

GERENCIA TECNOLOGICA

INFORMATICA

Informatics Technology Management

ITIS

Modelos ágiles de desarrollo software
Agile models of software development

Modelamiento y Diseño orientado a objetos
Object Oriented Design

Administración de la Configuración de software
Administration of Software Configuration

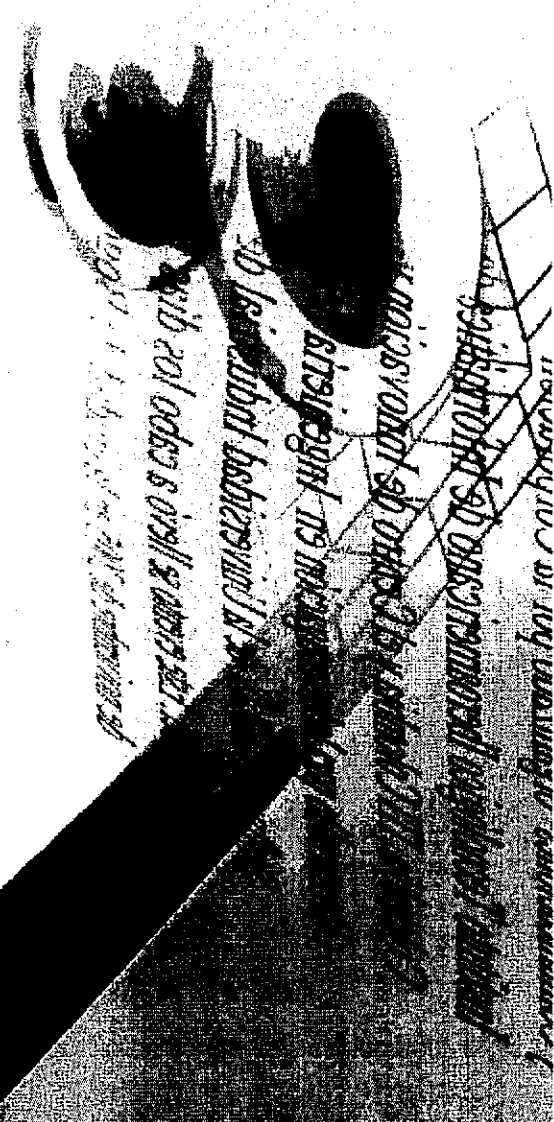
Modelos Educativos
Educative Models

Desarrollo curricular, modelos de conocimiento para la creación de contenidos de aprendizaje
knowledge models for the creation of learning contents

Estrategias de negocios y comercialización de software
Strategies of businesses and commercialization of software

ITI
Colombia
Instituto Tecnológico Interdisciplinario de Investigación en Ciencias

Universidad Industrial de Santander
CIDLIS



GERENCIA TECNOLOGICA INFORMÁTICA

Informatics Technology Management

DIRECTOR

Dr. Ricardo Llamosa Villalba
rllamos@iticol.org

COORDINADOR

Ing. Mónica Andrea Rico Martínez
mrico@cidlisuis.org

ADMINISTRADOR DE CONOCIMIENTO

Ing. Andres Guerrero
anguerrco@iticol.org

DISEÑO Y DIAGRAMACION

José Fernando Esquivia Gómez

TRADUCCIONES

Ing. Mónica Andrea Rico Martínez
mrico@cidlisuis.org

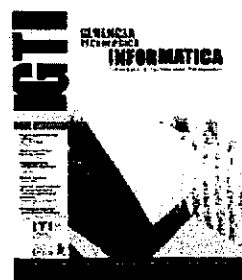
PORTADA E ILUSTRACIÓN

José Fernando Esquivia Gómez

IMPRESIÓN

CIDLIS - UIS

Universidad Industrial de Santander
Carrera 19 No. 35 - 02
Of. 330 - Tel. 670 10 62
652 01 23



Una Publicación de **ITI COLOMBIA**
Para el sector de la Investigación, académico y empresarial

Página Web: www.iticol.org
Periodicidad: Cuatrimestral

Número 10, Volumen 4
Fecha de Publicación: 06-12-2005
Contacto: revistagti@iticol.org

Publicidad y Comercialización

ITI Colombia
Cra 19 No. 35 - 02
Of. 330 - Tel. 6520123
Bucaramanga - Colombia

ISSN : 1657-8236

Indexada en :

PUBLINDEX. Índice Nacional de Publicaciones Serías, Científicas y Tecnológicas de Colombia
LATINDEX. Sistema Regional en Línea para revistas Científicas de América Latina, el Caribe, España y Portugal

La dirección de la revista GTI no se responsabiliza por el contenido de los artículos ni su publicación en otros medios. Se autoriza la reproducción total o parcial de la obra para fines educativos siempre y cuando se cite la fuente.

Índice

EDITORIAL	10
MODELO LIGERO PARA LA EVALUACIÓN DE PROCESOS SOFTWARE Francisco J. Pino, Felix García, Mario Piattini, Francisco Ruiz	11
CARACTERIZACIÓN Y MODELADO CON UML DE UNA HERRAMIENTA INFORMÁTICA PARA LA GENERACIÓN DE PORTALES DE CONOCIMIENTO ORIENTADO A GRUPOS Y CENTROS DE INVESTIGACIÓN UNIVERSITARIOS Carlos Andrés Guerrero	21
FACILITANDO EL ACCESO A LAS FUENTES DE INFORMACIÓN DE UN GRUPO DE MANTENIMIENTO DE SOFTWARE POR MEDIO DE UN MAPA DE CONOCIMIENTO Oscar Rodríguez, Ana Martínez, Jesús Faveia, Aurora Vizcaíno, Mario Piattini	31
METODOLOGÍA PARA LA ELABORACIÓN DE PROPUESTAS DE SISTEMAS DE ACTIVIDAD HUMANA ORIENTADA A CONSULTORIOS JURÍDICOS PARA EL DESARROLLO DE SU APRENDIZAJE ORGANIZACIONAL Victor Cárdenas, Jimena Vásquez	43
DISEÑO E IMPLEMENTACIÓN DE CURSOS VIRTUALES DE APRENDIZAJE BASADOS EN ESTÁNDARES EDUCATIVOS Daniel Salas, Juan Kelsy	53
DE REQUISITOS TEMPRANOS A REQUISITOS TARDÍOS: UN ENFOQUE BASADO EN METAS Alicia Martínez, Óscar Pastor	61

**LA ELICITACIÓN DE REQUISITOS EN
DESARROLLO GLOBAL DE SOFTWARE:
UN ENFOQUE COGNITIVO** **73**

Gabriela Aranda, Aurora Vizcaíno, Alejandra Cechich,
Mario Piattini

**GENERACIÓN DE MODELOS ORGANIZACIONALES
A PARTIR DE ESTRUCTURAS DE
REFINAMIENTO DE METAS** **81**

Oscar Pastor, Hugo Estrada

**MODELO INTELIGENTE PARA BASES DE
DATOS DISTRIBUIDAS** **91**

Ana Muñoz, José Aguilar, Rodrigo Martínez



FACILITANDO EL ACCESO A LAS FUENTES DE INFORMACIÓN DE UN GRUPO DE MANTENIMIENTO DE SOFTWARE POR MEDIO DE UN MAPA DE CONOCIMIENTO



AUTOR

Oscar M. Rodríguez Ellas

Magister en Ciencias de la Computación

Departamento de Ciencias de la Computación, CICESE

Docente Universidad de Sonora

Orodrigu@cicese.mx

MÉXICO

AUTOR

Aurora Vizcaíno Barceló

Doctor en Informática

Investigador Titular y Docente

Escuela Superior de Informática

Universidad de Castilla La Mancha

Aurora.Vizcaino@uclm.es

ESPAÑA

AUTOR

Ana I. Martínez García

Doctor en Ciencias de la Computación

Universidad de Manchester

Investigador Asociado y Docente

Departamento de Ciencias de la Computación

CICESE

martinea@cicese.mx

MÉXICO

AUTOR

Mario Piattini Velthuis

Doctor en Informática

Universidad Politécnica de Madrid

Investigador Titular y Docente

Universidad de Castilla La Mancha

Marlo.Piattini@uclm.es

ESPAÑA

AUTOR

Jesús Favela Vara

Doctor en Ciencias de la Computación

Universidad de Massachusetts

Investigador Titular y Docente

Departamento de Ciencias de la Computación,

CICESE

favela@cicese.mx

MÉXICO

Fecha de recepción del artículo: 4 de Noviembre de 2005
Artículo Tipo 2

Fecha de Aceptación del Artículo: 18 de Noviembre de 2005

RESUMEN.

