

**CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN SUPERIOR
DE ENSENADA**



**PROGRAMA DE POSGRADO EN CIENCIAS
EN CIENCIAS DE LA COMPUTACIÓN**

**METODOLOGÍA PARA EL DISEÑO DE SISTEMAS DE ADMINISTRACIÓN DEL
CONOCIMIENTO: SU APLICACIÓN EN MANTENIMIENTO DE SOFTWARE**

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de
DOCTOR EN CIENCIAS

Presenta:

OSCAR MARIO RODRÍGUEZ ELIAS

Ensenada, Baja California, México, Noviembre de 2007.

RESUMEN de la tesis de **Oscar Mario Rodríguez Elías**, presentada como requisito parcial para la obtención del grado de DOCTOR EN CIENCIAS en CIENCIAS DE LA COMPUTACIÓN. Ensenada, Baja California. Noviembre de 2007.

**METODOLOGÍA PARA EL DISEÑO DE SISTEMAS DE ADMINISTRACIÓN
DEL CONOCIMIENTO: su Aplicación en Mantenimiento de Software**

Resumen aprobado por:

Dra. Ana Isabel Martínez García
Codirector

Dra. Aurora Vizcaíno Barceló
Codirector

El conocimiento es considerado hoy en día como uno de los principales recursos de las organizaciones. Esto es particularmente cierto en las organizaciones dedicadas al desarrollo de software, dado que gran parte de las actividades que en estas se realizan están altamente basadas en el conocimiento que deben tener los encargados de llevarlas a cabo. Debido a esto, lograr una buena administración del conocimiento (AC) dentro de estas organizaciones es de gran importancia para que logren mantenerse en niveles competitivos en una sociedad cada vez más globalizada y demandante.

Si bien la AC no sólo concierne a cuestiones tecnológicas, ha sido la tecnología la que ha logrado su crecimiento y aceptación. Sin embargo, existen factores que provocan que los sistemas de AC tradicionales no sean del todo útiles en muchos casos. Las causas de lo anterior comúnmente se relacionan con el hecho de que estos sistemas no son diseñados tomando en cuenta las necesidades de conocimiento reales de los trabajadores.

Lograr desarrollar sistemas de AC que consideren el trabajo realmente desempeñado dentro de los procesos de las organizaciones es importante para lograr la aceptación de los mismos. Esto es de particular relevancia en las pequeñas organizaciones, dado que frecuentemente es difícil que cuenten con los recursos suficientes para realizar cambios grandes en sus procesos con el fin de introducir sistemas o estrategias de AC. Incluso, las pequeñas organizaciones antes de desarrollar o adquirir nuevos sistemas deben ser capaces de identificar el papel que juega la AC en sus actividades y sistemas actuales, buscando integrar éstos sistemas como parte de las estrategias de AC que se definan, de tal forma que éstas se alineen con las actividades que diariamente desempeñan los trabajadores.

En esta tesis se propone un enfoque metodológico para ayudar en el logro del objetivo anterior. Es decir, apoyar en el estudio de procesos organizacionales con el fin de obtener información que permita proponer diseños de sistemas o estrategias de AC que consideren las necesidades de conocimiento reales de los trabajadores y las herramientas que apoyan las actividades diarias. Esta metodología se compone de una serie de etapas y pasos que persiguen tres metas generales: 1) modelar el proceso desde un punto de vista del flujo del conocimiento dentro del mismo, 2) analizar el conocimiento, sus fuentes, su flujo y la forma en que estos elementos interactúan con las actividades del proceso, y 3) identificar los sistemas o herramientas que apoyan al proceso, desde el punto de vista del papel que éstas juegan, o podrían desempeñar, como sistemas de apoyo al flujo del conocimiento. Finalmente, la información obtenida del logro de estas metas, deberá servir de apoyo para el diseño de sistemas o estrategias de AC que busquen mejorar el flujo del conocimiento dentro del proceso.

Para apoyar la aplicación de la metodología, se proponen también un conjunto de técnicas en cada una de sus fases. Estas técnicas, y en general la metodología, fueron puestas en práctica en un caso de estudio en el dominio de un proceso de software, con el fin de medir sus posibles beneficios y limitaciones. Los resultados de este estudio proporcionan evidencia que sugiere que la metodología realmente puede apoyar en el logro del objetivo para el que fue diseñada, es decir, apoyar en el diseño de sistemas o estrategias de AC que consideren las necesidades de conocimiento de los trabajadores que desempeñan un proceso, así como las herramientas que éstos utilizan para realizar su trabajo diario.

Palabras clave: Administración del Conocimiento, Ingeniería del Software, Ingeniería de Procesos, Mantenimiento de Software, Flujo del Conocimiento

ABSTRACT of the thesis presented by **Oscar Mario Rodríguez Elías**, as a partial requirement to obtain the DOCTOR of SCIENCE degree in COMPUTER SCIENCE. Ensenada, Baja California, Mexico, November 2007.

**METHODOLOGY FOR THE DESIGN OF KNOWLEDGE MANAGEMENT
SYSTEMS: an Application in Software Maintenance**

Approved by:

Dra. Ana Isabel Martínez García
Thesis Co-Advisor

Dra. Aurora Vizcaíno Barceló
Thesis Co-Advisor

Nowadays, knowledge has become one of the main organizational resources. This is particularly true for software organizations, since much of their activities are highly based on the knowledge that employees in charge of these activities must have to accomplish them properly. Thus, having a good knowledge management (KM) strategy in these organizations is very important in order to stay in competitive levels on this day to day more demanding and globalized society.

Although KM is not only a matter of technology, it has been the technology which has contributed the most to the increment on the use and acceptance of KM in organizations. Nevertheless, there are some factors which do not permit the use of traditional KM systems in many cases. Some reasons for the latter are commonly related to the fact that these systems are not designed taking into account the real knowledge needs of the knowledge workers.

Developing KM systems considering the real work done in organizational processes is important to get them accepted by their potential users. This is of particular relevance in small and medium size organizations, since it is often difficult for those organizations to have enough resources for performing big changes into their processes to include KM systems or strategies. In fact, small organizations should be capable of identifying the role that KM plays in their activities and current systems before developing or acquiring new tools. They should search for ways to integrate these current systems as part of the KM strategies designed, in such a way that these should be aligned to the every day work carried out by their employees.

In this thesis, a methodological approach to achieve the last objective is presented. The methodology is oriented to aid in the study of organizational processes to obtain

information useful to propose the design of KM systems and strategies which consider the real knowledge needs of the knowledge workers; also considering the tools being used by them to achieve their job's activities. The methodology is composed of a set of steps that pursue three general goals: 1) to model organizational processes from the point of view of the knowledge that flows through them, 2) to analyze the knowledge, sources, their flow and the manner in which all these interact with the activities of the processes, and 3) to identify the tools or systems that support the process and that can play an important role as knowledge flow enablers. The information obtained by accomplishing the last goals, should be finally useful to aid in the design of KM systems or strategies focused on improving the knowledge flow within the studied processes.

To help in the application of the methodology, some techniques are also proposed for its different stages. These techniques and the methodology in general, were taken into practice in a case study in the domain of a software process, to evaluate their possible benefits and limitations. The results of this case study have given us useful insights into believing that the methodology is really useful to accomplish the objective for which it was developed, which is to aid in the design of KM systems or/and strategies which consider the knowledge needs of the knowledge workers, and the tools they use to accomplish their daily work.

Keywords: Knowledge Management, Software Engineering, Process Engineering, Software Maintenance, Knowledge Flow

Dedicatoria

A mi esposa, mis padres y mis hermanos.

Sin su apoyo no lo hubiera logrado.

