Producto de Trabajo" (*WorkProduct*) o artefacto, que es cualquier cosa que es producida, consumida o modificada por un proceso. Podría ser un fragmento de información, un documento, un modelo, código fuente, etc; "Definición de Trabajo" (*Work Definition*) que es una clase no abstracta de operación que describe el trabajo realizado en el proceso. Tiene entradas y salidas explícitas representadas mediante el constructor "Parámetro de Actividad" (*ActivityParameter*); "Actividad" (*Activity*), que es la principal clase en la que se especializa Definición de Trabajo y describe una parte o partes de trabajo realizadas por un rol de proceso tales como las tareas, las operaciones y acciones que un determinado rol realiza o asiste. Una actividad puede estar formada por un conjunto de elementos atómicos denominados "Pasos" (*steps*); Realizador del Proceso (*Process Performer*) que describe el encargado de realizar un conjunto de "Definiciones de Trabajo" que componen un proceso y "Rol del Proceso" (*ProcessRole*) que es una subclase de Realizador del Proceso y describe las responsabilidades asociadas a los "Productos de Trabajo" junto con los roles que realizan y asisten en actividades específicas.

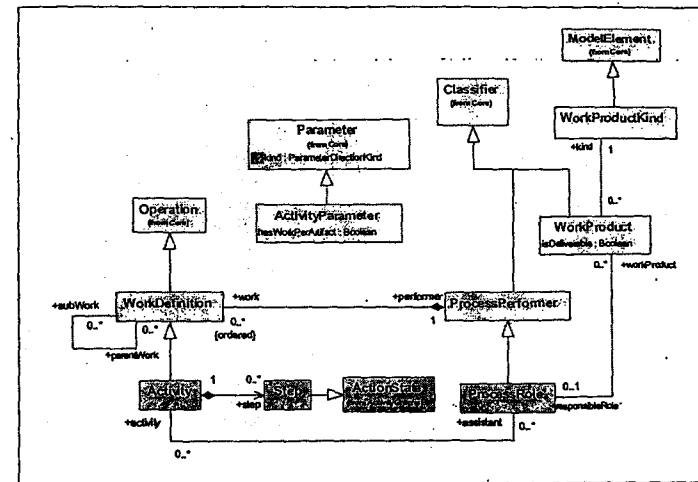


Figura 1. Paquete de SPEM estructura del proceso

- **Componentes del Proceso (*Process Components*).** Este paquete contiene los elementos necesarios para dividir una o más descripciones de proceso en partes auto-contenidas sobre las que se pueden aplicar procesos de gestión de la configuración o control de versiones.
- **Ciclo de Vida del Proceso (*Process Lifecycle*).** En este paquete se incluyen los elementos de definición de procesos que ayudan a definir cómo se van a ejecutar los procesos. Describen o restringen el comportamiento del proceso a realizar, y se utilizan para asistir en la planificación, ejecución y monitorización del proceso. Con estos elementos se establece el orden de ejecución del proceso y además permite definir iteraciones y fases.

En definitiva, SPEM constituye un metamodelo de referencia para la definición de forma consistente de los distintos modelos de procesos necesarios en una organización, lo que facilita su gestión integrada dentro de la arquitectura conceptual propuesta, al estar los conceptos de los distintos modelos agrupados bajo una terminología común.

#### 4. Incorporación de la medida en la arquitectura conceptual para la evaluación y mejora de procesos.

Un elemento fundamental a tener en cuenta a la hora de establecer un marco de trabajo para la mejora de los procesos es la posibilidad de definir indicadores objetivos de los mismos que permitan que una organización software en un momento dado pueda evaluar y mejorar sus procesos de una forma eficiente. Para ello es imprescindible establecer un marco de medición de los procesos.

Una buena base para establecer un proceso de medición es la proporcionada por CMMi [26]. En CMMi se incluye una nueva área clave de proceso denominada "Medición y Análisis". El objetivo de este área es desarrollar y establecer una capacidad de medición que se pueda usar para dar soporte a las necesidades de información de la organización, e implica una ampliación a los conceptos incluidos en el modelo CMM. Da soporte al resto de áreas de proceso proporcionando un marco de trabajo a las organizaciones a la hora de alinear los objetivos y necesidades de medición con un enfoque de medición basado en proporcionar resultados objetivos que sean útiles para la toma de decisiones y acciones correctivas. Este enfoque es consistente con las ideas de Goal-Question-Metric [1] y el estándar ISO 15939 [11]. En la siguiente figura se

representa mediante un diagrama de flujo de datos, los principales procesos relacionados con esta área, y las relaciones de datos entre los mismos:

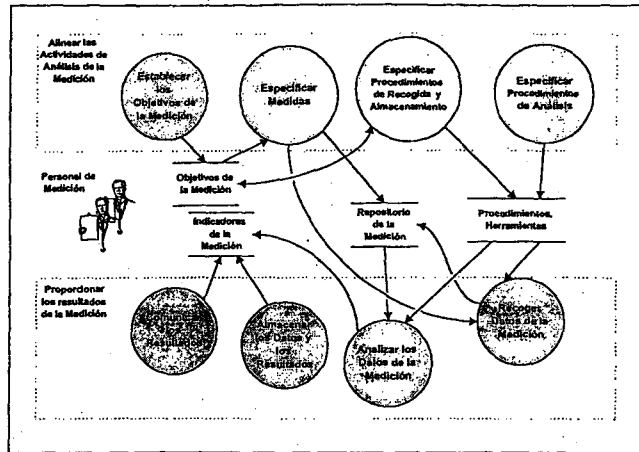


Figura 2. Área Clave "Medición y Análisis" de CMMi

Como se puede observar en la Figura 2, a la hora de establecer un proceso de medición efectivo en una organización es necesario conseguir dos objetivos fundamentales:

- **Alinear las actividades de análisis de la medición.** Para conseguir este objetivo en CMMi se identifican las siguientes prácticas: establecer los objetivos de la medición, especificar medidas, especificar procedimientos de recogida y almacenamiento y especificar procedimientos de análisis. A partir de estas prácticas se establece un plan para la medición y el análisis con el que se pretende resolver cuestiones tales como: ¿por qué se mide?, ¿qué se va a medir?, ¿cómo se va a medir?, etc....
- **Proporcionar los resultados de la medición.** Las prácticas asociadas con la consecución de este objetivo son: Recoger los datos de la medición, analizar los datos de la medición, almacenar los datos y resultados y comunicar los resultados. Por lo tanto, con estas prácticas se pretende establecer un buen proceso de recogida y comunicación de los resultados, ya que éstos deben proporcionarse a la persona adecuada para satisfacer sus necesidades de información.

Por lo tanto, el primer paso del proceso de medición es el de identificar los objetivos de la medición para, en un segundo paso, implementar el proceso de medición y análisis, lo que requiere la integración de la medición en los distintos procesos del trabajo de una organización. Como se puede apreciar en el marco de medición de CMMi es muy importante, para una organización que quiera implantar un proceso efectivo de medición, poder definir de forma precisa modelos concretos de medida que, siendo soportados por una herramienta integrada de medición, permitan una automatización adecuada y necesaria para la evaluación de los procesos.

La mayoría de los problemas de recolección de datos de un proceso de medición se deben fundamentalmente a una pobre definición de las medidas software que se están aplicando. Por todo ello es importante no sólo recoger los valores propios del proceso de medición, sino representar adecuadamente los metadatos asociados con dichos valores. En [14] se define un método para la especificación de modelos de medición con el objeto de capturar las definiciones y relaciones entre medidas software. El marco de trabajo propuesto está compuesto por tres niveles de abstracción para la medición, partiendo desde un modelo

genérico de la medida, hasta la automatización de la recogida de los valores de las métricas a nivel de proyecto. Esta idea de abstracción es fundamental para poder integrar el proceso de medición de una forma efectiva en una organización.

Con todo ello, resulta muy conveniente incorporar un metamodelo genérico para la medición, a partir del cual pueda ser posible derivar modelos concretos de medida que constituyan la base en una organización para sus procesos de evaluación y mejora. En la Figura 3 se representa en UML nuestra propuesta de metamodelo de medida basada en el estándar ISO 15939:

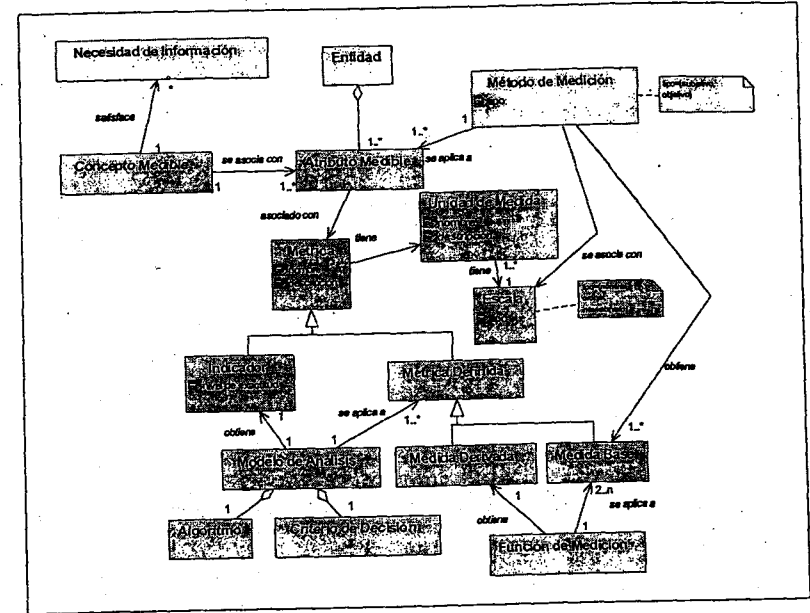


Figura 3. Metamodelo genérico de la medida

Como se puede observar en la Figura 3, desde el punto de vista de la medición, los elementos sobre los que se pueden medir propiedades son "Entidades". Una entidad es un objeto (como por ejemplo, un proceso, producto, proyecto o recurso) que puede ser caracterizado mediante la medición de sus atributos. Los "Atributos" describen propiedades o características de las entidades, que pueden ser distinguidas de forma cuantitativa o cualitativa mediante medios humanos o automáticos. Los atributos tienen como objetivo satisfacer determinadas necesidades de información, como por ejemplo, "la necesidad de comparar la productividad de desarrollo software respecto de un valor determinado". Esta relación abstracta entre atributos y necesidades de información se representa mediante el elemento denominado "Concepto Medible", que en este caso sería "ratio de productividad del desarrollo software". Como atributos se podrían usar atributos de tamaño del producto desarrollado, o de esfuerzo de desarrollo.

Todo atributo medible se asocia con una métrica, que es una abstracción de los diferentes tipos de medidas que se usan para cuantificar y tomar decisiones sobre las entidades. Toda métrica tiene asociada una unidad de medida (como por ejemplo, líneas de código) que a su vez pertenece a una escala determinada. De acuerdo al estándar se distinguen cuatro escalas que son: nominal, ordinal, intervalo y ratio, aunque se pueden establecer otras clasificaciones como la de Kitchenham *et al.* [14]. Existen tres tipos de métricas, que son:

- **Medida Base**, que se define en función de un atributo y el método necesario para cuantificarlo (una medida es una variable a la que se le asigna un valor). Captura información sobre un atributo simple.
- **Medida Derivada**, que es una medida definida en función de dos o más valores de medidas base.
- **Indicador**. Es una medida que proporciona una estimación o evaluación de atributos específicos derivados de un modelo con respecto a unas necesidades de información. Los indicadores son la base para el análisis y la toma de decisiones.

Estas medidas son las que se tienen que presentar a los usuarios encargados del proceso de medición. Los procedimientos para calcular cada uno de los tipos de métricas son los siguientes:

- Los valores de las medidas base se obtienen con "Métodos de Medición", que consisten en una secuencia lógica de operaciones descritas de forma genérica, usados para cuantificar un atributo con respecto a una escala específica. Estas operaciones pueden implicar actividades como, contar ocurrencias u observar el paso del tiempo. El mismo método de medición se puede aplicar a múltiples atributos.
- Las medidas derivadas se obtienen mediante la aplicación de una "Función de Medición", que es un algoritmo o cálculo realizado para combinar dos o más medidas base. La escala y unidad de la medida derivada depende de las escalas y unidades de las medidas base.
- Los indicadores se obtienen a partir de un "Modelo de Análisis". Un modelo de análisis produce estimaciones o evaluaciones relevantes ante las necesidades de información definidas. Consiste en un algoritmo o cálculo que combina una o más medidas base y/o derivadas con determinados criterios de decisión. Todo criterio de decisión está compuesto de una serie de valores límite u objetivo usados para determinar la necesidad de actuar o investigar, o para describir el nivel de confianza ante un resultado determinado. Este criterio ayuda a interpretar los resultados de la medición.

