



**Jornadas Iberoamericanas de Ingeniería de Requisitos y Ambientes de Software
Workshop Iberoamericano de Engenharia de Requisitos e Ambientes de Software**

Santo Domingo, Heredia, Costa Rica
del 3 al 6 de abril del 2001

Memorias

Edición

Francisco Torres Rojas
José E. Araya Monge
Yarima Sandoval Sánchez

Apoyo

Proyecto CYTED/IDEAS

Realización

Instituto Tecnológico de Costa Rica
Departamento de Ingeniería en Computación

Presentación

Estimados colegas

IDEAS'2001 marca la cuarta ocasión en que se realizan las Jornadas Iberoamericanas de Ingeniería de Requisitos y Ambientes de Software. En particular, es la segunda vez que se realizan en Costa Rica lo cual es un motivo de orgullo para el Comité Organizador y para el Instituto Tecnológico de Costa Rica como institución anfitriona.

Estas jornadas pretenden constituir un foro dedicado tanto a la presentación de nuevos resultados de investigación como a la identificación de nuevas direcciones a explorar dentro del ámbito de Ingeniería de Software.

En respuesta a la solicitud del Comité del Programa, 75 trabajos fueron recibidos y 34 de estos fueron seleccionados para su presentación pública e inclusión en este volumen. Los criterios de selección incluyeron originalidad, importancia y calidad.

Deseamos expresar nuestro agradecimiento, en primer lugar, a los autores de las ponencias y al Comité de Programa por su excelente labor. También va nuestro profundo agradecimiento a todas las instituciones y empresas patrocinadoras de esta actividad.

Dr. Francisco Torres Rojas
Presidente del Comité Organizador

Instituto Tecnológico de Costa Rica
Editorial: Centro de Información Tecnológica (CIT)
Serie Manuales Didácticos, No. 11

Telefax: +506 551-6343
E-mail: jcarvajal@itcr.ac.cr
P.O. Box: 159-7050 Cartago, C.R.

ISBN: 9968-32-000-

005.3

J 82 i-2

Jornadas Iberoamericanas de Ingeniería de Requisitos de Ambientes de Software (4ª Sto. Domingo, San José, C.R.: 2001 - abr. 3-6)

IDEAS'2001: memorias/Francisco Torres Rojas, José Enrique Araya Monge y Yarima Sandoval Sánchez, eds.--Cartago, Costa Rica: Instituto Tecnológico de Costa Rica. Centro de Información Tecnológica (CIT), 2001.
p.: ils.--(Serie manuales didácticos, no. 11)

Correo electrónico de los editores: torres@ic-itcr.ac.cr, jaraya@ic-itcr.ac.cr
ysandoval@ic-itcr.ac.cr

ISBN: 9968-32-000-

Incluye ponencias en español, portugués e inglés.

A la cabeza de la portada: 4º Workshop Iberoamericano de Engenharia de Requisitos e Ambientes de Software.

1. Ingeniería de Software. 2. Ingeniería de Requisitos. 3. Tecnología Software Orientada a Objetos. 4. Métodos Formales en Ingeniería del Software.
5. Ambientes de Desarrollo de Software. 6. Patrones de Análisis y Diseño.
7. Desarrollo Basado en Componentes. 8. Tecnología del Software para Sistemas Concurrentes y Distribuidos. 9. Sistemas Concurrentes. 10. Sistemas Distribuidos. 11. Tecnología Software para Aplicaciones Web. 12. Congresos.
13. Software. 14. Internet. 15. World Wide Web. 16. Bases de Datos.
17. Costa Rica.

Esta actividad fue patrocinada por: El Programa Iberoamericano de Ciencia y Tecnología para el desarrollo (CYTED), Proyecto IDEAS, la Red Iberoamericana de Tecnología de Software (RITOS) y el Instituto Tecnológico de Costa Rica. Abril del año 2001.

Tabla de Contenidos

<i>Visualization of Internal Functions of an Enterprise in Systems Designs</i> C. G. M. Johansson.....	1
<i>Influencia de la Sentencia SELECT en el Mantenimiento de Programas SQL</i> A. Martinez, M. Piattini.....	12
<i>A Formal Model for Semantic Statements of Analysis Patterns</i> A. Cechich.....	22
<i>Process Diagrams: An Extension of UML Activity Diagrams for the Modelling of Software Processes</i> H. Oktaba, C. Alquicira.....	31
<i>Bancos de Dados Heterogêneos: Definindo um Modelo para Consulta Através de Ambiente Web</i> D. Moreira, L. P. Streck, E. Kroth.....	41
<i>Método para Engenharia de Requisitos Norteados por Necessidades de Informação</i> A. Novelli, M. A. Grigas.....	52
<i>Un Framework para la Consulta e Integración de Fuentes de Datos Heterogéneas y Distribuidas sobre la Web</i> J. F. Aldana, M. M. Roldán, A. C. Gómez, M. I. Yagüe.....	65
<i>XML Schema, RDF(S) e UML: uma Comparação</i> V. Braganholo, C. A. Heuser.....	78
<i>Evaluación de Metodologías para el Diseño de WIS: La propuesta de un Framework</i> L. Cernuzzi, M. González.....	91
<i>Reutilización del Software a partir de Requisitos Funcionales en el Modelo de Mecano: Comparación de Escenarios.</i> O. López, M. A. Laguna, J. M. Marqués.....	104
<i>Tratamiento de Atributos Derivados en Entornos de Producción Automática de Software Orientado a Objetos</i> M. Albert, E. Campos, V. Pelechano, O. Pastor.....	117
* <i>Metamodelos y Flujos de Trabajo para la Gestión de Proceso de Mantenimiento del Software</i> F. Ruiz, M. Piattini, F. García, M. Polo.....	132
<i>Lenguaje Visual para Asignar Comportamiento Complejo a Objetos de Mundos Virtuales Distribuidos con base al Modelo Cuerpo-Alma</i> L. Balladares, R. Menchaca, R. Quintero.....	143
<i>Propuesta para Integrar Bases de Datos que Contienen Información de la Web</i> A. do Carmo, R. Motz.....	156
<i>Una Aproximación Metodológica para Obtener Prototipos de Interface de Usuario a partir de Modelos de Requisitos</i> J. Sánchez, R. Vañó, M. Burbano.....	169
<i>Ingeniería de Requisitos Aplicada al Modelado Conceptual de Interfaz de Usuario</i> E. Insfrán, P. J. Molina, S. Martí, V. Pelechano.....	181
<i>Um Método Baseado em Requiritos para Seleção de COTS</i> C. Frota, J. Castro.....	193
<i>Efficient Assessment of Requirements Management Tools</i> R. Plösch.....	209