Entre los problemas que enfrentan las organizaciones de desarrollo de software, y en particular los grupos encargados del mantenimiento de software, se encuentran la pérdida o desaprovechamiento del conocimiento que existe en las distintas fuentes disponibles dentro de estos grupos. Esto se produce debido a que con frecuencia esas fuentes son desconocidas o difíciles de localizar. Los Mapas de Conocimiento (MC) son una técnica que puede ser de utilidad para abordar este problema, dado que pueden ser usados para describir las fuentes de conocimiento disponibles, el conocimiento que puede ser obtenido de las mismas, y cómo éstas pueden ser localizadas. Un aspecto importante para la creación de un MC, es definir qué conocimiento y qué fuentes deberán ser tomadas en cuenta. Con este fin, la identificación del flujo del conocimiento dentro de una organización puede ayudar a identificar qué conocimiento es el requerido en las actividades que sus miembros deben llevar a cabo, así como las fuentes donde ese conocimiento puede ser obtenido, con el fin de centrar nuestro interés en aquellas que realmente puedan ser de ayuda para los usuarios del MC. En este trabajo describimos el proceso que se llevó a cabo para la creación de un MC en el ámbito del mantenimiento de software, por medio de una metodología para identificar flujos de conocimiento con base en técnicas de modelado de procesos. El objetivo es plantear nuestra experiencia en la realización de este esfuerzo, con la finalidad de que algunas de estas ideas puedan ayudar a otras organizaciones de desarrollo de software en trabajos similares.

PALABRAS CLAVES

Mapa de conocimiento
Gestión de conocimiento

Identificación de flujos de conocimiento
Ingeniería de software
Mantenimiento de software
Mejoramiento de proceso software

ABSTRACT

Some of the problems in software development organizations, particularly in the groups in charge of software maintenance, are the scarce use of the knowledge that can be obtained from the sources available in the organization. This is because those sources are often unknown or difficult to locate. Knowledge maps can help to address this problem since these maps can be used to describe the knowledge sources available in the organization, the knowledge that can be obtained from them, and how or where those sources can be accessed. To create a knowledge map, an important factor is to define what knowledge and which sources should be considered. The identification of how knowledge flow through the organization can help to accomplish this, since it can help to identify the knowledge required by the members of the organization to carry on their activities, as well as the sources in which that knowledge can be obtained, in order to focus on those sources that may be really useful to the users of the knowledge map. This paper describes the process followed to create a knowledge map for a software maintenance group following a methodology to identify knowledge flows based on process modelling techniques. The main goal of this paper is to present our experience in the development of the map, expecting that some of these ideas could be useful to other software organizations that could engage similar works in the future.

KEYWORDS

Knowledge map
Knowledge management
Knowledge flows identification
Software engineering
Software maintenance
Software process improvement¹

INTRODUCCIÓN

El mantenimiento de software es una actividad donde el conocimiento juega un papel muy importante. Los Encargados del Mantenimiento (EM) de software deben saber, por ejemplo, qué cambios hacer, dónde y cómo hacerlos, y qué efectos pueden causar en el resto del sistema. Para obtener este conocimiento, a su vez se requiere conocer la estructura del sistema, las relaciones entre los distintos módulos, las herramientas que usarán para la modificación del sistema, el lenguaje de programación, etc. Con frecuencia, los EM no cuentan con todo el conocimiento que requieren para realizar su tarea por lo que deben consultar la documentación disponible, o a compañeros de trabajo [31]. Sin embargo, en ocasiones las fuentes de información son difíciles de localizar, además de que no se conoce la información que contienen e incluso, lo que es más grave, se desconoce su existencia [28]. Como consecuencia, parte del conocimiento existente dentro de los grupos de mantenimiento se pierde o no es utilizado.

La Gestión de Conocimiento (GC) provee técnicas y métodos que pueden ayudar a reducir la pérdida y el desaprovechamiento del conocimiento, por ejemplo,

ayudando a compartirlo entre los miembros de un grupo [24]. La GC puede proveer varios beneficios para las organizaciones de desarrollo y mantenimiento de software [17], entre los que se destacan, mejoras en la calidad de los productos y procesos, reducciones de costos y defectos [10]. Sin embargo, antes de desarrollar sistemas de GC para un grupo particular, es importante identificar qué conocimiento se quiere gestionar, dónde se almacena ese conocimiento, y dónde se requiere. En general, es importante entender cómo fluye el conocimiento dentro del grupo al que se quiere apoyar con el sistema de GC, dado que el objetivo final de este tipo de sistemas es ayudar a que el conocimiento fluya hacia donde se requiere en el momento que se requiere [5].

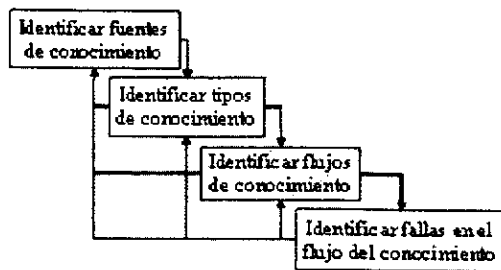
En este artículo se describe cómo se llevó a cabo la construcción de un mapa de conocimiento que ayude a disminuir la pérdida y desaprovechamiento del mismo dentro de un grupo de mantenimiento de software. Lo anterior se llevó a cabo a través del uso de la metodología KoFI [25] que facilita la identificación de flujos de conocimiento en los procesos de las organizaciones. La utilidad de este mapa ha sido probada por medio de su uso en un prototipo diseñado para apoyar escenarios específicos dentro del grupo estudiado. Las siguientes secciones de este artículo están organizadas de la siguiente manera: primero, en la siguiente sección se presenta una breve descripción de la metodología KoFI. Después, en la tercera sección se describe la forma en que fue construido el mapa de conocimiento, para después, en la cuarta sección describir su utilización dentro de un prototipo para apoyar la GC en un grupo de mantenimiento de software. Posteriormente, se introducen algunas de las principales lecciones aprendidas de la realización de este trabajo. Finalmente, las conclusiones de este trabajo se presentan en la sección seis.

2. KOFI: UNA METODOLOGÍA PARA IDENTIFICAR FLUJOS DE CONOCIMIENTO

KoFI es una metodología centrada en la identificación de flujos de conocimiento, por medio del estudio y modelado de los procesos de una organización, con enfoque en la identificación del conocimiento requerido y generado en las principales actividades del proceso, así como las fuentes donde éste es almacenado y obtenido [25].

KoFI sigue dos propósitos principales, 1) obtener información que ayude a la creación de un mapa de conocimiento que posteriormente sea utilizado para la generación de una base de conocimiento para el grupo estudiado, y 2) obtener requerimientos que apoyen el diseño de sistemas de GC para dicho grupo. Con este fin, KoFI define cuatro etapas principales (ver Figura 1): 1) identificación de fuentes de conocimiento, 2) identificación de temas de conocimiento, 3) identificación de flujos de conocimientos, e 4) identificación de problemas que afecten los flujos de conocimiento.

Figura 1. Etapas de la metodología KoFI



2.1 IDENTIFICACIÓN DE FUENTES DE CONOCIMIENTO

En esta etapa se identifican los principales tipos de fuentes de conocimiento e información que son utilizados dentro del proceso estudiado. Entre las fuentes que deben ser consideradas se encuentran:

- Documentos utilizados o generados dentro del grupo (libros, manuales, tutoriales, reportes, etc.).
- Elementos que componen los productos generados por el grupo. Por ejemplo los sistemas de software, tanto el código fuente como los sistemas ejecutables, en el caso del mantenimiento de software.
- Personas involucradas en los procesos; como los miembros del grupo o personas externas que son consultadas por estos, etc.
- Sistemas de información empleados por el grupo, como bases de datos, memorias organizacionales, bases de conocimiento, etc.
- Herramientas que pueden ser usadas por los miembros del grupo para adquirir información o conocimiento sobre aspectos específicos. Por ejemplo, ambientes de desarrollo de software, sistemas de control de versiones, analizadores de código, etc. (para el caso del mantenimiento de software).