A partir de este metamodelo de referencia es posible medir cualquier elemento de un modelo de procesos o datos. Por ejemplo, en SPEM no se identifican de forma explícita elementos de metamodelado (sólo se puede incluir una métrica como una clase de Guía) para la medición. Por ello, con la incorporación de esta extensión de la medida al metamodelo SPEM se ofrece la posibilidad de, a la hora de definir modelos de procesos concretos, especificar qué métricas van a ir asociadas con las características del proceso (ya sea sobre actividades, o sobre los productos de trabajo u otra entidad medible). Si aplicamos este metamodelo sobre los elementos del modelo SPEM, podríamos medir elementos destacados tales como la clases *Actividad*, *Producto de Trabajo* y *Realizador del Proceso*. Estos elementos del modelo tienen un conjunto de atributos medibles, como podría ser para una actividad: "el número de actividades con la que tiene una dependencia de tipo precede". Dicho atributo sería calculado con una métrica para satisfacer una necesidad de información como "Evaluar el Acoplamiento de los Procesos" y la unidad de medida en este caso sería de tipo "Ratio". De esta forma se facilita la labor de un proceso de evaluación y mejora, al tener registradas de forma cuantitativa el desempeño de los procesos software que se llevan a cabo en una determinada organización.

**5. GenMETRIC. Herramienta Extensible para la gestión integrada del proceso de medición.**

Con el objetivo de dar soporte automático al proceso integrado de medición comentado en el apartado anterior, hemos desarrollado la herramienta GenMETRIC. GenMETRIC es una herramienta extensible para la definición, cálculo y visualización de métricas software. Esta herramienta para la gestión integrada del proceso de medición, da soporte a la definición y gestión de métricas tanto de artefactos como de procesos software. Además, la herramienta da soporte al metamodelo de la medida basado en ISO 15939 que ha sido propuesto para un mejor soporte y gestión del proceso integrado de medición. En la siguiente figura se representan los componentes de la herramienta que han sido diseñados siguiendo la arquitectura clásica de tres capas [16]:

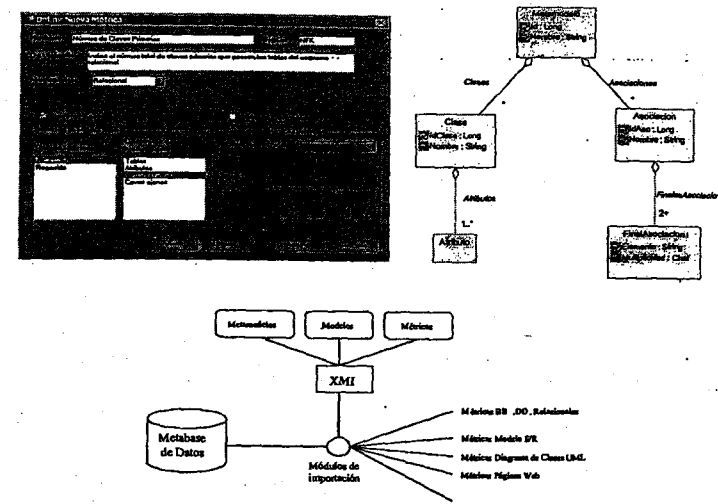


Figura 4. Arquitectura de tres capas de GenMETRIC

Las capas básicas de la arquitectura de la herramienta representadas en la figura son:

- **Presentación**. Desde el punto de vista de uso de la herramienta se han definido dos roles de seguridad que son: *Administrador*, que tiene el control sobre la funcionalidad total de la herramienta, lo que le permite poder definir, calcular y visualizar cualquier métrica y *Usuario*, que tiene acceso al cálculo y visualización de las métricas que hayan sido ya definidas.
- **Dominio**. Formado por las clases encargadas de gestionar el cálculo de las métricas en función de la información contenida en los distintos metamodelos y modelos definidos.
- **Almacenamiento**. Para la gestión del proceso de medición la herramienta puede importar información sobre los siguientes elementos representados en forma de documentos XMI [21]:
  - **Metamodelos del dominio** sobre los que se quieren definir métricas. Los elementos de cada metamodelo son almacenados para poder medirlos. Por ejemplo, si lo que interesa medir es una base de datos relacional, habría que definir previamente los elementos del metamodelo relacional, que serían: Tabla, Atributo, Relación, etc...
  - **Modelos del Dominio**. Los modelos son instancias de los metamodelos y son sobre los que interesa realizar un proceso de medición. Por ejemplo, el esquema (modelo) de una base de datos de un banco, sería una instancia del metamodelo relacional sobre el que podríamos realizar un proceso de medición.
  - **Modelos de Métricas**. Los modelos de métricas permiten almacenar las métricas definidas de forma consistente. Para ello, los modelos de métricas usados por la herramienta, son instancias del metamodelo de la medida propuesto en el apartado anterior.

