

ICIE
VII Congreso Internacional de Ingeniería Informática
25, 26 y 27 de Abril de 2001, Buenos Aires - Argentina



VII Congreso Internacional de Ingeniería Informática

Herramienta para la representación de procesos software mediante modelos MOF

Luis MÁRQUEZ, Félix GARCÍA, Francisco RUIZ, Mario PIATTINI, Macario POLO

Grupo Alarcos

Escuela Superior de Informática, Universidad de Castilla-La Mancha
Ronda de Calatrava, s/n. 13071, Ciudad Real (España)

{ lmarquez|fgarcia }@proyectos.inf-cr.uclm.es; fruiz@inf-cr.uclm.es

Palabras clave:

Metamodelos de Procesos Software, Workflow, MOF, XMI.

Resumen

En este documento presentamos MANTIS-Metamod, una herramienta para representar metamodelos de procesos software basada en el estándar MOF (Meta-Object Facility) propuesto por el Object Management Group (OMG, 2000). Este estándar define un marco de trabajo utilizando los principios de modelado orientado a objetos para la definición de modelos de información para metadatos.

MANTIS-Metamod es un componente de MANTIS, un entorno integral para la gestión del proceso de mantenimiento del software (PMS). MANTIS utiliza 4 niveles conceptuales (basados en la arquitectura propuesta por MOF) para abordar la gran cantidad de aspectos diferentes que deben ser tenidos en cuenta al modelar, gestionar, controlar y ejecutar un proceso software. Aunque MANTIS está dedicado al PMS, la herramienta permite la meta-modelización de cualquier proceso software.



1- Introducción.

El mantenimiento del Software representa la etapa del ciclo de vida de un producto software que más recursos consume (Pigosky, 1996), y teniendo en cuenta sus características especiales, que lo diferencian significativamente del desarrollo, es muy útil disponer de métodos, técnicas y herramientas específicos (Piattini et al., 1998).

Por ello se hace necesaria la definición y construcción de un entorno para la gestión integral del proceso de mantenimiento del software, dada la complejidad que supone este tipo de proyectos (debido al tamaño y complejidad del producto a mantener y la dificultad de la tarea a realizar).

El proyecto MANTIS pretende la definición y construcción de un entorno integrado para la gestión del PMS ("integrated big-E Environment"). Con el término "big-E Environment" se considera MANTIS como un concepto más amplio que los conceptos de:

- Metodología (en el sentido habitual, es decir, una serie de métodos o técnicas relacionados)
- Entorno de Ingeniería del Software (EIS), es decir, una colección de herramientas software utilizadas para soportar actividades de ingeniería del software (ISO/IEC, 2000)

MANTIS incluye los diferentes aspectos que deben ser tenidos en cuenta para llevar a cabo proyectos de mantenimiento del software. Para la gestión de PMS, MANTIS integra, entre otros, a las personas (con ciertas habilidades y que desempeñan ciertos roles en el proyecto), las técnicas (metodologías) empleadas por las personas, las herramientas (que ayudan a cumplir los estándares) y las actividades (en las que participan los equipos y que ayudan a alcanzar hitos significativos).

2- MANTIS y MOF.

Un principio importante de la ingeniería del software moderna es la separación de un sistema en capas de encapsulación, que pueden especificarse, diseñarse y construirse de manera independiente en gran parte. Con esta filosofía y con el objetivo de reducir la complejidad, en MANTIS se definen 4 niveles conceptuales que están basados en el estándar MOF para metamodelización con orientación a objetos, propuesto por el Object Management Group, OMG (1999). En la tabla 1 se muestran estos cuatro niveles de la arquitectura MOF y su adaptación a MANTIS.

Nivel	MOF	MANTIS
M3	Modelo MOF (Meta-metamodelo)	Modelo MOF
M2	Meta-modelo	Metamodelo PMS
M1	Modelo	MANTEMA y otras técnicas (Modelo concreto del PMS)
M0	Datos	Instancias de PMS (proyectos concretos de mantenimiento del software)

Tabla 1. Niveles Conceptuales en MOF y MANTIS.