Con el fin de facilitar la clasificación de las distintas fuentes de información, se pueden definir esquemas de clasificación basados en una taxonomía, o una ontología [29] que no solo permite definir tipos de fuentes, sino también los atributos principales que deben ser considerados, así como las relaciones de estas fuentes con otras, e incluso con otros elementos del proceso. En algunos casos pueden existir estándares y/u ontologías desarrollados para el tipo de proceso estudiado que podrían ser tomados en cuenta.

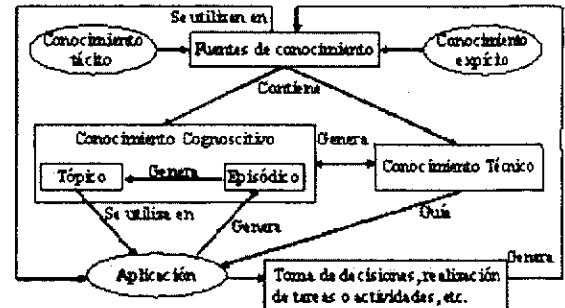
Las técnicas empleadas para la obtención de datos en esta etapa, son principalmente dos: entrevistas a los participantes del proceso, enfocadas en identificar los documentos, personas o sistemas de información que consultan los mismos; y análisis de los documentos y sistemas utilizados por los miembros del grupo estudiado.

2.2 IDENTIFICACIÓN DE TIPOS DE CONOCIMIENTO

Esta etapa consiste en la identificación de los principales temas de conocimiento requeridos y generados en el proceso

estudiado, para lo cual es necesario tener en cuenta los distintos tipos de conocimiento que existen dentro de una organización.

Figura 2. Modelo de las relaciones de los distintos tipos de conocimiento.



El conocimiento de una organización puede ser clasificado en explícito y tácito [19]. El explícito está expresado de una manera formal, como palabras, números, etc., por lo que puede ser fácilmente transferible en forma de documentos, registros en una base de datos, etc. El tácito es el conocimiento personal de cada individuo, corresponde con sus habilidades, experiencia, etc., y por su naturaleza es más difícil de comunicar y compartir. Tomando en cuenta cómo el conocimiento tácito se almacena en la mente de las personas, éste puede subdividirse en técnico o procedural, y declarativo o cognitivo [23]. El conocimiento técnico es dinámico y tiene que ver con la pericia para realizar una tarea; puede verse como la experiencia o habilidad para realizar alguna actividad. El conocimiento cognoscitivo es estático y se refiere a esquemas, modelos mentales, creencias y percepciones sobre conceptos (objetos, personas o eventos), sus propiedades y las relaciones entre ellos. Este a su vez puede dividirse en tópico o semántico, que tiene que ver con los significados y definiciones de las cosas; y episódico, que corresponde con la experiencia de las personas en la aplicación de los otros tipos de conocimiento. La Figura 2 muestra un modelo de las relaciones entre los distintos tipos de conocimiento desde el punto de vista de su aplicación en la realización de tareas, toma de decisiones, etc., y de cómo esta aplicación de conocimiento permite a su vez generar nuevo conocimiento.

Desde el punto de vista organizacional, la importancia del conocimiento radica en su aplicación [3]. Por tanto, la identificación del conocimiento debe centrarse en el conocimiento involucrado en las actividades y toma de decisiones dentro del proceso. Con base en el modelo mostrado en la Figura 2, es importante identificar qué conocimiento requieren los involucrados en el proceso para realizar sus actividades, tanto la información o datos requeridos en éstas, como las habilidades o experiencia que ayuden a realizarlas. Es importante identificar con qué conocimientos cuentan los participantes en el proceso, qué conocimiento obtienen y qué conocimiento se genera que pueda ser útil en otras actividades. Entre los aspectos a considerar se encuentran los siguientes:

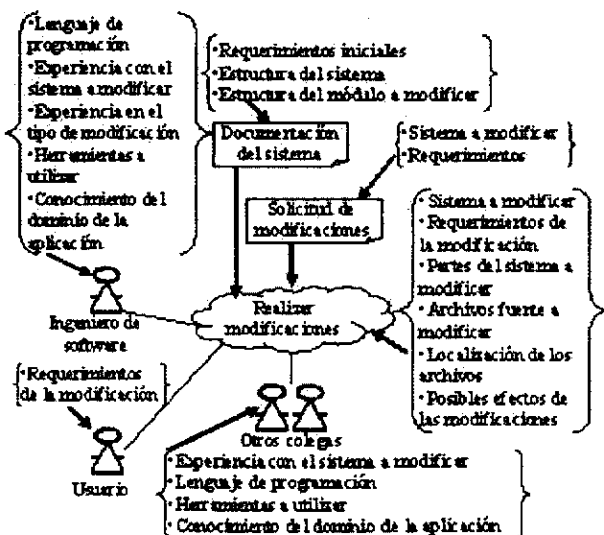
- Conocimiento acerca de la estructura de la organización, sus normas, cultura, etc.
- Conocimiento de los procesos de la organización, como actividades, tareas, personas involucradas, etc.
- Conocimiento dependiente de las actividades. Por ejemplo, los procedimientos para realizarlas, herramientas de soporte utilizadas para llevarlas a cabo, etc.
- Conocimiento del dominio de trabajo (o giro) de la organización o grupo estudiado.
- Otros tipos de conocimiento que puedan ser importantes. Por ejemplo, puede ser importante conocer qué empleados saben hablar otros idiomas, o tienen habilidades que no son parte del trabajo diario, pero que podrían ser útiles en situaciones especiales.

Las técnicas empleadas para la obtención de datos en esta etapa también se centran en entrevistas, análisis de documentos, y fuentes de información en general. Así se identifica la información y conocimiento que requieren los encargados del proceso, y el que existe almacenado en los documentos o sistemas de información usados dentro de la organización. Sin embargo, la información obtenida en esta etapa puede ser complementada por medio del modelado de flujos de conocimiento realizado en la siguiente etapa de la metodología.

2.3 IDENTIFICACIÓN DE FLUJOS DE CONOCIMIENTO

En esta etapa se hace uso de técnicas de modelado de procesos, dado que éstas pueden ayudar a identificar entradas y salidas de conocimiento en las principales actividades de un proceso, las fuentes donde ese conocimiento es obtenido o almacenado, flujos de conocimiento entre actividades, roles o fuentes de información; así como los problemas que pueden afectar el flujo de conocimiento [1, 8].

Figura 3. Ejemplo del uso de una gráfica para identificar el conocimiento y fuentes involucradas en un proceso.



Existe distintas técnicas y lenguajes para modelar procesos que pueden ser empleadas [7]. La Figura 3 muestra un ejemplo del uso de gráficas ricas para este fin, donde se identifican los principales conocimientos requeridos para realizar una actividad, el conocimiento que aporta el rol encargado de llevarlo a cabo, así como las fuentes que pueden ser consultadas, junto con el conocimiento que puede ser obtenido de éstas. La gráfica rica es una técnica por lo general utilizada en las etapas iniciales de un estudio de procesos, y que permiten obtener una visión global del mismo a distintos niveles de detalle [18], por lo que es posible usarlas para identificar los elementos principales involucrados en el proceso; por ejemplo el flujo del conocimiento entre actividades, roles o sistemas de información [11].

2.4 IDENTIFICACIÓN DE PROBLEMAS EN LOS FLUJOS DE CONOCIMIENTO

El objetivo de la identificación y modelado de tipos y fuentes de conocimiento, así como de flujos de conocimiento dentro del proceso, no es sólo entender cómo fluye el conocimiento dentro del mismo, sino también identificar los problemas que puedan estar evitando que el conocimiento fluya adecuadamente. Por lo tanto, durante la realización de las tres primeras etapas de la metodología, es importante identificar este tipo de problema, por ejemplo, tipos de fuentes de conocimiento que no son consultadas, ya sea porque no son conocidas por los miembros del grupo, o porque son difíciles de acceder, o no se sabe el conocimiento o información que contienen.