La información importada por la herramienta sobre los distintos metamodelos y modelos del dominio y de métricas, se almacena de forma persistente en una *Metabase* o *Repositorio*. A partir de la información de la *Metabase* se gestiona el cálculo de métricas. Los distintos modelos y metamodelos necesarios son definidos y representados en XMI mediante la herramienta MANTIS-Metamod [7].

Con la herramienta propuesta se proporciona un entorno integrado y automático para la medición. Al ser una herramienta genérica, es posible la definición de cualquier métrica nueva sobre los metamodelos de dominio existentes sin tener que codificar nuevos módulos. Además, la herramienta es extensible, lo que

facilita la incorporación de nuevos metamodelos de dominio, como podría ser un metamodelo para definir los elementos del web (formado por páginas web, enlaces entre páginas, etc.) y poder de esta forma medir modelos de dominio concretos, como sitios web. Además, al trabajar con documentos XMI se facilita la comunicación y posibilidad de importar de una forma abierta nuevos metamodelos del dominio, o modelos de dominio y métricas almacenados en otros repositorios basados en MOF.

## 6. Conclusiones y trabajos futuros.

En este trabajo se ha presentado un marco conceptual para la mejora basado en el modelado y la medición integrada de los procesos software. Para el modelado de los procesos bajo una terminología común se usa el metamodelo SPEM, y como marco integrado para la medición se ha definido un metamodelo, basado en el estándar ISO 15939, que facilita la gestión de un proceso integrado de medición para promover la mejora de procesos en una organización. Como soporte a la medición integrada, se ha desarrollado la herramienta *GenMetric*, para la definición, cálculo y visualización de métricas software, que permite la incorporación de nuevos tipos de métricas y nuevos tipos de elementos a medir, al estar diseñada su arquitectura de forma genérica y extensible.

Con el marco propuesto, cualquier organización dedicada al desarrollo y/o mantenimiento del software puede definir y evaluar sus procesos de una forma efectiva, como paso previo para promover la mejora de los mismos. Además, al estar el marco de trabajo basado en el estándar MOF, es posible la extensión y modificación de los elementos del mismo de una forma sencilla mediante la incorporación o modificación de los metamodelos necesarios, al estar todos representados siguiendo la terminología común proporcionada por el modelo MOF. Entre las líneas futuras destacan:

- Refinamiento del metamodelo genérico de la medida presentado teniendo en cuenta la propuesta de Olsina *et al* [17], para unificar las dos propuestas con un metamodelo de la medida que sea la referencia común en las organizaciones.
- Descripción de modelos concretos de medida, a partir del metamodelo genérico, para dar soporte de una forma efectiva a la evaluación y mejora de los procesos software.
- Incorporación de la gestión del conocimiento en el marco integrado para obtener un alto grado de calidad de servicio [15].

## Agradecimientos

Este trabajo forma parte del proyecto TAMANSI parcialmente financiado por la "Consejería de Ciencia y Tecnología, Junta de Comunidades de Castilla-La Mancha" (referencia PBC-02-001) y del proyecto DOLMEN, parcialmente financiado por la Subdirección General de Proyectos de Investigación, Ministerio de Ciencia y Tecnología (TIC 2000-1676-C06-06).

## Referencias.

- [1] V.R. Basili, G. Caldiera and H. D. Rombach. *Goal Question Metric Paradigm*. In John J. Marciniak, editor, *Encyclopedia of Software Engineering*, volume 1. John Wiley & Sons, 1994. 528-532.
- [2] C. Calero, M. Piattini y M. Genero. "Empirical Validation of referential metrics. Information Software and Technology". *Special Issue on Controlled Experiments in Software Technology*. Vol.43, Nº 15, 2001.
- [3] J.C. Derniame, B. A. Kaba and D. Wastell. *Software Process: Principles, methodology and technology. Lecture Notes in Computer Science 1500*. Springer, 1999.
- [4] O. Diaz, M. Piattini y C. Calero. "Measuring triggering-interaction complexity on active databases". *Information Systems Journal*. Elsevier Science. Vol. 26, Nº 1, 2001.
- [5] W.A. Florac y, A. D. Carleton. *Measuring the Software Process. Statistical Process Control for Software Process Improvement*. SEI Series in Software Engineering. Addison Wesley, 1999.
- [6] A. Fuggetta. *Software Process: A roadmap. The future of Software Engineering*, ed. A. Finkelstein. ACM, Press, 27-34, 2000.