Metamodelos y Flujos de Trabajo para la Gestión del Proceso de Mantenimiento del Software ¹

Francisco Ruiz, Mario Piattini, Felix García, Macario Polo
Grupo Alarcos
Escuela Superior de Informática, Universidad de Castilla-La Mancha
Ronda de Calatrava, s/n. 13071, Ciudad Real (España)
{fruiz|mpiattin|fgarcia|mpolo}@inf-cr.uclm.es

Resumen

Los Entornos de Ingeniería del Software (EIS) tienen como objetivo poder manejar y controlar, de forma integrada y automática, los datos y actividades de un determinado proceso o grupo de procesos del software [1]. En este trabajo proponemos una arquitectura multinivel para la construcción de un EIS para la gestión integral del Proceso de Mantenimiento del Software (PMS). Esta arquitectura de 4 niveles permite representar los diferentes datos y meta-datos utilizados en la gestión del PMS mediante el modelado de conceptos a diferentes niveles de abstracción: meta-metamodelos genéricos, metamodelos de procesos software, procesos software concretos (en nuestro caso, el PMS), e instancias de ejecución de un proceso determinado (un proyecto concreto). Con dicha arquitectura, el EIS puede incorporar conceptual y tecnológicamente nuevos servicios orientados a los procesos organizativos que dan soporte al mantenimiento (especialmente los de Gestión y Mejora), de acuerdo al modelo de procesos del ciclo de vida de ISO 12207 [2].

1. Introducción.

En los últimos años, la comunidad científica internacional ha reconocido la importancia - especialmente económica - que tiene el mantenimiento del software. Diversos estudios han demostrado que esta etapa supone la mayor parte de los costes totales de un producto software a lo largo de su ciclo de vida [3] y que sus características especiales, que lo diferencian significativamente del desarrollo, hacen muy útil disponer de métodos, técnicas y herramientas específicos [4].

Muy recientemente ha sido propuesto un nuevo modelo de ciclo de vida del software en el cual el mantenimiento se representa por varias etapas (evolución, servicio, retirada, y cierre) que "siguen" a una etapa de desarrollo inicial [5]. Cada etapa tiene sus propias actividades, herramientas y consecuencias de negocio. Esto supone el reconocimiento implícito de la importancia, diferencias, tipos y dificultades del mantenimiento.

Por las razones citadas, últimamente se han llevado a cabo esfuerzos para abordar el problema del mantenimiento del software teniendo en cuenta sus especificidades. En esta línea se encuentran las metodologías específicas para el mantenimiento, como MANTEMA [6], o propuestas sobre la mejora del PMS como la de [7], que sugiere dos métodos: mejora basada en la medida; y mejora basada en la capacidad de la organización (para lo que propone una versión adaptada de CMM).

Multitud de empresas en todo el mundo se enfrentan al reto de tener que dar servicio de mantenimiento del software a otros departamentos o a otras organizaciones. Para ello, deben abordar "proyectos de mantenimiento" que, por las causas ya comentadas, se deberían llevar a cabo desde una perspectiva

¹ Este trabajo ha sido parcialmente financiado con cargo a los proyectos "MANTIS: Entorno para el Mantenimiento Integral del Software" (CICYT-FEDER 1FD97-1608TIC, de la Unión Europea y la Comisión Interministerial de Ciencia y Tecnología de España), y "MPM: Mejora del Proceso de Mantenimiento" (TA15-1999, programa ATYCA del Ministerio de Industria y Energía de España y Atos ODS).

específica, aunque reutilizado lo común con los proyectos de desarrollo de software (por ejemplo, técnicas de prueba, normas de documentación, etc.). Estos proyectos de mantenimiento suelen tener un presupuesto considerable (debido a la complejidad y tamaño del producto a mantener y la dificultad de la tarea a realizar). Por ello, es muy importante para tener garantías de éxito utilizar al máximo posible las "buenas prácticas" de la ingeniería del software y de la gestión de proyectos.