Sin embargo, en esta cuarta y última etapa de la metodología se recopilan los problemas que se hayan encontrado y se clasifican con el fin de comenzar a buscar posibles soluciones. Esto se realiza mediante la definición de escenarios de problemas, los cuales son descripciones textuales de ejemplos que muestren cómo se da el problema dentro del proceso. Una vez realizado esto, el siguiente paso es definir escenarios alternativos que ejemplifiquen cómo se podría resolver ese problema, de forma que estos escenarios alternativos sirvan para definir las características que debe cubrir un sistema que dé soporte al tipo de problemas encontrados.

El uso de escenarios es una técnica que puede ser empleada para identificar especificaciones de diseño de sistemas de software [6], al definir la manera en que se espera que el sistema se comporte bajo situaciones específicas. Esto permite obtener una idea de los resultados que pueden ser obtenidos al desarrollar un sistema que cumpla con las características descritas en el escenario.

3. CREACIÓN DEL MAPA DE CONOCIMIENTO

Un mapa de conocimiento puede ser descrito de diferentes maneras dependiendo de la perspectiva desde la cual sea desarrollado [16]. En nuestro caso hemos adoptado la definición de Davenport y Prusak [8], quienes definen un mapa de conocimiento como guías que ayudan a trazar o encontrar el conocimiento de una organización, por ejemplo, por medio de apuntadores a las fuentes de ese conocimiento. Desde esta perspectiva, un mapa de conocimiento puede ser

desarrollado identificando las principales fuentes de información y conocimiento existentes dentro de una organización, así como la información y conocimiento que pueden ser obtenidos de ellas.

En esta sección describimos la creación de un mapa de conocimiento para un grupo de mantenimiento de software. La información para la generación de este mapa fue obtenida por medio de la realización de entrevistas a los miembros del grupo, análisis de documentos y sistemas de información, así como el modelado de procesos y flujos de conocimiento dentro del grupo, siguiendo los pasos de la metodología KoFI [25]. Una vez que fueron identificados y clasificados los principales tipos de conocimiento y sus fuentes, se desarrolló un metamodelo que nos ayudó a definir las relaciones tanto entre los tipos de conocimiento y sus fuentes, como de éstos con las actividades donde son requeridos o generados. Finalmente este metamodelo fue la base de la estructura del mapa.

A la par, se hizo también una revisión de la literatura referente al conocimiento requerido por los encargados del desarrollo de software en lo general, y mantenimiento en lo particular, así como de las fuentes que utilizan para obtener este conocimiento. Entre la literatura revisada se encuentran estudios sobre el uso de documentación en el mantenimiento de software (ej. [15, 28]), ontologías para el mantenimiento de software (ej. [9, 13, 20, 26]), estándares y metodologías de mantenimiento de software (ej. [12, 21]), y el SEWBOK [2].

3.1 DESCRIPCIÓN DEL GRUPO ESTUDIADO

Antes de describir la forma en que se clasificaron las fuentes de conocimiento y tipos de conocimiento para la generación del mapa, daremos algunos detalles del grupo estudiado. Este grupo corresponde al departamento de informática (DI) de un centro de investigación en el norte de México. Al momento del estudio, el DI estaba constituido por 14 personas, un jefe del departamento, una secretaria, 6 ingenieros de software, y seis programadores de apoyo. Este departamento se encarga del mantenimiento de 5 tipos de sistemas de información empleados dentro del centro: estos incluyen sistemas para la administración de recursos humanos, financieros, y materiales; seguimiento de la productividad del personal académico y de investigación, y servicios a estudiantes.

3.2 FUENTES DE CONOCIMIENTO

Las fuentes de información y conocimiento consultadas por los encargados del mantenimiento del software pueden ser muy variadas, como se observa en diversos estudios (ej. [14, 28]). En ocasiones dependen de las políticas de la empresa, de su infraestructura, e incluso de si usan alguna metodología, estándar o modelo de proceso específico (véase por ejemplo [12, 21]). Sin embargo, es posible extraer algunas generalizaciones para ayudar a clasificar los distintos tipos de fuentes de conocimiento. Por ejemplo, considerando si son personas, documentos, etc.; o la información que puede ser obtenida de ellas. A continuación presentamos un esquema de clasificación de tipos de fuentes de conocimiento realizado para el grupo estudiado, y la manera en que estas fuentes

fueron descritas para posteriormente integrarse como parte del mapa de conocimiento.

3.2.1 Clasificación de fuentes de conocimiento

Con base a nuestras observaciones en el grupo estudiado, así como en la literatura revisada, se clasificaron las fuentes de información y conocimiento consultadas por el grupo en cuatro categorías, (I) documentos, (II) personas, (III) elementos de los sistemas a ser mantenidos, y (IV) herramientas de soporte y sistemas de información; descritos a continuación:

La categoría de documentos, agrupa toda clase de documentos, tanto en papel como electrónicos. Esta categoría ha sido dividida en seis tipos:

1. Documentación del sistema agrupa todos los documentos con la información de las aplicaciones que son mantenidas por el grupo, como documentos de requerimientos, diseño del sistema, etc.
2. Documentación técnica es usada para agrupar documentos que aportan información y conocimiento técnico necesario para las tareas del mantenimiento, como libros sobre lenguajes de programación, manuales sobre las herramientas usadas durante el mantenimiento, ambientes de desarrollo; etc.
3. Documentación para usuarios es utilizada para agrupar los documentos dirigidos a los usuarios de las aplicaciones mantenidas por el grupo, por ejemplo los manuales de usuario, de instalación, de configuración, etc.
4. Documentación de la organización es usada para clasificar los documentos relacionados con la vida interna de la organización, tales como los que aporten información sobre la estructura de la organización, sus procesos, políticas, normas, etc.
5. Documentación del proceso de mantenimiento es usada para agrupar todos los documentos directamente relacionados con el proceso de mantenimiento de software, entre ellos podemos citar las solicitudes de cambios, reportes de errores, planes y proyectos de mantenimiento, historial de modificaciones, planes de prueba, de administración de la configuración, etc.
6. Otros documentos es usada para clasificar todos aquellos documentos que no son considerados en las otras cinco sub-categorías.

La categoría personas agrupa todos los tipos de personas que son consultadas por los miembros del grupo estudiado. Esta a su vez se ha subdividido en tres sub-categorías o tipos:

1. Usuarios y clientes. Aun cuando los usuarios y clientes juegan roles distintos en el proceso de mantenimiento [22], en el grupo estudiado no existe una distinción al respecto, por lo que los hemos considerado como uno solo. En este caso, tanto los usuarios como el grupo de mantenimiento forma parte de la misma organización.
2. Miembros del grupo. Aquí se consideran a las personas que colaboran dentro del grupo de mantenimiento.
3. Otros expertos. Esta sub-categoría corresponde a todas las personas que no son usuarios o miembros del grupo, pero que

son consultados por estos últimos para obtener conocimientos especializados, por ejemplo sobre el dominio de la aplicación a ser modificada, sobre el lenguaje de programación a emplear, etc. Estos expertos pueden ser tanto internos como externos a la organización. Por ejemplo, algunos de los miembros del grupo reportaron consultar expertos por medio de grupos de interés en Internet o listas de correo electrónico.

La categoría elementos del sistema agrupa los elementos que constituyen los productos a ser mantenidos, sin incluir la documentación de los mismos. Esta categoría fue subdividida en tres sub-categorías:

1. Sistema ejecutable. El sistema ejecutable con frecuencia es usado por los mantenedores para analizar el comportamiento del mismo. Por ejemplo, los miembros del grupo reportaron emplear el sistema ejecutable para trazar problemas siguiendo los pasos realizados por los usuarios, con el fin de identificar bajo qué condiciones se da el problema, en qué módulo específico se da, etc.
2. Código fuente. Esta es posiblemente la principal fuente de información usada por los mantenedores de software [28], ya que representa el estado real del sistema, por lo que es la fuente de información más actualizada si se quiere conocer, por ejemplo, la estructura interna del sistema.
3. Bases de datos del sistema. Las bases de datos son un componente esencial dentro de los sistemas mantenidos por el grupo estudiado. Los miembros del grupo reportaron que en ocasiones consultan las bases de datos para analizar las relaciones entre las distintas entidades de datos (tablas), así como los flujos de datos entre la base de datos y el sistema y viceversa, para evaluar si estos flujos se realizan adecuadamente.