Agradecimientos

Primeramente quiero dar gracias a Dios por darme la fuerza necesaria para concluir este trabajo.

Quiero dar un agradecimiento muy especial a mis dos directoras de tesis: Dra. Ana Isabel Martínez García y Dra. Aurora Vizcaíno Barceló. Aun siendo el primero de los hijos mayores, supieron guiarme en esta difícil tarea. Gracias de todo corazón, espero que durante este nuevo camino que inicia continuemos fomentando esta amistad y colaboración que ha nacido entre nosotros.

A los miembros de mi comité de tesis Dr. José Antonio García Macías, Dr. Guillermo Licea Sandoval, y muy especialmente al Dr. Jesús Favela Vara, cuyos consejos y comentarios acertados contribuyeron en gran medida al desarrollo de esta tesis y a los resultados derivados de ella.

A todo el personal del Departamento de Ciencias de la Computación de CICESE, con quienes conviví durante estos más de seis años, y que compartieron conmigo sus experiencias y conocimientos. Especialmente al Dr. Gilberto López Mariscal; espero no haberlo defraudado por la tardanza en la culminación de esta tesis.

Al grupo ALARCOS de la Escuela Superior de Informática de la Universidad de Castilla-La Mancha, en Ciudad Real, España, por haberme recibido y apoyado durante mis dos estancias académicas. En especial, al Dr. Mario Piattini, quien contribuyó de gran forma para el logro de esta tesis, primero observando acertadamente la necesidad de administrar el conocimiento durante el mantenimiento de software, y después por sus ideas y observaciones. Así mismo a los tesistas del grupo, con quienes pasé muchos buenos momentos allá en la madre patria: Juan Pablo, Mateus, Elvira, José Antonio, Nacho, María Ángeles, Luís, María Angélica y Alfonso, Gabriela, Rodolfo. Y finalmente, pero no menos importante, a Aurora y Javier, que me llevaron a conocer su tierra.

A todos mis compañeros de generación de la maestría, y todas esas otras generaciones que a lo largo de este tiempo convivieron conmigo, especialmente aquellos que han tenido la suerte de convertirse en Colabos (ahora Calafios), que hicieron mi estancia en el CICESE más amena.

A todos los amigos que hice en Ensenada, especialmente a Domitilo, Polo, Pedro, Edgar y Emiliano, con quienes conviví en la Casa del Estudiante Sinaloense.

A mi esposa y amiga, Ma. Guadalupe Vaal, que tuvo la fuerza y la voluntad de ser paciente, y me dio fuerza con su amor y manteniéndose a mi lado, aun en la distancia.

A mis padres, Oscar Mario y Lourdes, que supieron inculcarme el gusto por el estudio y por prepararme día con día, no sólo en lo profesional, sino en la vida; y a mis hermanos, Andrés y su esposa Betty, Lulú, Malti y Jesús.

Al CICESE por el apoyo económico para los viajes que realicé, y por proporcionar los recursos necesarios para poder culminar esta tesis.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca otorgada desde Septiembre de 2003 a Febrero de 2007.

Y a todos aquellos que de una u otra forma intervinieron en el logro de esta meta y en facilitar mi estancia en Ensenada durante el trayecto.

Tabla de Contenido

Dedicatoria	vii
Agradecimientos	viii
Tabla de Contenido	x
Lista de Figuras	xv
Lista de Tablas	xix
Capítulo I Introducción	1
<i>I.1 Planteamiento del problema</i>	<i>3</i>
I.1.1 Necesidades de AC en el mantenimiento del software.....	3
I.1.2 Problemas con los enfoques tradicionales de AC.....	5
I.1.2.1 <i>La importancia de los procesos de trabajo y las tecnologías que los apoyan</i>	<i>6</i>
I.1.2.2 <i>Pequeñas y grandes empresas</i>	<i>7</i>
<i>I.2 Objetivo de la investigación</i>	<i>8</i>
<i>I.3 Preguntas de investigación</i>	<i>8</i>
<i>I.4 Metodología de investigación</i>	<i>10</i>
I.4.1 Revisión bibliográfica.....	11
I.4.2 Aplicación a casos ejemplo.....	13
I.4.3 Experimento.....	14
I.4.4 Aplicación en un caso real	15
I.4.5 Algunos comentarios finales sobre la evolución de la investigación	15
<i>I.5 Contribuciones principales de esta tesis</i>	<i>17</i>
<i>I.6 Contenido del documento</i>	<i>18</i>
Capítulo II Marco teórico	21
<i>II.1 Administración del conocimiento</i>	<i>21</i>
II.1.1 Definición de conocimiento.....	22
II.1.2 El flujo del conocimiento.....	26
II.1.3 Modelos de flujo del conocimiento	27
II.1.3.1 <i>El proceso de creación del conocimiento</i>	<i>27</i>
II.1.3.2 <i>El ciclo del conocimiento</i>	<i>29</i>
II.1.4 El proceso de administración del conocimiento	31
II.1.5 Modelo integrado.....	34
II.1.5.1 <i>Creación/Adquisición del conocimiento</i>	<i>35</i>
II.1.5.2 <i>Externalización del conocimiento</i>	<i>36</i>
II.1.5.3 <i>Almacenamiento/Mantenimiento del conocimiento</i>	<i>37</i>
II.1.5.4 <i>Transmisión del conocimiento</i>	<i>38</i>
II.1.5.5 <i>Internalización del conocimiento</i>	<i>39</i>
II.1.5.6 <i>Aplicación/Evolución del conocimiento</i>	<i>40</i>
<i>II.2 Administración del conocimiento con enfoque en procesos organizacionales</i>	<i>42</i>
<i>II.3 Administración del conocimiento en ingeniería de software</i>	<i>43</i>
<i>II.4 Requerimientos para el estudio del proceso de mantenimiento de software con enfoque en la AC</i>	<i>46</i>
<i>II.5 Resumen del capítulo</i>	<i>48</i>

Tabla de Contenido

(Continuación)

Capítulo III KoFI: una metodología para el estudio de flujos de conocimiento	50
<i>III.1 Descripción general de KoFI</i>	<i>51</i>
<i>III.2 Fases de la metodología KoFI.....</i>	<i>54</i>
<i>III.3 Modelado del proceso con enfoque en el flujo de conocimiento.....</i>	<i>55</i>
<i>III.4 Análisis de flujos de conocimiento.....</i>	<i>57</i>
III.4.1 Escenario ejemplo.....	58
III.4.2 Identificación de fuentes de conocimiento	59
III.4.2.1 Clasificación de fuentes de conocimiento.....	61
III.4.2.2 Descripción de fuentes de conocimiento	63
III.4.3 Identificación de tipos de conocimiento	66
III.4.3.1 Clasificación de tipos de conocimiento	69
III.4.3.2 Descripción de tipos de conocimiento	71
III.4.4 Identificación de flujos de conocimiento.....	75
III.4.5 Identificación de problemas en el flujo del conocimiento.....	76
<i>III.5 Análisis de herramientas de soporte al flujo del conocimiento.....</i>	<i>80</i>
<i>III.6 Resumen del capítulo</i>	<i>82</i>
Capítulo IV Modelado de procesos con enfoque en el flujo de conocimiento.....	83
<i>IV.1 Modelado de flujos de conocimiento en procesos de ingeniería de software</i>	<i>83</i>
<i>IV.2 Adaptación de la técnica de Gráfica Rica</i>	<i>88</i>
IV.2.1 Descripción de la técnica de Gráfica Rica	89
IV.2.2 Adaptaciones propuestas.....	91
<i>IV.3 SPEM-KF: Adaptación del Metamodelo para Ingeniería de Procesos de Software (SPEM).....</i>	<i>94</i>
IV.3.1 Descripción de SPEM.....	95
IV.3.2 Extensión de SPEM	100
IV.3.2.1 El conocimiento como un producto del trabajo.....	101
IV.3.2.2 Relaciones y dependencias entre los conceptos de conocimiento	104
IV.3.2.3 Notación.....	106
IV.3.2.4 Diagramas	107
<i>IV.4 Recomendaciones en el uso de los diagramas.....</i>	<i>120</i>
<i>IV.5 Resumen del capítulo.....</i>	<i>122</i>
Capítulo V MAHAC: Un marco de trabajo para el análisis de herramientas de apoyo a la administración del conocimiento.....	124
<i>V.1 Características para el análisis de herramientas de administración del conocimiento.....</i>	<i>124</i>
V.1.1 Definir el Dominio de aplicación.....	130
V.1.2 Definir el tipo y estructura del conocimiento	130
V.1.3 Definir actividades de AC o etapas del flujo de conocimiento apoyadas.....	131
V.1.4 Definir los principales aspectos técnicos	131
V.1.5 Estructura general de MAHAC.....	132
<i>V.2 Dominio de aplicación.....</i>	<i>132</i>