Entre estas buenas prácticas se encuentra el uso de herramientas adecuadas para realizar las diferentes tareas del proyecto, tanto las específicas de desarrollo o mantenimiento del software (CASE genéricos, depuradores, analizadores de código, etc.), como las orientadas a la gestión y mejor ejecución del proyecto. El uso de manera integrada de todas estas posibles herramientas es de gran utilidad porque aumenta la productividad, reduce la posibilidad de errores y facilita la gestión, supervisión y control.

Buscando estos objetivos se han formulado diversas propuestas: PCTE "Portable Common Tool Environment" [8], el modelo de referencia para EIS de ECMA/NIST [9], o la propuesta STARS "Software Technology for Adaptable, Reliable Systems" de la USAF [10].

Pero ninguna de las propuestas formuladas tiene en cuenta las características especiales del mantenimiento del software, si bien sus aportaciones genéricas y modelos arquitecturales pueden ser aplicables a este proceso en particular. El proyecto MANTIS intenta cubrir esta carencia mediante la definición y construcción de un entorno para la gestión integral del PMS. En MANTIS se define la forma en que se organiza, gestiona, mide, soporta y mejora el PMS, reutilizando para el mantenimiento la definición de proceso software de Demniamé y otros [11]. El presente trabajo se enmarca dentro del proyecto MANTIS.

En MANTIS se abordan los diferentes aspectos que hay que tener en cuenta cuando se realizan proyectos de mantenimiento de software, integrándolos conceptualmente con una perspectiva de tipo "Big-E Environment" (entorno con mayúsculas) similar a la propuesta de Cockburn [12]. La idea es más amplia que los siguientes conceptos, a los cuales integra:

- Metodologías en el sentido habitual, es decir, una serie de métodos o técnicas relacionados.
- Entornos de Ingeniería del Software (EIS), es decir, una colección de herramientas software utilizadas para soportar actividades de ingeniería del software [1].

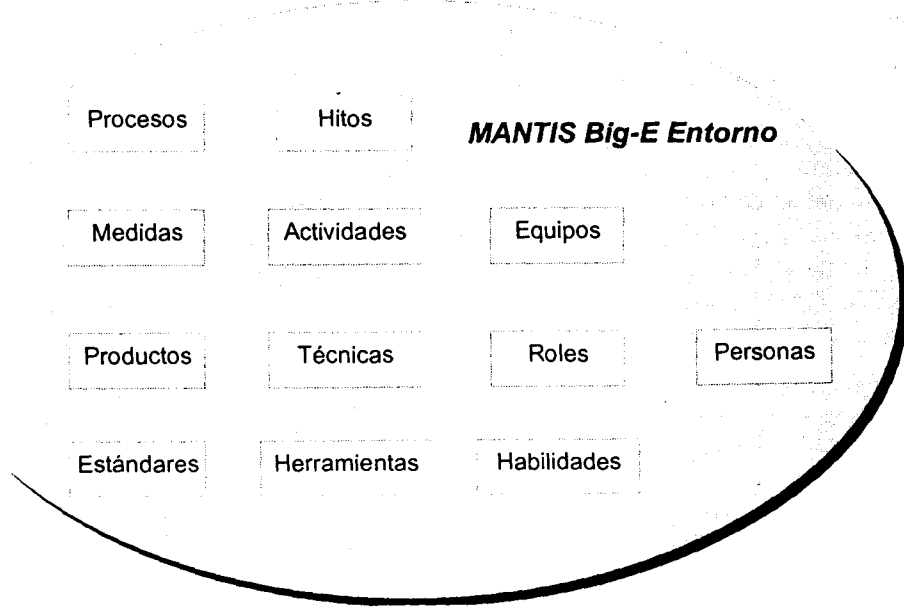


Figura 1. Aspectos considerados en el Big-E Entorno MANTIS.

El "Big-E Entorno" de MANTIS para gestionar proyectos de mantenimiento del software tiene en cuenta, entre otros, los aspectos siguientes (ver figura 1):

- Las personas con ciertas habilidades desempeñan ciertos roles en el proyecto trabajando juntas en diversos equipos (grupos de personas).
- Utilizan técnicas (metodologías) para construir productos que siguen ciertos estándares (normas) y satisfacen medidas (criterios) de calidad. Los procesos también deben satisfacer criterios de calidad.
- Las técnicas requieren ciertas habilidades y herramientas; las herramientas ayudan a cumplir los estándares.
- Los equipos participan en actividades que pertenecen a procesos englobados dentro del proyecto; cada actividad realizada ayuda a alcanzar hitos significativos que indican cómo avanzan los procesos.

Para poder abordar todos estos aspectos, hemos definido los siguientes requerimientos generales:

- Orientación a las herramientas: los servicios de MANTIS son provistos mediante herramientas software siguiendo la filosofía de los EIS's integrados.
- Dirigido a procesos: asegurando la consistencia con los procesos y actividades descritos en los estándares ISO 12207, de procesos del ciclo de vida del software [2], y 14764, del PMS [13].
- Escalabilidad: adaptación a las necesidades de organizaciones pequeñas, medianas y grandes.
- Adopción de una metodología específica para el PMS, en particular MANTEMA, [14] [6].
- Soporte al trabajo técnico (el PMS propiamente dicho) y a las actividades organizativas y de gestión.
- Utilización de la tecnología basada en el estándar XML [15] y sus derivados (XMI, ...) para el repositorio de datos y metadatos.