En el nivel M0 están los ejemplares de proyectos reales y concretos de mantenimiento del software con restricciones de tiempo, costes, etc. Los datos manejados en este nivel son instancias de los conceptos definidos en el nivel superior M1. El modelo concreto que utilizamos en el nivel M1 está basado en la metodología MANTEMA y en un conjunto de técnicas adaptadas a las particularidades del mantenimiento: estimación de esfuerzo, estimación de riesgos, auditoría del proceso (Ruiz et al., 2000), etc. El nivel M2 se corresponde con el metamodelo del PMS (algunos autores no distinguen claramente en la nomenclatura entre modelo y metamodelo llamando modelo tanto al correspondiente al nivel M2 como al del nivel M1). Por ejemplo, el concepto genérico de Activity utilizado en M2, se instancia en las actividades "*Análisis de Informe de Problemas*" o "*Implementación de la Modificación*" en M1, y éstas a su vez, en instancias del nivel M0 como "*Análisis de informe del problema nº 36 del proyecto PATON*".

En el último nivel conceptual de MANTIS, el M3, el metamodelo de PMS es representado en un modelo-MOF. Un modelo-MOF está formado básicamente por dos tipos de objetos: clases-MOF y asociaciones-MOF (estos son los principales objetos para nosotros, aunque también existen otros: paquetes para fines de reutilización, tipos de datos, ...). Por tanto, todos los conceptos representados en el nivel M2 se consideran ahora ejemplares de clases-MOF o de asociaciones-MOF. Por ejemplo, "Actividad", "Actor" o "Artefacto" serán ejemplares de clase-MOF; y "Actividad usa Recursos" o "Artefacto es_entrada_de Actividad" son ejemplares de asociación-MOF. Un modelo-MOF se puede representar, entre otros, mediante diagramas UML o mediante el lenguaje MODL (*Meta Object Definition Language*), pero para su manejo automático y su portabilidad entre herramientas en un (EIS), que es lo que interesa en MANTIS, es mucho mejor representarlo utilizando algún estándar de intercambio de metadatos. Por esta razón, en MANTIS se utiliza XMI (*XML Metadata Interchange*) (OMG, 1999), basado en XML (*eXtensible Markup Language*), para almacenar los metamodelos.

MOF es un modelo para especificar, construir, gestionar, intercambiar e integrar metadatos en sistemas software, permitiendo una flexible integración de sistemas. MOF describe un lenguaje de modelado abstracto alineado con el núcleo del UML de OMG.

En la siguiente figura, se representa mediante un diagrama UML un resumen de la jerarquía de clases que componen el modelo MOF.