Tabla de Contenido

(Continuación)

V.2.1	El uso del conocimiento.....	133
V.2.2	Alcance del conocimiento.....	134
V.2.3	El dominio del conocimiento.....	135
V.3	<i>Estructura del conocimiento.....</i>	135
V.4	<i>Soporte al flujo del conocimiento.....</i>	138
V.5	<i>Aspectos Técnicos.....</i>	140
V.5.1	Nivel de autonomía.....	141
V.5.2	Distribución del conocimiento.....	143
V.6	<i>Una matriz para la aplicación de MAHAC.....</i>	146
V.7	<i>Resumen del capítulo.....</i>	147
Capítulo VI Resultados: utilización de la metodología.....		149
VI.1	<i>Caso de estudio.....</i>	151
VI.1.1	Contexto del caso de estudio.....	151
VI.1.2	Utilización de la metodología KoFI.....	152
VI.1.2.1	<i>Modelado del proceso.....</i>	153
VI.1.2.2	<i>Identificación y clasificación de fuentes y áreas de conocimiento.....</i>	154
VI.1.2.3	<i>Identificación de flujos de conocimiento.....</i>	163
VI.1.2.4	<i>Problemas con el flujo del conocimiento.....</i>	169
VI.1.2.5	<i>Propuestas para mejorar el flujo del conocimiento.....</i>	170
VI.1.2.6	<i>El Sistema de Control de Proyectos como herramienta de apoyo al flujo del conocimiento.....</i>	173
VI.1.3	Resultados del caso de estudio.....	177
VI.1.3.1	<i>Lecciones aprendidas del caso de estudio.....</i>	178
VI.1.3.2	<i>Limitaciones del caso de estudio.....</i>	180
VI.2	<i>Un segundo caso de estudio.....</i>	181
VI.2.1	Resumen de los resultados del cuestionario.....	181
VI.2.2	Discusión del segundo caso de estudio.....	183
VI.3	<i>Experimento.....</i>	184
VI.3.1	Principales resultados del experimento.....	185
VI.3.1.1	<i>Las extensiones de SPEM permiten identificar el conocimiento involucrado en las actividades.....</i>	185
VI.3.1.2	<i>Las extensiones de SPEM permiten identificar fuentes de conocimiento...186</i>	186
VI.3.1.3	<i>Las extensiones de SPEM permiten identificar transferencias de conocimiento.....</i>	186
VI.3.1.4	<i>La nomenclatura es adecuada.....</i>	187
VI.3.1.5	<i>La experiencia en UML puede facilitar el entendimiento de los modelos..188</i>	188
VI.3.2	Conclusión del experimento.....	188
VI.4	<i>Resumen del capítulo.....</i>	189
Capítulo VII Conclusiones, aportaciones y trabajo futuro.....		191
VII.1	<i>Conclusiones.....</i>	191
VII.2	<i>Aportaciones.....</i>	193

Tabla de Contenido

(Continuación)

VII.2.1	Publicaciones	196
VII.3	<i>Trabajo futuro</i>	200
VII.3.1	Implementación y evaluación de los cambios al sistema de control de proyectos 201	
VII.3.2	Seguimiento del segundo caso de estudio.....	201
VII.3.3	Mejoras en las adaptaciones a SPEM	201
VII.3.4	Desarrollo de una herramienta de apoyo para la aplicación de la metodología	202
VII.3.4.1	<i>Sistema de apoyo para el modelado del proceso</i>	202
VII.3.4.2	<i>Generación de perfiles de conocimiento</i>	203
VII.3.4.3	<i>Automatización de las descripciones de tipos y fuentes de conocimiento</i> ..	204
VII.3.4.4	<i>Automatización de la construcción de mapas de conocimiento</i>	204
VII.3.5	Realización de más casos de estudio	204
Referencias	206
Apéndice A	Experimento	220
A.1	<i>Introducción</i>	220
A.2	<i>Preparación: descripción del experimento</i>	220
A.2.1	Objetivo del experimento.....	221
A.2.2	Hipótesis del experimento	221
A.2.3	VARIABLES A MEDIR	222
A.2.4	Material empleado	224
A.3	<i>Realización del experimento</i>	225
A.3.1	Descripción de la muestra.....	225
A.3.2	Metodología.....	226
A.3.3	Aplicación del experimento.....	227
A.4	<i>Análisis de resultados</i>	228
A.4.1	Método de evaluación.....	229
A.4.2	Resultados.....	230
A.4.3	Discusión de los resultados.....	243
A.5	<i>Conclusiones</i>	248
A.6	<i>Anexos</i>	248
A.6.1	Cuestionario Inicial.....	249
A.6.2	Ejercicio primer grupo (Ejercicio 1a).....	250
A.6.3	Ejercicio segundo grupo (Ejercicio 1b)	256
A.6.4	Cuestionario final.....	258
Apéndice B	Caso de Estudio: Análisis del proceso de mantenimiento del Departamento de Informática del CICESE	263
B.1	<i>Introducción</i>	263
B.2	<i>Descripción del proceso de mantenimiento en el Departamento de Informática del CICESE</i>	263
B.2.1	Roles en el proceso	264
B.2.2	Descripción general del proceso	269

Tabla de Contenido

(Continuación)

<i>B.3</i>	<i>Observaciones y recomendaciones</i>	305
B.3.1	Fuentes de conocimiento en el proceso	306
B.3.2	Áreas de conocimiento involucradas en el proceso	309
B.3.3	Flujos de conocimiento.....	312
B.3.4	Problemas observados en el proceso	312
Apéndice C Segundo caso de estudio: cuestionario		313
<i>C.1</i>	<i>Cuestionario sobre la aplicación de la metodología KoFI</i>	313
C.1.1	A.- Contexto del estudio:	313
C.1.2	B.- Aprendizaje y entendimiento de la metodología	314
C.1.3	C.- Aplicación de la metodología	316
C.1.4	D.- Resultados de la aplicación de la metodología.....	317
C.1.5	E.- Datos generales	319
Apéndice D Modelo del uso de la metodología		320
<i>D.1</i>	<i>Vista general de la metodología KoFI</i>	320
<i>D.2</i>	<i>Fase de definición del proceso</i>	321
D.2.1	Modelado general con las adaptaciones de la Gráfica Rica.....	323
D.2.2	Modelado detallado con SPEM-KF	323
<i>D.3</i>	<i>Análisis del proceso</i>	325
<i>D.4</i>	<i>Análisis de herramientas que intervienen en el flujo del conocimiento</i>	329
Apéndice E Ontologías de fuentes y tipos de conocimiento		331
<i>E.1</i>	<i>Fuentes de conocimiento</i>	331
E.1.1	Taxonomía de fuentes de conocimiento	331
E.1.2	Ontología de fuentes de conocimiento.....	336
<i>E.2</i>	<i>Tipos de conocimiento</i>	337
E.2.1	Taxonomía de tipos de conocimiento	337
E.2.2	Ontología de tipos de conocimiento	342
<i>E.3</i>	<i>Referencias en el apéndice</i>	342