- [7] F. García, F. Ruiz, M. Piattini and M. Polo. "Conceptual Architecture for the Assessment and Improvement of Software Maintenance". *4th International Conference on Enterprise Information Systems (ICEIS'02)*. Ciudad Real, Spain, 610-617, April 2002.
- [8] M. Genero, J. Olivás, M. Piattini y F. Romero. "Using metrics to predict OO information systems maintainability". *13th International Conference Advanced Information Systems Engineering (CAISE'01)*, 388-401, 2001.
- [9] M. Genero, D. Miranda and M. Piattini. "Defining and Validating Metrics for UML Statechart Diagrams". In *6th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE)*, 120-136, 2002.
- [10] ISO IEC 15504 TR2:1998, part 2: *A reference model for processes and process capability*, ISO/IEC JTC1/SC7, 1998.
- [11] ISO IEC 15939, *Information Technology - Software Measurement Process*, Committee Draft, December 2000.
- [12] International Organization for Standardization (ISO). 2000. *Quality management systems - Fundamentals and vocabulary*. ISO 9000:2000. See [http://www.iso.ch/iso/en/iso9000-14000/iso9000/seleccion\\_use/iso9000family.html](http://www.iso.ch/iso/en/iso9000-14000/iso9000/seleccion_use/iso9000family.html)
- [13] International Organization for Standardization (ISO). 2000. *Quality management systems - Requirements* ISO 9001:2000
- [14] B. A. Kitchenham, R. T. Hughes y S.G. Linkman. "Modeling Software Measurement Data". *IEEE Transactions on Software Engineering*. 27(9), 788-804, 2001.
- [15] R. J. Kusters, J. Trienekens y W. Hassoldt. "On the Business Impact of Software Process Improvement". *Proceedings of the 26th Annual International Computer Software and Applications Conference (COMPSAC)*, 2002.
- [16] C. Larman. *Applying UML and Patterns*. Upper Saddle River, NJ: Prentice-Hall, 1998.
- [17] L. A. Olsina, M. F. Bertoa, G. J. Lafuente, M. A. Martín, M. Katrib y A. Vallecillo. "Un Marco Conceptual para la Definición y Explotación de Métricas de Calidad". *Proceedings de las VII Jornadas de Ingeniería del Software y Bases de Datos (JISBD)*. Madrid, Nov. 2002.
- [18] *OMG Unified Modeling Language Specification*; versión 1.4, septiembre 2001. Object Management Group. Available in <http://www.omg.org/technology/documents/formal/uml.htm>.
- [19] *Meta Object Facility (MOF) Specification*; versión 1.4, abril 2002. Object Management Group. In <http://www.omg.org/technology/documents/formal/mof.htm>.
- [20] *Software Process Engineering Metamodel Specification*; adopted specification, mayo-2002. Object Management Group. Available in <http://cgi.omg.org/cgi-bin/doc?ptc/02-05-03>.
- [21] *OMG XML Metadata Interchange (XMI) Specification*; versión 1.2, enero 2002. Object Management Group. In <http://www.omg.org/technology/documents/formal/xmi.htm>
- [22] M. Piattini, C. Calero, H. Sahraoui y H. Lonis. "Object-Relational Database Metrics". *L'object*. ISSN 1262-1137. HERMES Science Publications, Paris. Vol.7, Nº 4, 2001.
- [23] F. Ruiz, M. Piattini and M. Polo. "An Conceptual Architecture Proposal for Software Maintenance". *International Symposium on Systems Integration (ISSI, Intersymp'2001)*. Baden-Baden, Germany (2001), VIII:1-8
- [24] F. Ruiz, M. Piattini, F. García y M. Polo. "An XMI-based Repository for Software Process Metamodeling". *Proceedings of 4th International Conference on Product Focused Software Process Improvement (PROFES'2002)*. Rovaniemi, Finland. December 2002. *Lecture Notes in Computer Science (LNCS 2559)*, Markku Oivo, Seija Komi-Sirviö (Eds.). Springer. 546-558.
- [25] *The Capability Maturity Model: Guidelines for Improving the Software Process*, 1995. In <http://www.sei.cmu.edu/cmm/cmm.html>
- [26] *Capability Maturity Model Integration (CMMI<sup>SM</sup>)*, version 1.1. March 2002. In <http://www.sei.cmu.edu/cmmi/cmmi.html>



[www.uc.edu.py](http://www.uc.edu.py)



[www.uaa.edu.py](http://www.uaa.edu.py)



[www.cytcd.org](http://www.cytcd.org)