Finalmente, la categoría de herramientas de soporte agrupa las herramientas de software y sistemas de información usados por los miembros del grupo. Esta categoría ha sido subdividida en dos:

1. Herramientas de soporte al proceso de mantenimiento que agrupa todas las aplicaciones de software empleadas directamente para las labores del mantenimiento, como analizadores de código, sistemas de control de versiones de código, ambientes de desarrollo, y en general todo tipo de herramientas CASE.
2. Herramientas de soporte general se refiere a las aplicaciones de software usadas dentro del grupo, pero que no están directamente relacionadas con las actividades de mantenimiento. Entre estas se encuentran los portales web, memorias organizacionales, repositorios de documentos, motores de búsqueda de información, etc.

3.2.2 Descripción de las fuentes de conocimiento

Para definir cada fuente específica una vez definido el esquema de clasificación de las mismas, se usaron esquemas como el mostrado en la Tabla I. Estos esquemas nos permitieron identificar cada fuente, clasificarla, definir el medio por el cual puede ser localizada, así como empezar a definir la información y el conocimiento principal que puede ser obtenido de ellas. Estas descripciones fueron

posteriormente representadas en formato XML siguiendo el metamodelo y la Definición de Documento presentadas en las secciones 3.4 y 3.5.

3.3 TIPOS DE CONOCIMIENTO

El conocimiento puede ser clasificado en distintos tipos, por ejemplo en tácito y explícito. Como vimos anteriormente, tanto el conocimiento tácito como el conocimiento explícito pueden también ser subdivididos o clasificados, por ejemplo dependiendo del nivel de formalización del mismo [30]. En nuestro caso hemos decidido seguir una estructura de clasificación basada en temas agrupados en áreas de conocimiento, agrupadas a su vez en categorías, siguiendo un esquema similar al empleado para clasificar las áreas de conocimiento en el SEWBOK [2]. En nuestro esquema, las categorías de conocimiento denotan elementos estructurales de alto nivel de abstracción, usadas para organizar y clasificar áreas de conocimiento, las cuales son subdivisiones de las categorías que están lógicamente relacionadas con éstas, ya sea por herencia, agregación o composición, es decir, que sean una subdivisión de la categoría o parte de los componentes de la categoría. Los temas de conocimiento son subdivisiones de las áreas que representan conceptos básicos que tienen una descripción explícita y bien definida. Describen el conocimiento sobre un conjunto de elementos que pueden ser definidos como una unidad. Los temas de conocimiento pueden ser unidades básicas de conocimiento, es decir que no tengan más subdivisiones, o pueden estar constituidos por temas más especializados.

Tabla 1. Descripción de una fuente de Conocimiento

Nombre:	p1230_requerimientos.doc		
Categoría:	Documentación		
Tipo:	Documentación del Sistema / Requisitos		
Descripción:	Documento que contiene los requisitos del sistema de recursos financieros		
Localización			
Tipo	Descripción	Soporte físico	Formato
electrónico	Directorio: "c:\project\p1230\documentacion!" en el servidor de archivos de proyectos		word 2000
Conocimientos principales			
Concepto		Nivel	
Requisitos del sistema SIREFI		Avanzado	

3.3.1 Clasificación de temas de conocimiento

Dado que la importancia del conocimiento de una organización radica en dónde es aplicado [3]. La definición de los temas de conocimiento se basó en la identificación del conocimiento requerido por los miembros del grupo estudiado para realizar sus actividades. Rus et al. [27] proponen que dependiendo de

las actividades dentro de una organización de desarrollo de software, el conocimiento puede ser clasificado en: organizacional, referente a cómo manejar la compañía, cuáles son los objetivos del negocio, recursos humanos disponibles, etc.; de gestión, referente a la planeación seguimiento y gestión de proyectos; técnico, referente al conocimiento y habilidades para realizar las actividades del desarrollo de software, como análisis de requisitos, diseño, programación, pruebas, etc.; y del dominio, referente al conocimiento del dominio del tipo de aplicaciones o sistemas desarrollados o mantenidos por la organización. Siguiendo esta idea hemos definido tres categorías principales para clasificar las áreas de conocimiento dentro del grupo estudiado: (I) la primera para las actividades directamente relacionadas con el proceso de mantenimiento, (II) la segunda para las actividades relacionadas con la vida dentro de la organización a la que pertenece el grupo, y (III) la tercera para agrupar conocimientos y habilidades generales que no son parte del trabajo diario pero que es importante tener en cuenta. A continuación se describen cada una de estas categorías, así como las principales áreas de conocimiento dentro de cada una de ellas.

La categoría del conocimiento de las actividades de mantenimiento agrupa la mayoría de las áreas de conocimiento que han sido definidas, las cuales han sido agrupadas en tres sub-categorías:

1. Fundamentos de computación e informática, que agrupa áreas de conocimiento relacionadas con temas como algoritmos, estructuras de datos, sistemas operativos, lenguajes de programación, técnicas y paradigmas de programación, etc.
2. Ingeniería de software es una categoría que ha sido extraída de la anterior dada su importancia para el proceso de mantenimiento. En esta categoría se agrupan áreas de conocimiento relacionadas con las distintas fases del ciclo de vida del software, y el proceso de desarrollo del mismo. Por ejemplo, administración de proyectos de software, ingeniería de requisitos, análisis y diseño de sistemas, técnicas de mantenimiento, pruebas, metodologías y lenguajes de modelado de sistemas, manejo de herramientas CASE, etc.
3. Conocimiento de la aplicación es una categoría usada para agrupar el conocimiento relacionado con las aplicaciones mantenidas por el grupo. Esta categoría a su vez se subdivide en sub-categorías, una para cada producto o sistema. Así también, para cada producto se cuenta con dos sub-categorías más, una para el conocimiento del producto, que incluye temas de conocimiento relacionados con la estructura o arquitectura del producto, su funcionalidad, historial, y una referencia a su documentación. La otra sub-categoría se refiere al conocimiento del dominio de la aplicación, y agrupa los temas de conocimiento referentes a las actividades que son apoyadas por la aplicación.

La categoría sobre el conocimiento de la vida en la organización considera los temas o áreas de conocimiento que no están directamente relacionados con el proceso de mantenimiento, pero que todos los miembros del grupo deben conocer. Estas áreas incluyen el conocimiento sobre la estructura de la organización, sus normas, políticas, metas, etc.; conocimiento sobre otros procesos de la organización,

como sus actividades, roles involucrados, o recursos requeridos; conocimientos sobre los clientes de la organización; etc.

Finalmente, en la categoría de conocimientos generales se agrupan áreas o temas de conocimiento, y habilidades que no son parte del trabajo diario pero que puede ser importante tener en cuenta, como por ejemplo, habilidades de escritura, liderazgo, coordinación de grupos, para hablar otros idiomas, etc.

3.3.2 Descripción de temas de conocimiento

La descripción de temas de conocimiento específicos se hizo por medio de la definición de esquemas como el mostrado en la Tabla II. Se identificaron los temas principales, y se clasificaron. Posteriormente se definieron temas relacionados, por medio de la identificación del conocimiento cognoscitivo (tanto tópico como episódico) y técnico relacionado. La identificación de estos tipos de conocimiento se basó en las actividades que el grupo debe realizar, y la definición de los temas relacionados se refiere a lo que se requiere para poder llevar a cabo esa actividad. Siguiendo el ejemplo de la Tabla II, los temas de conocimiento tópico relacionados nos indican la información que se requiere para poder modificar el módulo de elaboración de cheques del sistema de finanzas. Los temas definidos como conocimiento episódico indican las situaciones que pueden generar que una persona obtenga conocimientos relacionados con el tema. Mientras tanto, el conocimiento técnico, indica aspectos sobre el cómo llevar a cabo ciertas actividades relacionadas.