Lista de Figuras

Figura	Página
1 Metodología seguida durante el proceso de investigación.	11
2 Modelo jerárquico del conocimiento.	24
3 Modelo conceptual de las relaciones entre los tipos de conocimiento.	25
4 Mecanismos de conversión del conocimiento	28
5 Integración de la jerarquía D-I-C con el proceso de creación de conocimiento.	29
6 El ciclo del conocimiento [Choo, 1999].	30
7 Extensión del modelo del ciclo del conocimiento de Choo [1999].	31
8 Modelo de interrelación de actividades de AC centrado en la aplicación y transferencia del conocimiento.	35
9 Vista general de la metodología KoFI.	55
10 Vista general del metamodelo de fuentes y tipos de conocimiento.	58
11 Memamodelo de fuentes de conocimiento.	62
12 Ejemplo del uso del metamodelo de fuentes de conocimiento. Presenta un esquema de clasificación para el escenario ejemplo.	63
13 Niveles de abstracción propuestos para clasificar tipos de conocimiento.	69
14 Ejemplo de clasificación de temas de conocimiento. Presenta un esquema de clasificación para un caso en el escenario ejemplo.	71
15 Elementos de la adaptación de la técnica de Gráfica Rica.	91
16 Modelo del escenario ejemplo por medio de una gráfica rica adaptada. Presenta los distintos tipos de conexiones entre los elementos del modelo y el conocimiento involucrado.	92
17 Modelo conceptual base del modelado de procesos con SPEM.	96
18 Elementos principales de la especificación de SPEM.	97
19 Íconos para los principales elementos de SPEM.	97
20 Ejemplos de diagramas basados en SPEM. Da una visión general de los tipos de diagramas que se pueden crear con el lenguaje de modelado propuesto en SPEM.	99
21 Integración de los conceptos de conocimiento como productos del trabajo.	101
22 Metamodelo de conceptos de conocimiento (elementos principales del paquete KnowledgeConcepts).	102
23 Elementos de organización integrados al paquete KnowledgeConcepts.	103
24 Relaciones definidas en el paquete KnowledgeConcepts (Los tres elementos en la parte superior pertenecen a la especificación de SPEM).	105
25 Dependencias que forman parte del paquete KnowledgeConcepts (Los tres elementos en la parte superior de la figura forman parte de la especificación de SPEM.)	106
26 Íconos de la notación de los elementos del paquete KnowledgePackage.	107
27 Ejemplo de un diagrama de casos de uso adaptado.	109
28 Ejemplo de un diagrama de actividad adaptado.	112
29 Ejemplo de diagramas de paquetes de conocimiento	114
30 Ejemplo de diagrama de clases.	115
31 Ejemplo de diagrama del flujo del conocimiento de una fuente.	117
32 Ejemplo de diagrama de flujo de conocimiento entre fuentes.	119

Lista de Figuras

(Continuación)

Figura	Página
33	Etapas del marco de trabajo para el análisis de herramientas de apoyo a la administración del conocimiento.132
34	Diagrama de casos de uso que presenta las principales fases del proceso y principales paquetes de conocimiento requerido y generado en las mismas.154
35	Diagrama de paquetes que clasifica áreas de conocimiento. Se muestran las relaciones de los paquetes de conocimiento (generales y específicos) requeridos en la fase de implementación de cambios.155
36	Diagrama de clases que identifica las relaciones entre fuentes y áreas de conocimiento.157
37	Diagrama de clases que especifica fuentes de conocimiento concreto. Se identifican las fuentes particulares donde pueden ser obtenidos los distintos temas agrupados en el paquete de conocimientos sobre el sistema SIDEPE.158
38	Diagrama de paquetes que identifica conocimiento requerido por los roles o actores en el proceso. Muestra las áreas de conocimiento que debe conocer el encargado de dar mantenimiento al sistema SIDEPE.160
39	Diagrama de clases que identifica relaciones entre fuentes y áreas de conocimiento. Se muestran las fuentes de conocimiento relacionado con los procesos apoyados por el sistema SIDEPE.161
40	Diagrama de actividades que muestra flujos de conocimiento, por medio de la transferencia de fuentes entre actividades y/o roles o actores del proceso.164
41	Diagrama de paquetes que muestra la información de la solicitud de cambios. Se muestran temas concretos que pueden obtenerse de la solicitud de cambios, junto con los relacionados con los requerimientos.165
42	Diagrama de transferencia de conocimiento de una fuente que muestra el flujo del conocimiento relacionado con la solicitud de cambios.166
43	Diagrama de clases que muestra las fuentes que contienen información sobre los cambios realizados.167
44	Diagrama de transferencias de conocimiento entre fuentes que muestra las fuentes que participan en la transferencia de la información relacionada con las solicitudes de actualización de servidores.168
45	Ejemplo de un paquete de conocimientos o información.234
46	Diagramas mostrados al grupo 1.235
47	Diagramas mostrados al grupo 2.236
48	Vista de las relaciones de los distintos roles participantes en el proceso.269
49	Vista general del proceso de mantenimiento del DI.270
50	Caso de uso de la fase de atención a usuarios. Muestra los roles que interactúan en los flujos de trabajo que se realizan en esta fase.271
51	Diagrama de paquetes de la fase de atención a usuarios. Muestra las actividades que desempeña cada rol que participa en dicha fase, así como los principales artefactos que son usados o generados durante la misma.271

Lista de Figuras

(Continuación)

Figura	Página
52	Secuencia de actividades durante el flujo de trabajo de atención de reporte de problema.272
53	Secuencia de actividades del flujo de trabajo atención de solicitudes de cambios.275
54	Página de inicio del sistema de Control de Proyectos del DI.276
55	Ventana de seguimiento de solicitudes en el Sistema de Control de Proyectos.277
56	Ventana para la captura de los datos de la solicitud.278
57	Secuencia de actividades de la etapa de obtención de requerimientos.281
58	Información contenida en la solicitud de cambios.....282
59	Casos de uso de la fase de implementación de cambios. Muestran los roles que interactúan en los flujos de trabajo que se realizan en esta fase.284
60	Diagrama de paquetes de la fase de implementación de cambios. Muestra las actividades que desempeña cada rol que participa en dicha fase, así como los principales artefactos que son usados, modificados o generados.285
61	Secuencia de actividades de la fase de implementación de cambios.....286
62	Ventana donde se muestra la lista de solicitudes capturadas en el SCP.287
63	Ejemplo de solicitud de cambio con estado en Espera de Requerimientos.287
64	Ejemplo de solicitud de cambio con estado En Proceso.....288
65	Ejemplo de la ventana de edición de solicitudes.289
66	Diagrama del paquete que corresponde al conocimiento que se debe tener para poder identificar las partes del sistema a modificar, los posibles efectos secundarios de los cambios, y la forma de realizar los cambios en el sistema.....290
67	Información que puede ser obtenida del script para modificar la base de datos.....291
68	Diagrama de actividad del flujo de trabajo de las pruebas de los cambios.295
69	Diagrama de actividad de la opción de pruebas con datos reales.....296
70	Modelo de capas en el control de versiones en el DI del CICESE.....298
71	Casos de uso de la fase de libración de cambios. Muestran los roles que interactúan en los flujos de trabajo que se realizan esta fase.....298
72	Diagrama de actividad de la etapa de actualización de servidores.299
73	Información contenida en la solicitud de modificación de la base de datos.....300
74	Información contenida en la solicitud de actualización de archivos.301
75	Vista del repositorio de manuales de usuario de las modificaciones que realiza el DI.303
76	Ejemplo de un manual de usuario de un módulo modificado por el DI.304
77	Vista general de las fases y pasos que constituyen la metodología KoFI.....321
78	Actividades de la descripción del proceso.....322
79	Información que debe poder obtenerse del modelo general del proceso.323
80	Modelado detallado del proceso con SPEM-KF (parte 1/2).....324
81	Modelado detallado del proceso con SPEM-KF (parte 2/2).....325
82	Etapa de identificación de fuentes de conocimiento.....326
83	Etapa de identificación de tipos de conocimiento.327