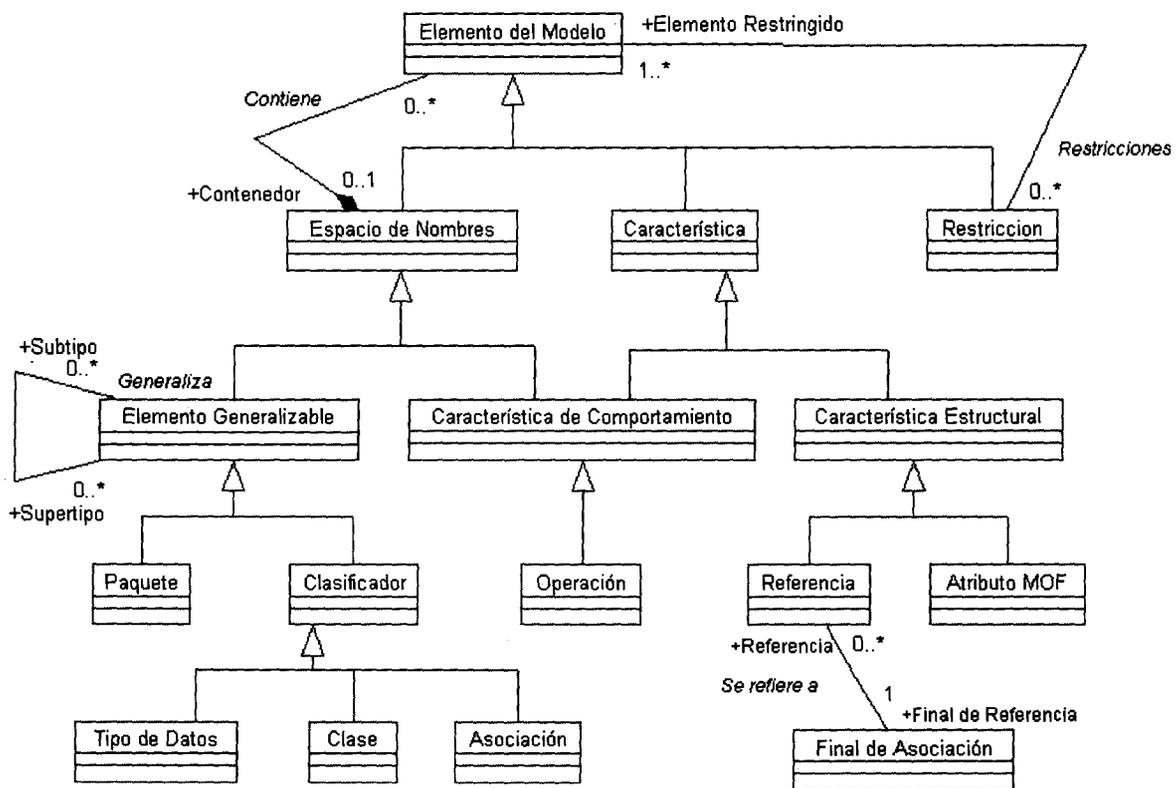


Figura 1. Jerarquía Clases MOF.

Con la incorporación de este último nivel conceptual y con la utilización de estándares, para metamodelado (MOF) y para intercambio de metadatos (XMI), se puede conseguir un entorno con la mayor flexibilidad para definir y compartir modelos y metamodelos. La inclusión del nivel M3 permite trabajar con diferentes versiones del metamodelo del PMS, condición necesaria para poder gestionar la mejora del proceso.



3- Descripción de la herramienta.

MANTIS-Metamod es una herramienta para representar metamodelos de procesos software basada en el estándar MOF propuesto por el Object Management Group (OMG, 2000).

La herramienta da soporte al modelado de procesos software en general (descritos según los niveles arquitecturales de MOF) pero en nuestro trabajo (enmarcado dentro del proyecto MANTIS) se aplica para el modelado del proceso de mantenimiento del software.

Por lo tanto la presente herramienta constituye el componente de MANTIS encargado de modelar los procesos del mantenimiento del software.

En este trabajo vamos a describir el componente de la herramienta encargado de la definición de las correspondencias entre el nivel 3 del modelo MOF (basado en clases MOF, asociaciones MOF) y el nivel 2 (instancias del nivel 3).

El objetivo que se persigue con este componente de la herramienta es facilitar la gestión integral del Proceso de Mantenimiento del Software (PMS) mediante la definición de metamodelos necesarios en dicha gestión, basándonos en una terminología común y en el mecanismo de abstracción que proporciona el nivel 3 de la arquitectura de niveles de MOF.

Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos ("3 tier Architecture") (Larman, 1998). La arquitectura de una aplicación es la vista conceptual de la estructura de ésta. Por ello, la arquitectura de la herramienta estará formada por 3 componentes o capas verticales, con el fin de reducir complejidad y proporcionar un cierto grado de encapsulación. Como se ha comentado estos niveles son:

- Interfaz
- Procesamiento
- Almacenamiento

La capa de interfaz será la encargada de obtener información del usuario, enviar la información del usuario para su procesamiento, recibir los resultados del procesamiento y presentación de los resultados al usuario. La capa de procesamiento tiene como funciones, recibir la entrada del nivel de presentación, interactuar con los servicios de almacenamiento para ejecutar las operaciones, y enviar el resultado procesado al nivel de presentación. Finalmente, la capa de almacenamiento es la encargada del almacenamiento, recuperación y mantenimiento de los datos.