Dada la gran cantidad de aspectos a considerar, se hace necesario utilizar varios niveles conceptuales para poder abordar mejor la complejidad de MANTIS. En el resto del trabajo realizamos una presentación de los niveles conceptuales de MANTIS y explicamos sus ventajas. En el apartado 2 se introduce de forma general la arquitectura conceptual de 4 niveles utilizada. En el apartado 3 se comentan los tres niveles inferiores: el metamodelo genérico de PMS utilizado (nivel M2), el modelo concreto asociado en el nivel M1, y los ejemplares o instancias de ejecución de proyectos, correspondientes al nivel M0. En el apartado 4 se amplía y detalla el uso de Flujos de Trabajo (del inglés *WorkFlow*) en el metamodelo del PMS para incorporar los aspectos de reificación². En el último apartado se comentan las principales conclusiones y líneas de trabajo en curso y futuras.

2. Niveles conceptuales de MANTIS.

Un principio importante de la ingeniería del software moderna es la separación de un sistema en capas de encapsulación, que pueden especificarse, diseñarse y llevarse a cabo de manera independiente en gran parte. Estas capas o niveles conceptuales ayudan a:

- ganar control intelectual del sistema dividiéndolo en abstracciones significativas;
- reducir la complejidad de construcción del sistema al dividirlo en piezas separables que pueden construirse de forma independiente por un número relativamente pequeño de especialistas;
- reducir los costes de mantenimiento al poder localizar más fácilmente el impacto de muchos cambios;
- facilitar la evolución del sistema al posibilitar la sustitución de capas completas de funcionalidad; y
- aumentar la generalidad del sistema al permitir abstracciones más generales en los niveles superiores.

En MANTIS, la orientación a procesos es una tesis fundamental, ya que supone un vehículo de integración muy importante [16]. Con ello se pueden abordar dos formas de integración simultáneamente: integración de los procesos de la organización con el EIS, e integración de las herramientas y artefactos en el contexto de los procesos. Esta orientación es conocida en la bibliografía como PSEE (*Process Sensitive Software Engineering Environment*).

² del inglés *enactment*, "transformación de conceptos abstractos en realidades u objetos concretos"; también se utilizan las expresiones animación o ejecución, aunque esta última no se corresponde exactamente con el significado del término

Los niveles conceptuales de MANTIS están basados en el estándar MOF (*Meta-Object Facility*) para metamodelado con orientación a objetos, propuesto por el *Object Management Group*, OMG [17]. La arquitectura MOF es similar a las arquitecturas de varios niveles de abstracción utilizadas en otros ámbitos (por ejemplo, el estándar ISO 10027 de Sistemas de Diccionarios de Recursos de Información). Los cuatro niveles arquitecturales de MOF son:

- M0: Nivel de Datos. Contiene los datos que necesitan ser descritos (ejemplo: datos de clientes y facturas).
- M1: Nivel de Modelos. Contiene los metadatos que describen los datos (ejemplo: el diagrama entidad-interrelación de la base de datos de clientes y facturas).
- M2: Nivel de Meta-Modelos. Contiene los metadatos que definen meta-modelos, es decir, lenguajes abstractos en los que se expresan los metadatos del nivel M1 (ejemplo: los conceptos utilizados en el modelo entidad-interrelación).
- M3: Nivel de Meta-Metamodelos. Contiene el lenguaje abstracto utilizado para expresar modelos MOF (siempre son los mismos conceptos independientemente del dominio de aplicación).

Entre los elementos de un nivel MOF y los del nivel inmediatamente inferior existen correspondencias del tipo "es una instancia de", de forma que un elemento del nivel Mi se puede considerar como una clase cuyos ejemplares son elementos del nivel Mi-1.

En el dominio de aplicación de MANTIS, es decir, en la gestión del PMS, los cuatro niveles conceptuales se corresponden con los modelos-MOF (M3), metamodelo genérico del PMS basado en una ontología del mantenimiento y el uso de flujos de trabajo (M2), modelo concreto del PMS (M1) basado en la metodología MANTEMA principalmente, y ejemplares o instancias de ejecución de proyectos reales de mantenimiento que se están llevando a cabo (M0), ver tabla 1.

Tabla 1. Niveles conceptuales MOF vs 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 ejecución del PMS (proyectos concretos de mantenimiento del software)

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 MANTIS, 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*) [18], basado en XML (*eXtensible Markup Language*), para almacenar los metamodelos.

3. Metamodelo, modelo y ejemplares del proceso de mantenimiento del software.

Aunque MANTIS y, en particular, su metodología asociada MANTEMA utilizan el modelo de procesos software de ISO (12207 y 14764), el carácter documental de estos estándares, y por tanto, su poca estructuración y formalización, hace que no sea fácil ni evidente extraer el meta-modelo de proceso software genérico o el modelo de PMS subyacentes [19]. Por esta razón, para definir el metamodelo del PMS (nivel M2), en MANTIS hemos optado por utilizar la propuesta de ontología informal para el mantenimiento del software formulada por Kitchenham y otros [20], ver figura 2.