Lista de Figuras

(Continuación)

Figura		Página
84	Etapa de identificación de flujos de conocimiento.	328
85	Etapa de identificación de problemas en el flujo del conocimiento.	329
86	Fase de análisis de herramientas de apoyo al flujo del conocimiento.	330
87	Esquema general de la ontología de fuentes de conocimiento.	336

Lista de Tablas

Tabla	Página
I Modelos del proceso de administración del conocimiento.....	33
II Plantilla para la definición de fuentes de conocimiento. Muestra información sobre un tipo de fuente hipotética extraída del escenario ejemplo.....	64
III Plantilla para de descripción de temas de conocimiento. La información mostrada corresponde con un caso hipotético extraído del escenario ejemplo.....	73
IV Ejemplo de descripción de un escenario de problema.....	79
V Recomendaciones en el uso de los distintos tipos de diagramas.....	121
VI Propuestas para comparar, caracterizar o clasificar sistemas de AC.....	127
VII Esquema para clasificar los procesos y actividades apoyados por las tecnologías de AC.....	133
VIII Estructura de niveles de granularidad para definir el alcance de una herramienta de AC.....	134
IX Esquema para clasificar el dominio del conocimiento.....	135
X Esquema de clasificación de estructuras de conocimiento.....	137
XI Conjunto de valores para definir el grado en que una herramienta interfiere con las etapas del flujo del conocimiento.....	139
XII Parámetros usados para medir el grado en que una herramienta de AC soporta la autonomía y la distribución del conocimiento.....	141
XIII Matriz de definición de características consideradas en MAHAC.....	147
XIV Ejemplo de descripción de una fuente de conocimiento. Presenta información sobre el documento “Formas_y_objetos_de_base_de_datos.doc”, el cual describe la estructura interna del sistema SIDEPE.....	159
XV Descripción del SCP como facilitador del flujo del conocimiento usando la matriz de definición de características consideradas en MAHAC.....	174
XVI Datos generales de los participantes en el experimento.....	228
XVII Valores para la medición del grado de acierto a las preguntas del ejercicio.....	229
XVIII Escala para medir la apreciación de los participantes.....	230
XIX Tiempos empleados por los participantes en la realización de la primera parte del ejercicio.....	231
XX Lista de preguntas hechas a los participantes en el ejercicio.....	232
XXI Grado de acierto en las respuestas dadas por los participantes en las preguntas....	233
XXII Promedio de calificación de los participantes, comparado con el tiempo total empleado en el ejercicio, y datos generales de los participantes.....	237
XXIII Comparación entre la calificación dada a la descripción del proceso hecha por los participantes, y su apreciación con respecto a la similitud con la descripción real del proceso, y de ésta última con los diagramas.....	238
XXIV Respuestas a las preguntas de apreciación de los participantes.....	240
XXV Muestra la apreciación de los participantes con respecto a distintos adjetivos asignados a los distintos elementos del lenguaje de modelado empleado en el ejercicio.....	241
XXVI Evaluación de los participantes con respecto a la utilidad de los distintos tipos de diagrama.....	242

CAPÍTULO I

Introducción

“Qué es la investigación, sino una cita a ciegas con el conocimiento” Will Henryⁱ

“Si hubiéramos sabido qué era lo que estábamos haciendo, no se le habría llamado investigación” Albert Einsteinⁱⁱ

En la actualidad, el conocimiento se ha convertido en un recurso de particular valía para muchas organizaciones, incluso llegando a ser considerado, en algunos casos, más valioso que los bienes materiales [Dalkir, 2005; Davenport y Prusak, 2000; Nonaka y Takeuchi, 1995]. Debido a esto, el manejo adecuado del conocimiento organizacional se ha vuelto una importante área de oportunidad que ha provocado un enorme interés, el cual puede verse claramente en la cantidad de publicaciones en revistas, libros y conferencias que existen y siguen surgiendo constantemente [Serenko y Bontis, 2004]. El término Administración del Conocimiento (AC) se ha acuñado para agrupar todos esos esfuerzos encaminados a lograr el mejor aprovechamiento del conocimiento en las organizaciones.

En años recientes, los expertos en el área se han percatado de que los enfoques de AC tradicionales se quedan cortos con respecto a las verdaderas necesidades de muchas organizaciones. Para disminuir este problema, Wiig [2004] sostiene que se requiere poner especial interés en las necesidades de los “trabajadores del conocimiento”, es decir, aquellos que requieren o generan dicho conocimiento en su trabajo diario; idea que es apoyada por otros autores, por ejemplo Dalkir [2005: p. 21]. Lo anterior ha llevado a la necesidad de enfoques que permitan identificar los requerimientos de conocimiento que existen dentro de los procesos de trabajo de las organizaciones [Maier y Remus, 2002]. Este tipo de propuestas sostienen que las iniciativas de AC deben estar centradas en los procesos de trabajo diario. Más aún, la integración de la AC en dichos procesos está siendo

considerada como una de las principales necesidades para el presente y futuro mediato [Davenport, 2007: p. 101; Scholl *et al.*, 2004].

Lo anterior se hace todavía más evidente en el caso de las pequeñas y medianas empresas, las cuales por lo general no tienen los recursos suficientes para involucrarse en estrategias de AC tradicionales que requieren cambios profundos en los modos de trabajar, así como de la adquisición de nuevas herramientas de trabajo [Sparrow, 2001; Wong, 2005]. En cambio, se ha recomendado que las pequeñas empresas, primero deben identificar las implicaciones que tiene la AC en su trabajo cotidiano, en sus procesos y sistemas, así como en las herramientas tecnológicas que los soportan [Sparrow, 2001]; deben buscar la manera de adaptar la AC a su actual cultura organizacional antes que involucrarse en un cambio de la misma [Wong, 2005]; y deben buscar aprovechar al máximo posible la infraestructura tecnológica existente dentro de sus procesos [Davenport *et al.*, 1998]. Para lograr lo anterior es necesario proveer a dichas empresas con instrumentos que les permitan analizar tanto sus procesos de trabajo como la infraestructura tecnológica que los apoyan, desde el punto de vista del papel que puede jugar la AC en dichos procesos y tecnologías.

En esta tesis se propone una metodología encaminada a lograr el objetivo anterior. La aplicación de la metodología se ejemplifica por medio de un proceso del área de la Ingeniería de Software, puesto que las actividades de desarrollo de software son especialmente intensivas en conocimiento [Chapin, 1987; Robillard, 1999]. Esto convierte a los procesos de software en un escenario interesante para estudiar la manera de facilitar el análisis de procesos organizacionales y su infraestructura tecnológica con un enfoque de AC. Por lo anterior, se ha elegido el proceso de mantenimiento de software como un caso adecuado para analizar la utilidad de una metodología que apoye en el estudio de procesos de software desde un enfoque del papel que en ellos puede jugar la administración del conocimiento.