Para la entrada datos del usuario, la aplicación está compuesta por un administrador de metamodelos como componente principal y por un sistema de ventanas que permite la descripción de forma visual de las clases que forman el núcleo del modelo MOF (Paquete, Clase, Tipo de Datos, Atributo, Operación, Referencia, Asociación, Final de Asociación y Restricción).

El administrador de metamodelos, al igual que el modelo MOF, está estructurado en forma de árbol: un paquete contiene clases y asociaciones, una clase contiene atributos y operaciones, una asociación contiene restricciones etc.

El sistema de ventanas asociado nos permite la descripción de las clases que forman el núcleo del modelo MOF. En la figura 2, puede verse la ventana asociada a la definición de una clase-MOF.

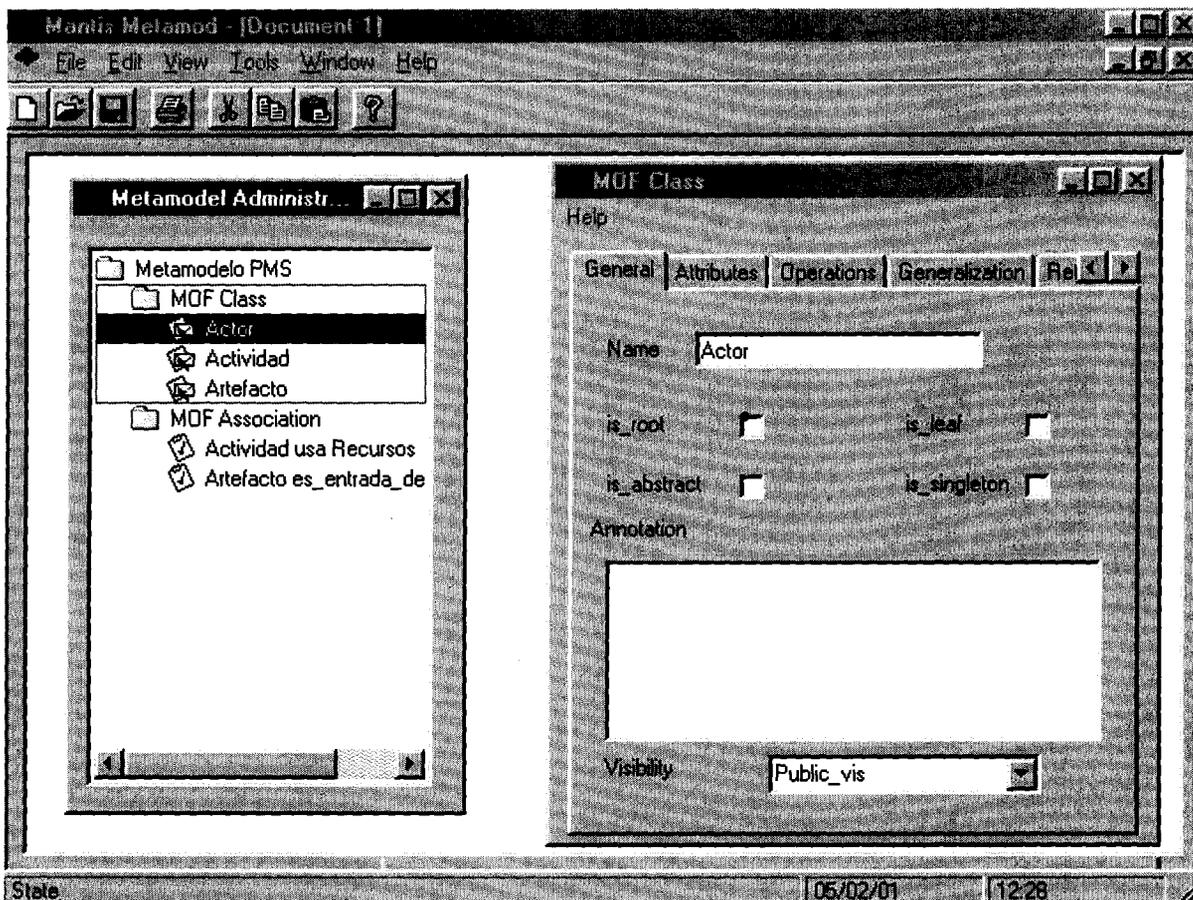


Figura 2- Aspecto de Mantis Metamod

Por ejemplo, una instancia de clase MOF podría ser la clase del nivel M2 "Actor" perteneciente al metamodelo de PMS. Dicha instancia de clase MOF, como se ve en la figura, permite que cualquier elemento del modelo pueda usarla (`visibility=public_vis`), no es abstracta (`is_abstract=false`), puede tener supertipos (`is_Root=false`), puede ser supertipo (`is_Leaf=false`) y como mucho puede existir una instancia del nivel M1 (`is_singleton=false`).

Los modelos definidos por el usuario a través de la interfaz, son validados y se representan internamente según la jerarquía de clases que describe el modelo MOF, especificada en el documento IDL adjunto al estándar MOF.



Para el almacenamiento de los metamodelos (nivel 2 definidos con MOF) se debe usar una representación que facilite el intercambio de información y, por lo tanto, proporcione una gran flexibilidad a la herramienta. Para tal fin se usará XMI.

El principal propósito de XMI es permitir fácil intercambio de metadatos entre herramientas de modelización (basadas en UML) y entre herramientas y repositorios de metadatos (basados en MOF) en entornos distribuidos heterogéneos.