Las descripciones de temas de conocimiento como la de la Tabla II, pueden después ser usadas para identificar qué fuentes de información o conocimiento pueden ser útiles para una determinada actividad; por ejemplo, al identificar fuentes que puedan proveer información sobre los elementos definidos como conocimiento tópico, o al identificar personas que hayan estado involucradas en situaciones como las definidas como conocimiento episódico. Los temas más relevantes definidos fueron posteriormente representados en formato XML siguiendo el metamodelo y la Definición de Documentos descritos a continuación.

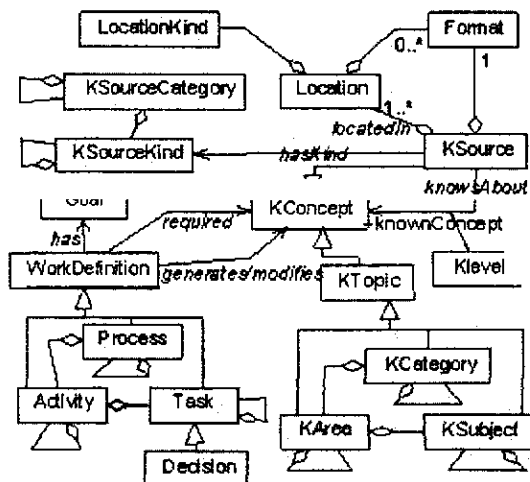
Tabla 2. Descripción de un tema de conocimiento

Tema de conocimiento	Modificar el módulo de elaboración de cheques en el sistema de finanzas
Categoría	Actividades de mantenimiento / Conocimiento de la aplicación
Área	Sistema de recursos financieros / Conocimiento del producto
Descripción	Conocimientos sobre el proceso de elaboración de cheques en el sistema de finanzas, y sobre cómo modificarlo
Conocimiento tópico	Cuál es el módulo de elaboración de cheques
	Cuáles son los archivos fuente del módulo
	Dónde se localizan los archivos
	Qué otros módulos están relacionados
Conocimiento episódico	Qué lenguaje de programación se usará
	Experiencia en el uso del módulo
	Experiencia en el desarrollo del módulo
Conocimiento técnico	Experiencia en la modificación del módulo
	Cómo acceder al módulo
	Cómo modificar el módulo
	Cómo identificar problemas en el módulo
	Cómo corregir problemas en el módulo

3.4 METAMODELO DE TIPOS Y FUENTES DE CONOCIMIENTO

Con base en los esquemas de clasificación de tipos y fuentes de conocimiento, se definió un metamodelo con el fin de definir las relaciones entre las fuentes y sus principales atributos, con el conocimiento e información que puede ser obtenido de ellas, y con las actividades donde, tanto el conocimiento como sus fuentes, pueden ser requeridos, o generados. Como se puede observar en la Figura 4, en el metamodelo, los tipos de conocimiento (KTopic) y sus fuentes (KSource) han sido considerados como conceptos de conocimiento (KConcept) que son requeridos, generados o modificados en las operaciones llevadas a cabo por el grupo, llamadas definiciones de trabajo (WorkDefinition [32]), las cuales pueden ser procesos, actividades o tareas. A su vez, cada fuente puede tener información o conocimiento sobre distintas áreas o temas específicos, así como sobre otras fuentes. El grado de detalle, especialización, experiencia, etc. sobre la información o conocimiento que puede ser obtenido de la fuente, puede determinarse por medio del uso de elementos KLevel. Finalmente, cada fuente puede tener una o más formas de localizarla (Location), así como un formato, el cual puede depender del tipo de localización, por ejemplo, pdf para un documento electrónico localizado en un repositorio de documentos, o html para uno localizado en una página web, etc. Los distintos tipos de localización son definidos por medio de elementos LocationKind.

Figura 4. Metamodelo de tipos y fuentes de conocimiento.



3.5 REPRESENTACIÓN DEL MAPA DE CONOCIMIENTO EN XML

El metamodelo definido fue la base de la estructura del mapa de conocimiento. El mapa fue creado por medio de la representación en formato XML de los distintos elementos del metamodelo. Las Figuras 5 y 6 muestran la descripción de documento de las fuentes y tipos de conocimiento respectivamente.

La descripción de las fuentes de conocimiento permite definir los mecanismos por medio de los cuales pueden ser localizadas, así como los temas referentes al conocimiento que puede ser obtenido de éstas. La descripción de los tipos de conocimiento permite especificar si éstos están relacionados con otros ya sea porque se refieren a conocimiento similar o equivalente, o porque para su aplicación se requiere del conocimiento de otros temas. Dado que existe la posibilidad de que algún tema de conocimiento pueda pertenecer a más de un área de conocimiento, se añadió la posibilidad de definir más de una clasificación. Finalmente, las descripciones de fuentes y temas de conocimiento fueron almacenados en una base de datos XML.

Figura 5. DTD para la representación de fuentes de conocimiento

```

<!DOCTYPE KSource [
<ELEMENT KSource (name, category, type, description,
located_at, knows_about) >
<!ATTLIST KSources id ID #REQUIRED>
<ELEMENT name (#PCDATA) >
<ELEMENT category (#PCDATA) >
<ELEMENT type (#PCDATA) >
<ELEMENT description (#PCDATA) >
<ELEMENT located_at (location)+ >
<ELEMENT location (location_type, description,
physical_support?, format) >
<ELEMENT location_type (#PCDATA) >
<ELEMENT description (#PCDATA) >
<ELEMENT physical_support (#PCDATA) >
<ELEMENT format (#PCDATA) >
<ELEMENT knows_about (known_concept)+ >
<ELEMENT known_concept (KConcept, KLevel) >
<ELEMENT KConcept EMPTY >
<!ATTLIST KConcept type CDATA #REQUIRED>
<!ATTLIST KConcept id CDATA #REQUIRED>
<ELEMENT KLevel (#PCDATA) >
]>
  
```

Figura 6. DTD para la representación de tipos de conocimiento

```

<!DOCTYPE KTopic [
<ELEMENT KTopic (name, description, classification,
related_topics) >
<!ATTLIST KTopic id ID #REQUIRED>
<!ATTLIST KTopic type (category | area | subject)
#REQUIRED>
<ELEMENT name (#PCDATA) >
<ELEMENT description (#PCDATA) >
<ELEMENT classification (primary, alternative*) >
<ELEMENT primary EMPTY >
<!ATTLIST primary topic_id CDATA #REQUIRED>
<ELEMENT alternative EMPTY >
<!ATTLIST alternative topic_id CDATA #REQUIRED>
<ELEMENT related_topics (related_topic)* >
<ELEMENT related_topic EMPTY >
<!ATTLIST related_topic topic_id CDATA
#REQUIRED>
<!ATTLIST related_topic relationship (similar |
requires) #REQUIRED>
]>
  
```

4. USO DEL MAPA DE CONOCIMIENTO

Para probar si el mapa puede ser usado para apoyar a los miembros del grupo en la búsqueda de fuentes de conocimiento que puedan ser útiles para sus actividades, se diseñó un sistema prototipo que hiciera uso de él [24]. Éste ayuda en la búsqueda de fuentes que puedan tener información o conocimiento relevante para las actividades de los Ingenieros de Mantenimiento (IM). El sistema muestra al IM una lista de los proyectos que debe llevar a cabo. Estos proyectos se refieren a solicitudes de cambios o correcciones de errores en el sistema que el IM debe resolver. Cuando el IM elige trabajar en uno de estos proyectos, el sistema consulta el mapa de conocimiento para obtener los temas requeridos para la actividad. Por ejemplo, si el IM debe solucionar un error en una aplicación, el sistema identifica la aplicación donde se dio el error, el módulo específico, el cliente que reportó el error, tipo de error, etc. Esta información se obtiene del reporte de error. Posteriormente, la misma se usa para definir nuevos temas de búsqueda, por ejemplo, el lenguaje de programación que se usará, los archivos fuentes que componen el módulo donde se dio el error, el dominio de aplicación de ese módulo, etc. Estos temas son identificados buscando el conocimiento requerido por la actividad que se va a realizar, así como las relaciones entre los distintos temas que se van identificando. El sistema también puede hacer uso del perfil de conocimiento del IM para descartar o dar más importancia a ciertos temas. Por ejemplo, si el IM cuenta con experiencia en el uso del lenguaje de programación en el que fue desarrollado el módulo, el sistema puede descartar este tema. Así también, si el sistema detecta que el IM no tiene experiencia en el dominio de aplicación del módulo, puede centrarse en buscar fuentes que tengan este conocimiento.