En este capítulo introductorio se describen los aspectos que motivan el trabajo realizado, así como los elementos que guiaron la realización del mismo. El resto del

contenido del capítulo consiste en lo siguiente: primeramente se presenta el planteamiento del problema, donde se indica la necesidad de un enfoque metodológico para apoyar a las organizaciones de desarrollo de software en el estudio de sus procesos, con el fin de definir estrategias de AC que consideren el trabajo que realizan los miembros de estas organizaciones, y la infraestructura tecnológica de la que se valen para su trabajo. Debido a lo extenso del ciclo de desarrollo de software, en esta tesis sólo nos centramos en el proceso de mantenimiento, sobre todo porque esta es la etapa que mayores recursos requiere dentro de las organizaciones de software [Polo *et al.*, 2003b], además de que muchos de sus problemas pueden ser considerados como problemas de administración de conocimiento [Anquetil *et al.*, 2007]. A partir de este análisis, se observa la necesidad de un enfoque metodológico de las características indicadas, no sólo para organizaciones de desarrollo de software, sino para organizaciones en lo general. Una vez planteado el problema, en segundo término se describen las preguntas de investigación que guiaron este trabajo, para posteriormente detallar los objetivos perseguidos durante el mismo. En seguida se detalla la metodología de trabajo seguida durante el desarrollo de esta investigación, y las principales contribuciones obtenidas durante la misma, para finalizar con la descripción del contenido del resto de este documento.

I.1 Planteamiento del problema

El planteamiento del problema al que se ha enfocado el trabajo reportado en este documento ha sido dividido en tres partes: 1) las necesidades de AC en grupos de mantenimiento del software, que es el área en la que se ha elegido mostrar la utilidad de las propuestas surgidas de esta investigación; 2) las dificultades de los enfoques tradicionales de AC; y 3) las diferencias que presentan las pequeñas empresas de desarrollo de software con respecto a sus necesidades en cuestiones de administración del conocimiento.

I.1.1 Necesidades de AC en el mantenimiento del software

La ingeniería del software es una profesión donde el conocimiento de quien la realiza juega un papel prioritario. Por ejemplo, los encargados del mantenimiento de software, además de conocimientos técnicos sobre las herramientas utilizadas para

desempeñar su trabajo (lenguajes de programación, ambientes de desarrollo, modelos de desarrollo, etc.) requieren conocer la estructura del sistema que van a modificar, los posibles efectos de los cambios que realicen, entre otras tantas cosas [Chapin, 1987; Seaman, 2002]; incluso se ha dicho que el mantenimiento del software es más intensivo en conocimiento que el desarrollo del mismo [Chapin, 1987], debido a las dificultades derivadas de tener que entender sistemas complejos que muy frecuentemente fueron desarrollados por personas distintas a quienes deben mantenerlos [Mayrhauser y Vans, 1995].

Como podemos observar de lo anterior, dentro de los procesos llevados a cabo en una organización de desarrollo de software, el mantenimiento de software presenta una serie de características y dificultades que hacen que la AC se vuelva de especial interés para este proceso. Los encargados del mantenimiento del software durante la realización de sus tareas se enfrentan a problemas como falta de documentación [Dart *et al.*, 1993; Singer, 1998] y el hecho de que conforme los sistemas envejecen, la documentación se vuelve menos confiable por no ser actualizada. También, con el paso del tiempo es más difícil contar con la ayuda de los desarrolladores originales [Munson, 1998] para consultarlos cuando la documentación no es suficiente. Por otra parte, es común que sean desarrolladores recién egresados de las escuelas o universidades quienes están a cargo de las labores de mantenimiento [Thomsett, 1998], por lo que pueden no contar con la experiencia suficiente que los lleve a realizar sus tareas de la mejor manera. Adicionalmente, aun cuando existan fuentes de información y conocimiento que puedan ser útiles para los encargados del mantenimiento, con frecuencia estas son desconocidas, o difíciles de localizar para los interesados [Seaman, 2002].

Los problemas anteriores hacen evidente la necesidad de proveer a los grupos de mantenimiento de software, de sistemas que les ayuden a administrar el conocimiento que requieren para sus actividades. Incluso, hay quienes consideran que muchos de los problemas del mantenimiento de software pueden ser vistos como problemas de administración del conocimiento [Anquetil *et al.*, 2007]. Más importante aún, el nivel de madurez de una organización de mantenimiento de software parece estar relacionada, de

una u otra manera, con la aplicación de técnicas o estrategias que pueden ser consideradas dentro del ámbito de la administración de conocimiento [Chapin, 2003].

La importancia de abordar los problemas de AC del mantenimiento en organizaciones de software se vuelve más evidente al considerar la gran cantidad de recursos que deben invertir estas organizaciones en dicha actividad, llegando en algunos casos a niveles mayores al 90%, con una tendencia al alza con el paso de los años [Polo et al., 2003b: p. vii]. Quizá por estas razones existe quien sostiene que los nuevos paradigmas de ingeniería del software deben centrarse en el mantenimiento y evolución del software, más que en su desarrollo inicial [Rajlich, 2006].

Por todo lo anterior, se hace evidente que lograr una buena administración del conocimiento existente en las organizaciones de desarrollo, y en especial de mantenimiento de software puede generar grandes beneficios; entre los que se pueden incluir incrementos en la productividad, mejoras en la calidad de los productos, reducción de costos, mejora de los procesos, etc. [Basili y Caldiera, 1995; Dingsøyr y Conradi, 2002; Lindvall y Rus, 2003]. Sin embargo, existen muy pocos trabajos que abordan el problema de la AC en el mantenimiento del software [Oliveira *et al.*, 2003]. Por lo tanto, es necesario buscar alternativas que ayuden a estudiar el proceso de mantenimiento de software de una organización con el fin de definir estrategias de AC.

Los sistemas de AC tradicionales presentan problemas que pueden ser una barrera para su aceptación por parte de los miembros de las organizaciones de software [Aurum *et al.*, 2007; Desouza, 2003; Rus *et al.*, 2001], por lo que se requieren mecanismos que apoyen en el diseño de sistemas y/o estrategias de AC que consideren las necesidades particulares de este tipo de grupos.

I.1.2 Problemas con los enfoques tradicionales de AC

La problemática con los enfoques de AC tradicionales ha sido dividida en dos aspectos: en primer lugar, la falta de enfoques que busquen integrar la AC en los procesos

de trabajo diario, así como con la infraestructura tecnológica que apoya dichos procesos; y en segundo lugar, la necesidad de enfoques orientados a las pequeñas empresas.

I.1.2.1 La importancia de los procesos de trabajo y las tecnologías que los apoyan

De las propuestas existentes, la gran mayoría se centra en facilitar el desarrollo de nuevos sistemas de AC. Muchas de ellas no toman en cuenta el estudio de los procesos de trabajo, y no hemos encontrado una que tome en consideración la infraestructura tecnológica con la que cuenta en el momento la organización como un paso en el análisis de las alternativas de AC. Esto es importante debido a que comúnmente algunas organizaciones cuentan con herramientas de trabajo que de una u otra manera pueden ser consideradas como sistemas de AC, como es el caso de las organizaciones de desarrollo de software [Lindvall y Rus, 2003]. Así mismo, si uno estudia con detalle los procesos de una organización de este último tipo, puede darse cuenta que sus miembros realizan ciertas actividades de AC aún cuando no lo hagan concientemente [Meehan y Richardson, 2002; Ward y Aurum, 2004]. Aun así, los enfoques tradicionales no toman en consideración esto.