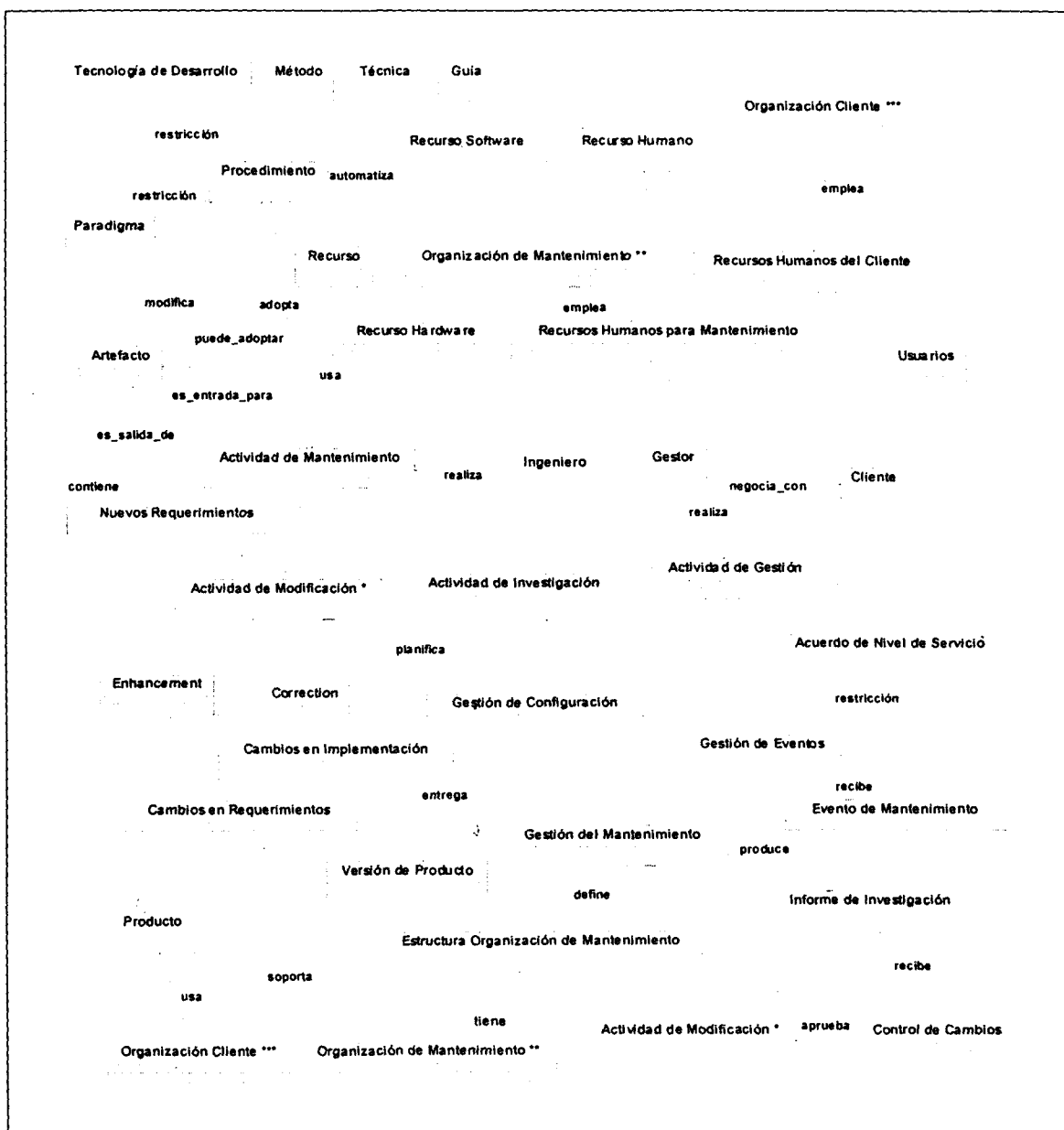


Figura 2. Vista resumida del metamodelo del PMS, expresada en UML.

Este metamodelo global está construido por la unión de cuatro metamodelos parciales centrados en las Actividades, los Productos, las Personas (*peopleware*) y el Proceso (*process organization*). Dicho muy brevemente, el modelo representa:

- cómo organizar actividades para mantener productos software,
- de que tipos pueden ser dichas actividades,
- cómo los métodos y herramientas (específicos o compartidos con el proceso de desarrollo) pueden ser aplicados a las actividades, y
- qué habilidades y roles son necesarios para llevar a cabo las actividades.

A pesar de la complejidad de este metamodelo, para contemplar todos los aspectos de MANTIS, es decir, del Big-E Entorno ya comentado, es necesario incorporar algunos elementos más, especialmente los referidos a la medida y sus relaciones. En MANTIS, los aspectos de medida están basados en una adaptación propia que contempla las dimensiones dinámica (desarrollo del proyecto) y estática (modelado del proceso), ampliando la propuesta al respecto del IESE [21].

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 [22], etc. En el nivel M0 están los ejemplares (instancias de ejecución) de proyectos reales y concretos de mantenimiento del software con restricciones de tiempo, costes, etc. Por ejemplo, el concepto genérico de "Actividad" utilizado en M2, se instancia en los tipos de actividades "Análisis de Informe de Problemas" o "Implementación de la Modificación" en M1, y éstas a su vez, en instancias de ejecución del nivel M0 como "Análisis de informe del problema nº 36 del proyecto PATON".

4. Flujos de trabajo para metamodelar el PMS.

El PMS es un tipo especial de proceso software (PS) y, por tanto, para diseñar su metamodelo pueden ser útiles las diversas propuestas de modelos de procesos software en general:

- modelo de PS del IESE [21];
- PROMENADE, *Process-Oriented Modelling and Enactment of Software Developments*, [23];
- APEL, *Abstract Process Engine Language*, [24];

.....