XMI usa XML como sintaxis para la transferencia y formato de intercambio.

Por lo tanto para el almacenamiento de los metamodelos (nivel 2) tenemos que definir un DTD que represente MOF (nivel 3), de forma que todos los documentos XML almacenados deberán respetar las reglas definidas en dicho DTD para poder ser considerados como instancias del modelo MOF válidas.

Con XMI se facilita y permite el intercambio de metamodelos basados en MOF, de forma que la herramienta debe proveer dos servicios fundamentales:

- 1- Almacenamiento de los modelos MOF definidos en la herramienta en un repositorio local de metadatos representados en XMI (para exportación).
- 2- Importación de metamodelos.

La capa de almacenamiento proporciona los servicios necesarios para el almacenamiento de metadatos y para la extracción de los mismos, sobre un repositorio de metadatos formado por documentos XML.

Estos servicios son similares a los que nos podría proporcionar un Sistema Gestor de Bases de Datos Relacionales, con la particularidad de que en nuestro caso de lo que disponemos es de un repositorio de metadatos almacenados en documentos XML, es decir, se dispone de una colección de documentos estructurados.

Para el almacenamiento de los modelos MOF en documentos XML y para la extracción de información de los mismos, se utiliza una librería de servicios que a su vez usa los servicios que proporciona el XML DOM (Document Object Model). Este modelo proporciona una colección de clases que representan la estructura jerárquica en que se basa todo documento XML. El modelo proporciona clases que representan Documentos, nodos, listas de nodos, etc, con las propiedades y métodos necesarios para construir documentos XML basándonos en la estructura jerárquica que los caracteriza.

A continuación se muestra un esquema de la herramienta basado en su capa de almacenamiento y que resume los conceptos comentados:

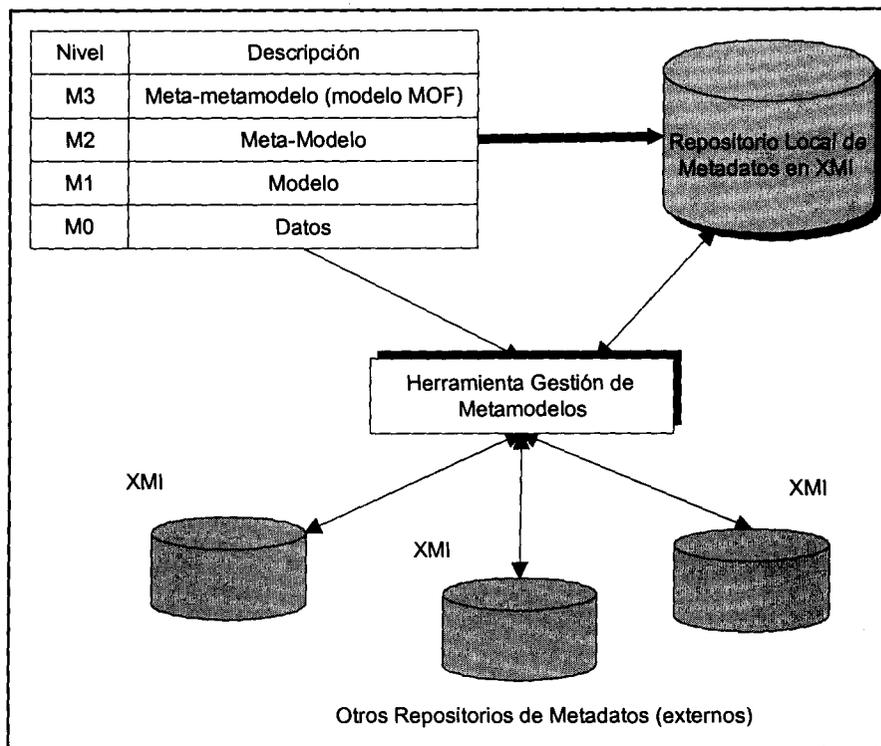


Figura 3- Esquema de la Aplicación

Como se muestra en la figura 3, para el almacenamiento de los modelos MOF la herramienta mantiene y gestiona un repositorio local de metadatos. Básicamente en esta figura están representados los dos servicios que proporciona la capa de almacenamiento:

- Almacenamiento de los modelos MOF definidos a través de nuestra herramienta en formato XMI, para facilitar su exportación.
- Importación de modelos de procesos (definidos según niveles MOF) basándonos en el formato de intercambio XMI, de forma que se puedan aprovechar modelos MOF útiles para la representación de procesos software independientemente de la plataforma y de la herramienta con la que hayan sido definidos.

Complementariamente a la herramienta que se propone pueden usarse otras de modelado visual basadas en UML para la visualización de los modelos MOF como diagramas de clases (por ejemplo Rational Rose). Esta funcionalidad se permite gracias a que, como se ha comentado, la herramienta dispone de la capacidad para intercambiar modelos MOF mediante el estándar de intercambio de metamodelos con XML (XMI).

4- Conclusiones

En el siguiente trabajo hemos presentado MANTIS-Metamod, herramienta para la descripción, importación y exportación de metamodelos de procesos software.

El objetivo perseguido con este componente de MANTIS es sacar provecho de las ventajas probadas que supone el uso de metamodelos para la gestión integral de procesos software. En MANTIS dicha integración está basada en el uso de una terminología común y en el mecanismo de abstracción que proporciona la arquitectura de niveles conceptuales descrita por el estándar MOF.

Una utilidad fundamental de la herramienta es que constituye el soporte necesario para la mejora de procesos software. En el contexto en que se enmarca esta herramienta es posible, gracias a la definición de las proyecciones adecuadas entre el nivel 3 y el nivel 2 de de MOF, se podría incorporar al metamodelo de proceso de mantenimiento del software (PMS) la subontología de la medida, mediante la agregación de las clases y asociaciones necesarias.

Actualmente estamos trabajando en ampliar las funcionalidades de nuestra herramienta. Uno de los objetivos principales a corto plazo es la de desarrollar los componentes encargados de dar soporte a la proyección entre los niveles M2-M1 y M1-MO de forma que se puedan integrar y manejar adecuadamente todos los conceptos manejados en el desarrollo de procesos software (desde su modelado hasta la gestión de la ejecución de proyectos concretos).

Agradecimientos:

Este trabajo está realizado en colaboración con la empresa Atos ODS con cargo a los proyectos MANTIS y MPM. MANTIS ha sido parcialmente financiado por la Union Europea y CICYT-España (1FD97-1608TIC). MPM ha sido parcialmente financiado por el Ministerio de Ciencia y Tecnología de España (FIT-070000-2000-307).