Diversos autores en el área de la AC han notado que las estrategias de AC deben comenzar a centrarse en las necesidades de los trabajadores del conocimiento, es decir, aquellos que requieren dicho conocimiento para realizar sus actividades diarias [Dalkir, 2005: p. 21; Wiig, 2004]. Incluso, expertos en el área de la AC apuntan que uno de los principales retos a futuro es integrar la AC en los procesos de trabajo diario de las organizaciones [Scholl et al., 2004]. Davenport [2007] sostiene que actualmente la forma más prometedora de lograr esta integración es por medio de las tecnologías que dichos trabajadores usan diariamente.

De lo anterior se deriva que los nuevos enfoques de AC deben centrarse en las necesidades de conocimiento de las tareas diarias de una organización, por lo que dichos enfoques deben primero analizar las necesidades de conocimiento de los trabajadores, y en dichas tareas buscar integrar las alternativas de AC con los procesos de trabajo [Maier y

Remus, 2002]. Así mismo, se debe considerar la infraestructura tecnológica que apoya la realización de dichos procesos como una parte importante de estas estrategias.

I.1.2.2 Pequeñas y grandes empresas

Los enfoques de AC tradicionales han sido desarrollados tomando en consideración las necesidades de las grandes empresas, como se puede observar en los ejemplos que presenta la literatura (i.e. [Ichijo y Nonaka, 2007; Nonaka y Takeuchi, 1995; Rao, 2005b; Tiwana, 2000]), y lo mismo se ve en el caso de la AC en la ingeniería del software, donde uno de los enfoques más utilizados es la Fabrica de Experiencias [Basili *et al.*, 1994; Dingsøyr y Conradi, 2002; Schneider *et al.*, 2002]. Este enfoque requiere de un grupo especializado en la recopilación, empaquetado, captura, mantenimiento, etc. de las experiencias y conocimiento, lo que requiere de inversiones considerables que no pueden ser apoyadas por las pequeñas empresas de desarrollo de software. Existen diversos aspectos que causan que las pequeñas organizaciones de desarrollo de software tengan necesidades diferentes a las grandes [Richardson y von Wangenheim, 2007]. Para empezar, tienen menos recursos, que con frecuencia son insuficientes para que puedan arriesgarse con enfoques de AC novedosos que requieran nuevas herramientas, grandes cambios en los procesos, etc., o que requieran tener personal especializado encargado de crear, mantener, capturar, etc. una base de conocimientos o experiencias.

Por otro lado, las pequeñas empresas deben tener muy en cuenta que una mala integración de las iniciativas de AC en su cultura organizacional sería una importante barrera para la adopción de las mismas [Wong, 2005]. Por lo tanto, deben asegurarse de que dichas iniciativas encajen con su forma de trabajo [Davenport *et al.*, 1998], debido a que difícilmente tendrán los recursos para aventurarse en cambios de cultura organizacional que les requieran mucho tiempo. Antes de desarrollar o adquirir nuevos sistemas de AC que requieran hacer cambios en la forma de trabajar, las pequeñas organizaciones deben empezar a ver el papel que juega el conocimiento en su trabajo cotidiano; deben comenzar a ver las implicaciones que tiene la AC en sus procesos actuales, así como en la infraestructura tecnológica que usan para realizar dichos procesos [Sparrow, 2001].

De todo lo anterior, se hace evidente la necesidad de un enfoque que permita a las pequeñas empresas, y en particular las de desarrollo y mantenimiento de software, analizar sus procesos desde el punto de vista del papel que juega el conocimiento en ellos, considerando las herramientas de trabajo que pudieran ser de utilidad como parte de una estrategia de AC. Además, si tomamos en cuenta que la mayoría (más del 85%) de las organizaciones de desarrollo de software están dentro del rango de pequeñas o medianas [García *et al.*, 2006; Richardson y von Wangenheim, 2007], lo anterior se hace más relevante.

A partir del análisis de las necesidades observadas en toda esta sección, se propone el siguiente objetivo.

I.2 Objetivo de la investigación

Diseñar un marco metodológico que apoye en la identificación, modelado y representación del conocimiento, sus fuentes y su flujo dentro de procesos de desarrollo y mantenimiento de software, así como la identificación y análisis de las herramientas que intervienen en el flujo del conocimiento en dichos procesos, con el fin de facilitar el diseño de sistemas o estrategias encaminadas a mejorar los flujos de conocimiento que se dan dentro de estos procesos.

I.3 Preguntas de investigación

Hemos visto que una de las necesidades existentes es lograr una integración de la AC con los procesos de trabajo. Si consideramos que el principal objetivo de la AC es permitir que el conocimiento fluya hacia donde se requiere cuando se requiere [Borghoff y Pareschi, 1997], entonces un paso inicial para diseñar sistemas de AC para una organización particular, es entender la manera en que el conocimiento fluye a lo largo de los procesos de dicha organización [Nissen y Levitt, 2002]. En este sentido y con el fin de facilitar la identificación y entendimiento de los flujos de conocimiento en los procesos de desarrollo y mantenimiento de software, en este trabajo nos propusimos investigar *de qué*

manera se puede apoyar para facilitar el modelado de flujos de conocimiento en procesos de desarrollo de software, ¿Existe alguna alternativa disponible que permita hacer esto?

Una vez obtenida la respuesta a la pregunta anterior nos interesa saber *si el modelado de flujos de conocimiento en un grupo de mantenimiento de software puede ayudar a definir requerimientos para el diseño de un sistema o estrategia de AC para dicho grupo*. Con el fin de obtener estos requerimientos, otras preguntas que deben responderse son *¿Qué información debe buscarse durante el análisis de los modelos antes mencionados? ¿Qué tipo de información debe estar representada de manera explícita en los modelos para facilitar su análisis?*

Por otro lado, el desarrollo de una estrategia de AC no sólo debe considerar el desarrollo o adquisición de nuevas herramientas. Como ya se dijo, en las organizaciones con frecuencia existen aplicaciones que de una u otra forma apoyan algunas de las fases de la AC. Esto nos lleva a preguntarnos *¿Cómo facilitar la identificación de las herramientas que son usadas dentro del proceso y que pueden ser útiles dentro una estrategia de AC? ¿Cómo identificar el papel que dichas herramientas pueden jugar dentro de dicha estrategia? Una vez identificadas dichas herramientas ¿es posible basar una estrategia de AC alrededor de las mismas o al menos una de ellas, de forma tal que a través de dicha herramienta se pueda integrar la AC en el proceso al que la misma da soporte?*

Finalmente *¿es posible definir un marco metodológico que agrupe todo lo anterior?* Es decir, que considere el análisis de flujos de conocimiento en los procesos de una organización de software, tomando en cuenta las herramientas que apoyan al proceso y que pueden servir como facilitadoras del flujo de conocimiento. Esto con el fin de identificar requerimientos y alternativas de sistemas o estrategias de AC que estén integradas con dichos procesos.

I.4 Metodología de investigación

Con el fin de dar respuesta a las preguntas planteadas, y lograr el objetivo propuesto, se siguió una metodología de trabajo que consistió en una extensa investigación bibliográfica para la identificación de distintos aspectos clave:

- 1) la información que se debe considerar al analizar flujos de conocimiento en procesos organizacionales,
- 2) la forma de modelar procesos organizacionales de tal manera que se resalte el flujo del conocimiento que se da en los mismos, y
- 3) las características que se deben tener en cuenta al momento de analizar herramientas de apoyo a la administración del conocimiento.

Después de la revisión bibliográfica se hicieron propuestas que incluyeron un conjunto de pasos y actividades a tomar en cuenta para analizar procesos desde el punto de vista del flujo de conocimiento, así como técnicas para apoyar en la realización de estos pasos y actividades, como lo fueron dos lenguajes de modelado de procesos con enfoque en el flujo del conocimiento. Dichas propuestas fueron puestas en práctica por medio de casos ejemplo, con el objetivo de analizar, primero, si los lenguajes de modelado propuesto permitían representar explícitamente los elementos principales requeridos para realizar el análisis de flujos de conocimiento, y en segundo lugar, para identificar fallas o faltantes en las técnicas que apoyan los pasos de la etapa de análisis.