Recientemente, algunos autores [25] han sugerido la posibilidad de utilizar Flujos de Trabajo (FT's) para abordar los procesos software, sacando partido de la similitud existente entre ambas tecnologías. La utilidad de los Sistemas de Gestión de Flujos de Trabajo (SGFT), del inglés *Workflow Management Systems*, en la automatización de procesos de negocio ha sido demostrada con creces y, puesto que el PMS puede ser considerado como parte de un proceso de negocio más amplio³, es razonable considerar que la tecnología de FT's es capaz de aportar al PMS una perspectiva más amplia (que podríamos llamar Tecnología de Procesos) en línea con el Big-E Entorno de MANTIS. Este planteamiento ha sido confirmado por algunos desarrollos concretos de herramientas basadas en FT's para dar soporte al PS en general [26].

Las razones anteriores nos han llevado a utilizar FT's para ampliar el metamodelo del PMS, incorporando los aspectos de reificación, en lugar de optar por alguna de las alternativas de modelos de PS genéricos. Por esta causa, en el nivel conceptual M2 de MANTIS se utilizan los conceptos de FT's según el modelo de referencia de la *Workflow Management Coalition* [27]. Para la WfMC un FT es "la automatización, total o parcial, de un proceso de negocio", es decir, la representación del mismo en un formato entendible por las computadoras. En MANTIS el proceso de negocio está definido por el Big-E Entorno (generalizando el PMS de ISO, que sólo considera los aspectos técnicos).

Al igual que los Sistemas de Gestión de Bases Datos son herramientas software para ayudar a manejar bases de datos, los SGFT's son herramientas para utilizar los FT's. De acuerdo con la WfMC [28], un SGFT es

³ En ambos casos se habla de actividades que forman el proceso, de artefactos (productos, documentos o datos) que utilizan, modifican o producen dichas actividades, de personas que participan desempeñando ciertos roles, de herramientas utilizadas (por ejemplo, aplicaciones software), de aspectos de colaboración en equipos de trabajo, etc.

"un sistema que define, crea y gestiona la ejecución de FT's mediante el uso de software, siendo capaz de interpretar la definición del proceso, interactuar con los participantes y, cuando es requerido, invocar el uso de herramientas y aplicaciones".

En el uso de SGFT's existen dos fases claramente diferenciadas:

- a) Fase de Diseño (*Design-Time*), referida al modelado conceptual del FT, y
- b) Fase de Ejecución (*Run-Time*), referida a la reificación del FT.

Para gestionar proyectos de mantenimiento de software también debemos tener en cuenta ambas fases y, por tanto, en MANTIS se deben incluir los aspectos de reificación de FT's junto a los conceptuales que metamodelan el PMS. Con este objetivo, en el nivel M2 hemos construido un modelo de FT's basado en las propuestas de la WfMC y en aportaciones de otros autores [29], pero adaptado para contemplar los aspectos del Big-E Entorno de MANTIS. Este modelo se muestra en la figura 3.



Figura 3. Metamodelo de flujos de trabajo en el nivel M2 de MANTIS.

En la figura anterior están colocados en la zona izquierda los elementos relacionados con la fase de diseño, y en la zona derecha los relacionados con la fase de ejecución. Las clases "SGFT" y "Actor" están compartidas por ambas fases (en el sentido de reutilización de paquetes UML), mientras que las clases "Instancia de FT", "Instancia de Actividad", "Historia de Instancias de FT", e "Item de Trabajo" son específicas de la Fase de Ejecución. A continuación comentaremos con un poco más de detalle la parte donde se modela la "Especificación del FT", es decir, la estructura de actividades que lo forman y sus interrelaciones.

Tradicionalmente, una "Especificación de FT" se ha representado utilizando diagramas de flujo como el de la figura 4. De esta forma, una "Especificación" está formada por un conjunto de "Nodos" (figuras geométricas) interconectados mediante "Flujos de control" (flechas). Los nodos pueden ser "Actividades" (rectángulos) o "Condiciones de Transición" (círculos). Una condición puede ser de tipo "Or-split" o de tipo "Or-join". Las condiciones Or-split y Or-join permiten representar bifurcaciones y fusiones, es decir, caminos de ejecución opcionales. Para representar caminos de ejecución concurrentes se utilizan actividades con más de un flujo de control empezando en ellas (inicio de la concurrencia) o más de un flujo de control acabando en ellas (final de la concurrencia o sincronización).

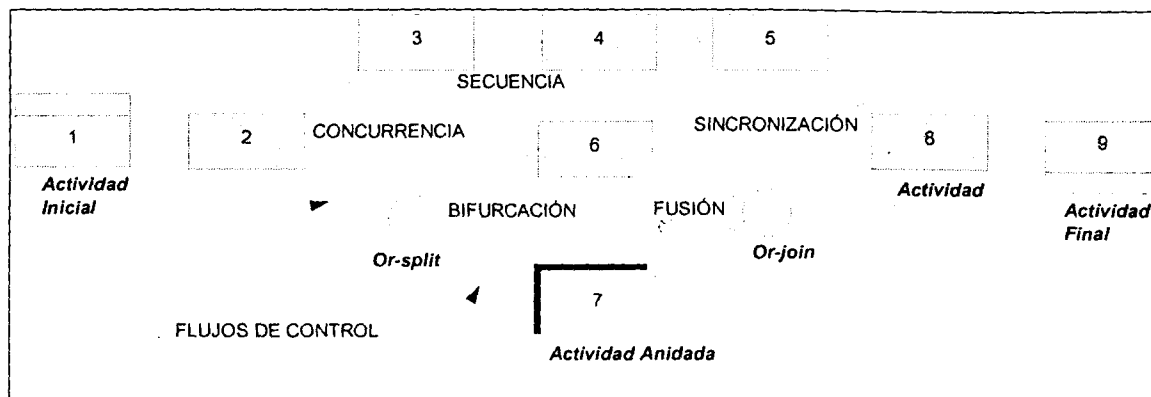


Figura 4. Ejemplo de diagrama de flujo de trabajo.