Referencias:

- (Becker y Webby, 1999) Becker-K., U., y Webby, R., *A Comprehensive Schema Integrating Software Process Modeling and Software Measurement*, Fraunhofer Institute, IESE report N° 047.99/E. v1.2, 1999.
- (ISO/IEC, 2000) ISO/IEC JTC1/SC7/WG4 15940 working draft 5: *Information Technology –Software Engineering Environment Services*, juny-2000.
- (Larman, 1998) Larman, C. *Applying UML and Patterns*. Up-per Saddle River, NJ: Prentice-Hall, 1998.
- (OMG, 1999) *OMG XML Metadata Interchange (XMI)*, v. 1.1, oct-1999.
- (OMG, 2000) *OMG Meta Object Facility (MOF) Specification*, v. 1.3 RTF, sep-1999. In <http://www.omg.org>.
- (Piattini et al., 1998) Piattini, M.; Ruiz, F.; Polo, M., Bastanchury, T., Fernández, I., Martínez, M.A. *Mantenimiento del Software: Conceptos, Métodos*,

- Herramientas y Outsourcing*. Ed Ra-Ma, España 1988.
- (Pigoski, 1996) Pigoski, T.M., *Practical Software Maintenance. Best Practices for Managing your Investment*. Ed. John Wiley & Sons, USA 1996.
- (Polo et al., 1999) Polo, M.; Piattini, M.; Ruiz, F.; Calero, C.; *MANTEMA versión 2.0: una Metodología para el Mantenimiento del Software*. UCLM, Dep. de Informática, informe técnico UCLM-DI-99-01.
- (Ruiz et al., 2000) Ruiz, F.; Piattini, M.; Polo, M.; Calero, C.; *Audit of Software Maintenance*. In "Auditing Information Systems". Idea Group Publishing, USA 2000.
- (WfMC, 1995) WfMC TC00-1003 1.1: *Workflow Management Coalition. The Workflow Reference Model*, jan-1995.

Autores:

Luis Márquez es ingeniero técnico en informática de gestión por la Universidad de Castilla-La Mancha (España). Actualmente se encuentra trabajando en la representación de metamodelos de procesos software y su reutilización posterior para definir modelos de procesos software específicos. Desarrolla su actividad dentro del grupo de investigación Alarcos de la UCLM.

Félix García es ingeniero técnico en informática de gestión por la UCLM (España). Actualmente se encuentra trabajando en el diseño de una arquitectura abierta para los repositorios basada en el uso de los últimos estándares de compartición de metadatos (XMI) y de metamodelización (MOF). Desarrolla su actividad dentro del grupo de investigación Alarcos de la UCLM.

Francisco Ruiz es Licenciado en Ciencias por la Univ. Complutense de Madrid. Actualmente realiza su tesis doctoral en entornos integrados para el mantenimiento del software. Es Profesor Titular de Escuela Universitaria en el Dep. de Informática de la UCLM. Ha publicado cinco libros y tiene treinta publicaciones en congresos, conferencias y revistas internacionales sobre ingeniería del software y bases de datos. Pertenecer a diversas asociaciones científicas y profesionales (ACM, IEEE-CS, ATI, AEC, AENOR, ISO JTC1/SC7 y ACTA). Está interesado en modelado de proceso software y en gestión de proyectos software.

Mario Piattini es Licenciado y Doctor en Informática por la Univ. Politécnica de Madrid. CISA por la ISACA (Information System Audit and Control Association). Es Profesor Titular en el Dep. de Informática de la UCLM. Autor de varios libros y publicaciones sobre bases de datos, ingeniería del software y sistemas de información. Es líder del Grupo Alarcos de investigación, localizado en la Escuela Superior de Informática de la UCLM, en Ciudad Real (España). Sus temas de interés son: diseño avanzado de bases de datos, calidad en bases de datos, métricas software, métricas orientadas a objetos y mantenimiento del software.

Macario Polo es Profesor Ayudante en el Dep. de Informática de la UCLM. Es Licenciado en Informática por la Univ. De Sevilla y Doctor en Informática por la UCLM. Sus actuales temas de trabajo incluyen mantenimiento del software y mejora de procesos.