Una vez que se ha terminado de definir la lista de temas a buscar, el sistema inicia la búsqueda haciendo consultas al mapa de conocimiento para identificar las fuentes relacionadas con los temas definidos. Al finalizar la búsqueda, el sistema envía un mensaje al IM informándole el total de fuentes encontradas y que pudieran ser de ayuda. Si el IM decide consultar alguna de estas fuentes, el sistema muestra una ventana como la de la Figura 7, donde se listan las fuentes que fueron encontradas, ordenadas según el tipo de fuente, así como datos sobre cómo localizar esa fuente, y el

conocimiento que ésta tiene y que está relacionado con la actividad que el IM lleva a cabo.

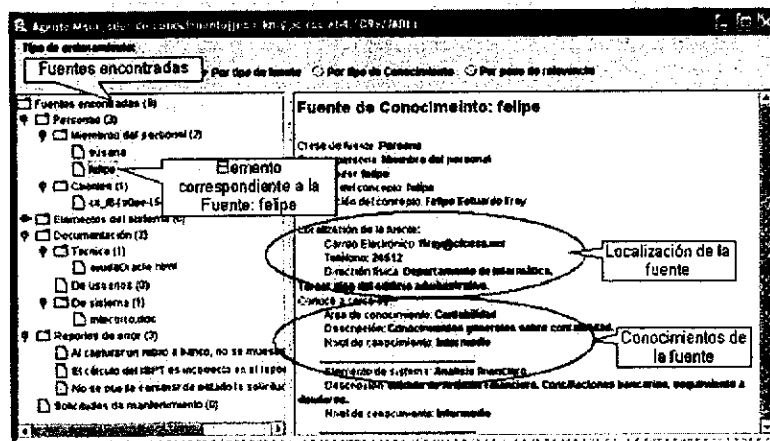
5. LECCIONES APRENDIDAS

En esta sección describimos las principales lecciones aprendidas durante la realización de este trabajo, las cuales se derivan principalmente de las dificultades que se presentaron durante la creación del mapa de conocimiento, y de la búsqueda de soluciones a las mismas.

Foco en el flujo del conocimiento. Al inicio del estudio, uno de los problemas principales fue definir qué temas de conocimiento debían ser tomados en cuenta dentro del mapa. Durante las entrevistas realizadas a los miembros del grupo, así como por el análisis de documentos, nos percatamos de que clasificar todos los temas que podían surgir, podría llevarnos varias semanas; además, algunos de ellos podrían no ser relevantes, por lo que dedicar tiempo a clasificarlos y describirlos podría implicar una pérdida de tiempo. Sin embargo, una vez iniciado el modelado de los flujos de conocimiento dentro del grupo por medio de la identificación de las principales actividades que debían ser llevadas a cabo, el conocimiento requerido para realizarlas, y las fuentes que los IM consultaban para obtenerlo, fue cuando empezamos a clarificar qué temas eran realmente útiles. Esto nos hizo ver la importancia de centrarnos primero en los flujos de conocimiento que se dan dentro del grupo, con el fin de identificar el conocimiento que actualmente requieren para sus actividades, y de esa manera, buscar las fuentes que puedan ser útiles para obtener dicho conocimiento, para posteriormente buscar la manera de facilitar la identificación y el acceso a esas fuentes.

Definición y clasificación de temas de conocimiento. Dado que distintas personas pueden definir un mismo ambiente de muy diversas formas, resultó complicado obtener una estandarización sobre cómo definir cada tema a ser tenido en cuenta, así como su clasificación. En la búsqueda de una solución a este problema, analizamos distintas ontologías que han tratado de agrupar los principales conceptos involucrados en un proceso de mantenimiento de software [9, 13, 20, 26], así como el estándar de mantenimiento de software de la IEEE [12]. Sin embargo, esto no resolvió el problema, ya que sólo

Figura 7. Ejemplo del uso del mapa de conocimiento para buscar y clasificar fuentes de conocimiento.



se incrementaron las posibilidades de como nombrar y clasificar cada concepto. Por ejemplo, la ontología de Kitchenham et al. [13] fue tomada como base para las de Dias et al. [9], y Ruiz et al. [26]; sin embargo, estas dos ontologías, además de considerar elementos distintos, tienen formas distintas de nombrar los mismos conceptos, así como de agrupar las relaciones entre los distintos elementos del proceso de mantenimiento. Al final, se dejó la posibilidad de especificar cuáles temas pueden ser similares, al incluir la definición de relaciones entre temas de conocimiento. Así también, se dejó la posibilidad de agrupar temas dentro de varias categorías, como se mostró en la Definición de Tipo de Documento (DTD) para la representación de temas de conocimiento en XML de la Figura 6.

Granularidad en la clasificación de temas. La clasificación de los temas de conocimiento se realizó por medio de un esquema basado en categorías, áreas y temas, siguiendo un enfoque similar al utilizado en el SEWBOK [2]. Definir las categorías principales no fue un problema, puesto que seguimos un enfoque basado en las actividades y procesos principales dentro del grupo, similar al mencionado por Rus et al. [27]. Sin embargo, definir las áreas y los temas de conocimiento específicos fue confuso en varios casos. Por ejemplo, si tenemos un área que corresponde con lenguajes de programación, y uno de los lenguajes usado es C++. Podría no resultar del todo claro definir si el lenguaje C++ debe ser considerado como un tema específico o como un área de conocimiento, ya que en algunos casos podría ser suficiente con buscar fuentes con información general sobre el lenguaje, pero en otros casos podría ser necesario un nivel de mayor detalle, por ejemplo, buscar fuentes con información sobre cómo acceder a bases de datos por medio de alguna versión específica de C++ (ANSI C++, Visual C++, etc.). Por nuestra experiencia creemos que tratar de definir un esquema de clasificación de temas de conocimiento que sea genérico sería un trabajo sumamente complicado, si no imposible, por lo que consideramos que este tipo de decisiones dependerá de las necesidades particulares de cada organización. Por tanto, el esquema de clasificación de temas de conocimiento descrito en este trabajo, sólo es un ejemplo de un caso particular que puede servir como guía para la realización de otros similares.

Un trabajo complicado. El proceso llevado a cabo en la definición del mapa de conocimiento, resultó un trabajo complicado desde el punto de vista del tiempo requerido para el análisis y la captura de la información, y el esfuerzo intelectual que requieren muchas de las decisiones que se deben tomar. Por lo tanto, consideramos que la realización de este tipo de esfuerzos para apoyar a las organizaciones de desarrollo de software a hacer un mejor manejo del conocimiento con el que cuentan, debe ser dejado a personal externo a las mismas, como lo han propuesto otros enfoques para apoyar el manejo del conocimiento en organizaciones de software (por ejemplo, la fábrica de experiencias [4]). Sin embargo, una vez creada la estructura y capturada la información inicial del mapa de conocimiento, una empresa pequeña o mediana con recursos limitados como la mayoría de las empresas de software en Latinoamérica, difícilmente podrá mantener a un grupo especial encargado del

mantenimiento del mapa, por lo que consideramos que es necesario proveer medios que faciliten este mantenimiento por parte de los mismos miembros de la organización. Por ejemplo, proporcionando medios para la creación de nuevos temas o áreas de conocimiento, la inclusión de nuevas fuentes de conocimiento, etc. El desarrollo de este tipo de soporte constituye parte de nuestro trabajo futuro.