En el modelo de FT's de MANTIS, se utilizan las clases "Especificación de FT", "Nodo del diagrama", "Flujo de control", "Condición de transición" y "Actividad" para modelar lo anterior. Con respecto al resto del modelo de la figura 3, cabe destacar que la clase "Especificación de tipo de tarea" contiene las propiedades que se pueden abstraer de una actividad y que por tanto son independientes del flujo de trabajo, es decir, incluye las propiedades generales de un conjunto de actividades del mismo tipo. Además, una especificación de tipo de tarea puede ser de tipo simple (no descomponible en otras tareas) o anidada. Puesto que una tarea anidada tiene una estructura interna igual que un FT, puede ser representada - utilizando el mecanismo de recursividad - también mediante una "Especificación de FT". La ejecución de esta tarea supondrá la ejecución del FT subyacente en la misma.

En MANTIS, la realización de un proyecto concreto y real de mantenimiento de software supondrá:

- en la fase de diseño, el modelado conceptual del PMS mediante la especificación de "su" FT asociado; y
- en la fase de ejecución, la reificación de dicho FT creando los ejemplares adecuados de las clases "Instancia de FT", "Instancia de Actividad" e "Item de Trabajo" y los ejemplares de las asociaciones entre ellos y con la clase "Actor" (estas últimas son para asignar ítems de trabajo a personas, equipos o aplicaciones).

En relación con el último nivel conceptual de MANTIS, el M3, todos los conceptos representados en la figura 3 se tratan como instancias de clase-MOF o de asociación-MOF. Por ejemplo, "Especificación de Flujo de Trabajo", "Actividad", "Tarea simple", "Actor" son ejemplares de clase-MOF; y "Tarea simple Utiliza Recurso de procesamiento" o "Flujo de control Empieza en Nodo de diagrama" son ejemplares de asociación-MOF.

Con la incorporación del nivel conceptual M3 y con la utilización, como ya se ha comentado, de estándares de metamodelado (MOF) y de intercambio de metadatos (XMI) es posible conseguir un entorno con la mayor flexibilidad para definir y compartir datos y metadatos (modelos y metamodelos). A título de ejemplo, en la figura 5 se muestra un fragmento de documento XMI correspondiente a la parte del modelo-MOF que representa la clase "Nodo del Diagrama" de la figura 3.

```

<?xml version = '1.0' encoding = 'ISO-8859-1' ?>
<!DOCTYPE XMI SYSTEM 'Model.dtd' >
<XMI xmi.version = '1.0'>
<XMI.header>
....
</XMI.header>
<XMI.content>
....
<!-- ===== Meta::Nodo del diagrama[Class] ===== -->
  <Model.Class xmi.id = 'a3A2606240370'>
    <Model.ModelElement.name>Nodo del diagrama</Model.ModelElement.name>
    <Model.ModelElement.annotation></Model.ModelElement.annotation>
    <Model.GeneralizableElement.isRoot xmi.value = "false"/>
    <Model.GeneralizableElement.isLeaf xmi.value = "false"/>
    <Model.GeneralizableElement.isAbstract xmi.value = "false"/>
    <Model.GeneralizableElement.visibility xmi.value = "public_vis"/>
    <Model.Class.isSingleton xmi.value = "false"/>
    <Model.ModelElement.container>
      <Model.Namespace xmi.idref = 'a3A26049B01F4' /> <!-- Meta -->
    </Model.ModelElement.container>
    <Model.Namespace.contents>
      ....
    </Model.Namespace.contents>
  </Model.Class>
....
</XMI.content>
</XMI>

```

Figura 5. Fragmento de modelo-MOF en XMI.

5. Conclusiones y líneas de trabajo.

Para gestionar cualquier proceso software - como el PMS - de una manera integrada y estructurada es muy útil considerar diversos niveles de abstracción, tanto para modelar el proceso, como para gestionar las diversas instancias de ejecución (proyectos concretos).

En este trabajo presentamos MANTIS, un entorno integral para la gestión del mantenimiento del software, cuyas principales características son:

- una arquitectura conceptual en 4 niveles;
- un modelo del PMS basado en una propuesta ontológica del mantenimiento ampliada con aspectos de medida y otros;
- utilización de FT's para ampliar el metamodelo del PMS con aspectos de reificación;
- aprovechamiento del modelo de referencia de SGFT's para el modelado conceptual de los FT's asociados a proyectos de mantenimiento, y para la reificación de dichos FT's;
- aprovechamiento de las ventajas del uso de estándares de metamodelado (MOF) y de intercambio de metadatos (XMI); y
- aplicación de esta arquitectura para la construcción de un EIS orientado a procesos.

Con esta arquitectura es posible disponer de un marco conceptual y una herramienta software para gestionar mejor los proyectos de mantenimiento de una organización.


Entre las líneas de trabajo en curso y futuras, merecen destacarse las siguientes:

- a) Incorporar al modelo de PMS la subontología de la Medida, fundamental para contemplar adecuadamente procesos de soporte como el de auditoría (Ruiz et al, 2000), u organizativos como el de Gestión y el de Mejora. La subontología de la Medida integra en el modelo global los conceptos de medida, métrica, valor, y atributos asociados a las actividades, artefactos y recursos.

- b) Construcción de una herramienta horizontal que permita definir desde el nivel M3 hasta el M1 todos los conceptos manejados. Esta herramienta se debe poder comunicar con herramientas comerciales de SGFT mediante los estándares comentados.

6. Referencias.

- [1] ISO/IEC JTC1/SC7/WG4 15940 working draft 5: *Information Technology – Software Engineering Environment Services*, Juny-2000.
- [2] ISO/IEC 12207: *Information Technology – Software Life Cycle Processes*, 1995.
- [3] Pigoski, T.M., *Practical Software Maintenance. Best Practices for Managing your Investment*. Ed. John Wiley & Sons, USA, 1996.
- [4] 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, 1998.
- [5] Rajlich, V.T. y Bennett, K.H., "A Staged Model for the Software Life Cycle". *IEEE Computer*, July 2000, pp. 66-71.
- [6] Polo, M., Piattini, M., Ruiz, F. y Calero, C. "MANTEMA: A Software Maintenance Methodology based on the ISO/IEC 12207 Standard". *Proceedings of the 4th IEEE International Software Engineering Standards Symposium*. IEEE Computer Society Press, Curitiba (Brazil) 1999, pp. 76-81.
- [7] Niessink, F. *Perspectives on Improving Software Maintenance*. PhD Thesis, Vrije Universiteit, Netherland. In <http://www.opencontent.org/openpub/>, 2000.
- [8] Long, F., Morris, E., *An Overview of PCTE: A Basis for a Portable Common Tool Environment*. Technical Report CMU/SEI-93-TR-1, 1993.
- [9] ECMA/NIST *Reference Model for Frameworks of Software Engineering Environments*, 3rd edition. TR-55, June 1993. In <http://www.ecma.ch>.
- [10] US Air Force: *Software Engineering Environment Integration Process. Summary-level Definition*. Informal TR STARS-PV03-A032/001/00, 1996.
- [11] Derniame, J-C.; Kaba, B.A.; y Wastell, D. (Eds.); *Software Process: Principles, Methodology and Technology*. LNCS 1500. Springer-Verlag, 1999.
- [12] Cockburn, A., "Selecting a Project's Methodology". *IEEE Software*, July/August 2000, pp. 64-71.
- [13] ISO/IEC FDIS 14764: *Software Engineering - Software Maintenance* (draft), Dec-1998.
- [14] 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.
- [15] W3C *Extensible Markup Language (XML) 1.0* (second edition), Oct-2000. In <http://www.w3.org/>.
- [16] Randall, R. y Ett, W., "Using Process to Integrate Software Engineering Environments". *Proceedings of the Software Technology Conference*, Salt Lake City, USA, 1995.
- [17] OMG *Meta Object Facility (MOF) Specification*, v. 1.3 RTF, Sep-1999. In <http://www.omg.org>.
- [18] OMG *XML Metadata Interchange (XMI)*, v. 1.1, Oct-1999.
- [19] Blanck, C., "Transcribing Process Model Standards into Meta-Processes". *Proceedings of the 7th European Workshop on Software Process Technology*, pp. 55-68, LNCS 1780, Springer Verlag, 2000.
- [20] Kitchenham, B.A.; Travassos, G.H.; Mayrhauser, A. von; Niessink, F.; Schneidewind, N.F.; Singer, J.; Takada, S.; Vehvilainen, R. and Yang, H.; "Towards an Ontology of Software Maintenance". *Journal of Software Maintenance: Research and Practice*. 11, 365-389 (1999).
- [21] 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.
- [22] Ruiz, F.; Piattini, M.; Polo, M.; Calero, C.; "Audit of Software Maintenance". In Piattini (edit.), *Auditing Information Systems*, Idea Group Publishing, USA 2000, pp. 67-108.
- [23] Franch, X.; Ribó, J.M.; "Some Reflexions in the Modelling of Software Processes". *Proceedings of the International Process Technology Workshop*, Grenoble (Francia), 1999.
- [24] Dami, S.; Estublier, J.; Amiour, M.; "APEL: A Graphical Yet Executable Formalism for Process Modeling". *Automated Software Engineering*, 5(1): 61-96, January 1998.
- [25] Ocampo, C.; Botella, P.; *Some Reflections on Applying Workflow Technology to Software Processes*. TR-LSI-98-5-R, UPC, Barcelona, 1998.

- 
- [26] Penadés, M.C.; Canós, J.; Carsí, J.A.; "Hacia una Herramienta de Soporte al Proceso Software basada en la tecnología de Workflow". *IV Jornadas de Ingeniería del Software y Bases de Datos*; Cáceres, España, 1999.
 - [27] WfMC TC00-1003 1.1: Workflow Management Coalition. *The Workflow Reference Model*, Jan-1995.
 - [28] WfMC TC-1011 3.0: Workflow Management Coalition. *Terminology and Glossary*, Feb-1999.
 - [29] Liu, C.; Lin, X.; Zhou, X.; Orłowska, M.; "Building a Repository for Workflow Systems", *Proceedings of the 31st International Conference on Technology of Object-Oriented Language and Systems*. IEEE Computer Society Press, 1999, pp. 348-357.