6. CONCLUSIONES

En el presente artículo se describió el proceso realizado para construir un mapa de conocimiento para un grupo de mantenimiento de software, siguiendo una metodología para el estudio de flujos de conocimiento dentro de los procesos de una organización. La metodología se aplicó en un caso de estudio en un grupo encargado de dar mantenimiento a los sistemas de un centro de investigación en México. La experiencia en la realización de este caso de estudio, y en particular las principales lecciones aprendidas fueron descritas en este artículo. Estas lecciones constatan la importancia de estudiar los flujos de conocimiento para poder proveer mecanismos que faciliten la localización de fuentes de información y conocimiento dentro de los grupos de mantenimiento de software. Dada la dificultad observada durante el trabajo realizado para el desarrollo del mapa, consideramos que también es importante proveer medios que faciliten este trabajo. Este artículo aporta algunas ideas que en nuestro caso fueron útiles para la definición del mapa y que podrían ser aprovechadas en trabajos similares en el futuro.

7. AGRADECIMIENTOS

Agradecemos el apoyo brindado por CONACYT de México, a través del proyecto C01-40799 y la beca No. 164739 asignada al primer autor; por el Ministerio de Ciencia y Tecnología de España por medio del proyecto MAS, con número de concesión TIC2003-02737-C02-02; y por la Consejería de Educación y Ciencia de la Junta de Comunidades de Castilla-La Mancha de España, a través del proyecto ENIGMAS con número de concesión PBI-05-058.

8. REFERENCIAS

- [1] Abdullah, M. S., Benest, I., Evans, A., Kimble, C., "Knowledge Modelling Techniques for Developing Knowledge Management Systems", Proc. of the European Conference on Knowledge Management, Dublin, Ireland, 2002, pp. 15-25.
- [2] Abran, A., Moore, J. W., Bourque, P., Dupuis, R., Tripp, L. L., "SEWBOK: Guide to the Software Engineering Body of Knowledge: 2004 Version". IEEE Computer Society, Los Alamitos, California, 2004.
- [3] Alavi, M., Leidner, D. E., "Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues", MIS Quarterly, vol. 25, 2001, pp. 107-136.
- [4] Basili, V. R., Seaman, C. B., "The Experience Factory Organization", IEEE Software, vol. 19, 2002, pp. 30-31.
- [5] Borghoff, U. M., Pareschi, R., "Information Technology for Knowledge Management", Journal of Universal Computer Science, vol. 3, 1997, pp. 835-842.
- [6] Carroll, J. M., Rosson, M. B., "Getting Around the

- Task-Artifact Cycle: How to Make Claims and Design by Scenario", *ACM Transactions on Information Systems*, vol. 10, 1992, pp. 181-212.
- [7] Curtis, B., Kellner, M. I., Over, J., "Process Modeling", *Communications of the ACM*, vol. 35, 1992, pp. 75-90.
- [8] Davenport, T. H., Prusak, L., *Working Knowledge: How Organizations Manage What they Know*. Harvard Business School Press, Boston, Massachusetts, USA, 2000.
- [9] Dias, M. G. B., Anquetil, N., Oliveira, K. M. d., "Organizing the Knowledge Used in Software Maintenance", *Journal of Universal Computer Science*, vol. 9, 2003, pp. 641-658.
- [10] Dingsøyr, T., Conradi, R., "A survey of case studies of the use of knowledge management in software engineering", *International Journal of Software Engineering and Knowledge Engineering*, vol. 12, 2002, pp. 391-414.
- [11] Hansen, B. H., Kautz, K., "Knowledge Mapping: A Technique for Identifying Knowledge Flows in Software Organizations", *Proc. of the European Conference on Software Process Improvement (EuroSPI 2004)*, Trondheim, Norway, 2004, pp. 126-137.
- [12] ISO/IEC, "ISO/IEC FDIS 14764:1999, Software Engineering - Software Maintenance", Secretariat: Standard Council of Canada, Standard 1999.
- [13] Kitchenham, B. A., Travassos, G. H., Mayrhauser, A., Niessink, F., Schneidewind, N. F., Singer, J., Takada, S., Vehvilainen, R., Yang, H., "Towards an Ontology of Software Maintenance", *Journal of Software Maintenance: Research and Practice*, vol. 11, 1999, pp. 365-389.
- [14] Koskinen, J., Salminen, A., Paakki, J., "Hypertext support for the information needs of software maintainers", *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 16, 2004, pp. 187-215.
- [15] Lethbridge, T. C., Singer, J., Forward, A., "How Software Engineers Use Documentation: The State of the Practice", *IEEE Software*, vol. 20, 2003, pp. 35-39.
- [16] Lin, F.-r., Hsueh, C.-m., "Knowledge map creation and maintenance for virtual communities of practice", *Information Processing & Management*, vol. 42, 2006, pp. 551-568.
- [17] Lindvall, M., Rus, I., "Knowledge Management for Software Organizations", In *Managing Software Engineering Knowledge*, Aurum, A., Jeffery, R., Wohlin, C., and Handzic, M., Eds. Springer, Berlin, 2003, pp. 73-94.
- [18] Monk, A., Howard, S., "The Rich Picture: A Tool for Reasoning about Work Context", *Interactions*, vol. 5, 1998, pp. 21-30.
- [19] Nonaka, I., Takeuchi, H., *The Knowledge-Creation Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, 1995.
- [20] Oliveira, K. M., Anquetil, N., M.G. D., Ramal, M., Meneses, R., "Knowledge for Software Maintenance." *Proc. of the Fifteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'03)*, San Francisco, CA, 2003, pp. 61-68.
- [21] Polo, M., Piattini, M., Ruiz, F., Calero, C., "MANTEMA: A Complete Rigorous Methodology for Supporting Maintenance based on the ISO/IEC 12207 Standard", *Proc. of the Third Euromicro Conference on Software Maintenance and Reengineering (CSMR'99)*, Amsterdam (The Netherlands), 1999, pp. 178-181.
- [22] Polo, M., Piattini, M., Ruiz, F., Calero, C., "Roles in the Maintenance Process", *Software Engineering Notes, Special Interest Group on Software Engineering, ACM*, vol. 24, N°4, 1999, pp. 84-86.
- [23] Robillard, P. N., "The Role of Knowledge in Software Development", *Communications of the ACM*, vol. 42, 1999, pp. 87-92.
- [24] Rodríguez, O. M., Martínez, A. I., Favela, J., Vizcaíno, A., Piattini, M., "Understanding and Supporting Knowledge Flows in a Community of Software Developers", *Proc. of the 10th International Workshop on Groupware (CRIWG'2004)*, San Carlos, Costa Rica, 2004, pp. 52-66.
- [25] Rodríguez-Elias, O. M., Martínez-García, A. I., Vizcaíno, A., Favela, J., Piattini, M., "Identifying Knowledge Flows in Communities of Practice", In *Encyclopedia of Communities of Practice in Information and Knowledge Management*, Coakes, E. and Clarke, S. A., Eds. Idea Group Inc., Hershey, PA, USA, 2005, pp. 210-217.
- [26] Ruiz, F., Vizcaíno, A., Piattini, M., García, F., "An Ontology for the Management of Software Maintenance Projects", *International Journal of Software Engineering and Knowledge Engineering*, vol. 14, 2004, pp. 323-349.
- [27] Rus, I., Lindvall, M., Sinha, S. S., "Knowledge Management in Software Engineering: A State of the Art Report", *Data & Analysis Center for Software: ITT Industries*, Rome, NY 29 November 2001.
- [28] Seaman, C., "The Information Gathering Strategies of Software Maintainers", *Proc. of the International Conference on Software Maintenance (ICSM'2002)*, 2002, pp. 141-149.
- [29] Uschold, M., Gruninger, M., "Ontologies: Principles, methods, and applications", *The Knowledge Engineering Review*, vol. 11, 1996, pp. 93-136.
- [30] Valente, A., Housel, T., "A Framework to Analyze and Compare Knowledge Management Tools", *Proc. of the Knowledge-Based Intelligent Information Engineering and Allied Technologies (KES'2002)*, Crema, Italy, 2002, pp. 291-295.
- [31] Waiz, D. B., Elam, J. J., Curtis, B., "Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration", *Communications of the ACM*, vol. 36, 1993, pp. 63-77.
- [32] Especificación del Metamodelo para ingeniería de procesos de software: tomado de Internet: SPEM (<http://www.omg.org>).