De las observaciones obtenidas con los ejemplos, se identificaron posibilidades de mejora que dieron pie a una nueva etapa de investigación bibliográfica. De esta segunda etapa se derivó, principalmente, una nueva propuesta de modelado de procesos para apoyar los pasos requeridos durante la etapa de análisis de flujos de conocimiento, así como el conjunto de características que deben ser tomadas en cuenta a la hora de analizar sistemas o herramientas de AC. Debido a las suposiciones iniciales para la elección de la nueva propuesta de modelado, se realizó un experimento para obtener datos que nos permitieran saber si esas suposiciones podrían ser correctas. Del experimento se derivaron una serie de

observaciones que propiciaron varias modificaciones a la propuesta de modelado. Finalmente, una vez realizadas dichas modificaciones, se aplicó la metodología desarrollada a un caso real con el fin de analizar los posibles beneficios que pueden derivarse de su uso en la práctica. La secuencia de las distintas fases aquí descritas se muestra en la Figura 1, y son explicadas a continuación.

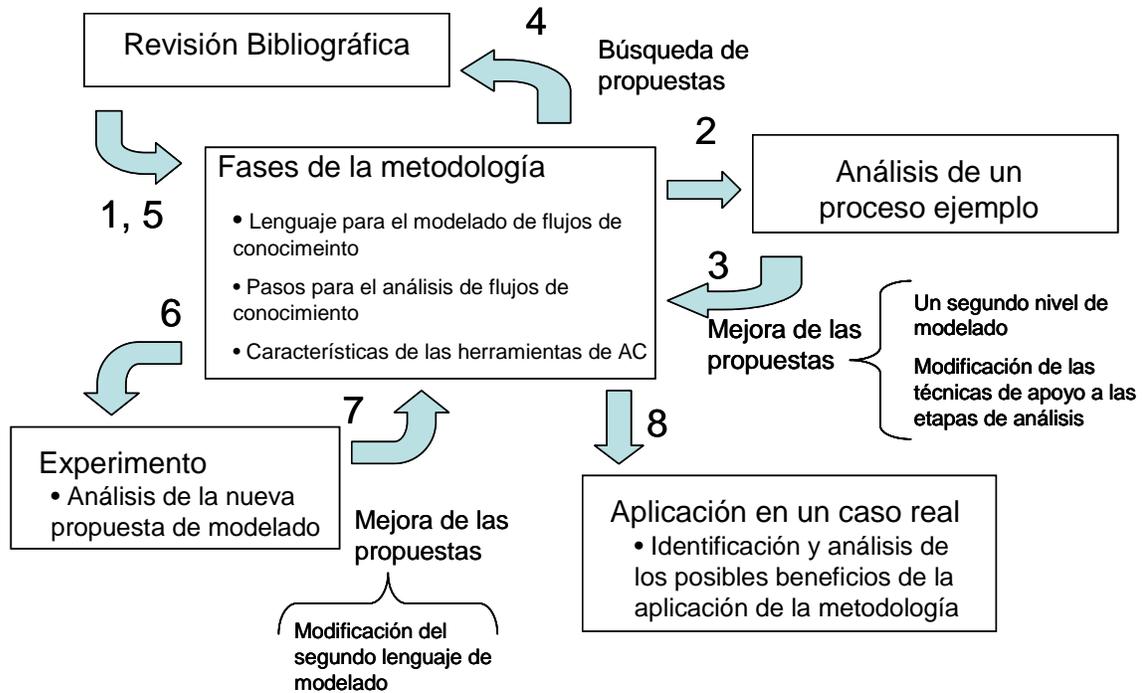


Figura 1. Metodología seguida durante el proceso de investigación.

I.4.1 Revisión bibliográfica

La mayor parte de los fundamentos para el desarrollo de la metodología propuesta en esta tesis tienen una base teórica. Es por esto que para su definición se realizó una extensa revisión bibliográfica. Como se mencionó, dicha revisión se dividió en dos etapas, en la primera se definió una primer propuesta de modelado de procesos con enfoque en el conocimiento y sus fuentes, así como las etapas a seguir durante el análisis de dichos modelos. Esto dio paso a la primera versión de la metodología, en la cual se definieron los principales pasos para el análisis de flujos de conocimiento, la información que debía ser

recabada durante dichos pasos, así como un conjunto de técnicas para apoyar el análisis. Estas propuestas se basaron en los resultados del análisis de bibliografía en varios temas que incluyeron:

- modelos teóricos sobre AC para identificar cuáles son las principales actividades de la AC;
- modelos del flujo del conocimiento para entender cómo fluye el conocimiento en las organizaciones;
- propuestas de sistemas o estrategias de AC, así como modelos para lograr el éxito en las iniciativas de AC, para identificar los principales aspectos que intervienen en dichas iniciativas;
- métodos para el diseño de sistemas y estrategias de AC;
- propuestas para la definición de estrategias o sistemas de AC basados en proceso;
- aplicación de la AC en la ingeniería y el mantenimiento del software, incluyendo sistemas de AC de uso en estos procesos;
- propuestas de modelado de procesos que tomen en cuenta el conocimiento y su flujo.

En la primera etapa de la investigación bibliográfica no se definió la manera de estudiar herramientas o sistemas de apoyo a los procesos desde un enfoque de su intervención en el flujo del conocimiento. Esta necesidad la identificamos posteriormente, por lo que se hizo durante la segunda etapa, en la cual también se definió un nuevo enfoque de modelado que permitiera obtener mayores detalles para facilitar el análisis del proceso. En este punto fue necesario analizar propuestas en aspectos no considerados anteriormente, los cuales incluyeron:

- modelado de procesos en organizaciones de software;
- propuestas para clasificar, comparar o evaluar sistemas de AC;
- teorías y propuestas para la administración de conocimiento distribuido;
- propuestas de sistemas de administración de conocimiento automatizados o autónomos;

- teorías para la clasificación o categorización de sistemas autónomos y niveles de autonomía.

Los temas anteriores se enfocaron principalmente en aspectos concretos relacionados con el nuevo enfoque de modelado propuesto, y con las características que deben ser tomadas en cuenta para estudiar sistemas, herramientas o tecnologías de AC. Finalmente, es importante aclarar que muchas de las propuestas estudiadas surgieron una vez que el proyecto de investigación se encontraba en curso (a partir del 2004). Además, debido al avance constante del área de la AC, con el objetivo de estar al tanto de estos cambios, el proceso de investigación bibliográfica debió continuar a lo largo de todo el proyecto, si bien no con la misma intensidad de las dos fases descritas en esta sección.

I.4.2 Aplicación a casos ejemplo

Con el objetivo de analizar las propuestas que se derivaron de la primera fase de investigación bibliográfica, éstas se llevaron a la práctica por medio del uso de un caso ejemplo. Este caso consistió en una descripción de un proceso de mantenimiento que se había obtenido con anterioridad a este trabajo de tesis. El caso consistió en utilizar el lenguaje de modelado propuesto para realizar un modelo del proceso con base en la descripción mencionada. Posteriormente, se usaron dichos modelos como base para el análisis del proceso siguiendo los pasos definidos por la metodología propuesta. De esta fase del estudio se obtuvieron dos observaciones importantes:

Carencias del lenguaje de modelado propuesto inicialmente. Si bien los modelos generados con el lenguaje utilizado permitieron obtener una visión general de los principales temas y áreas de conocimiento involucradas en el proceso, así como algunas de las principales fuentes, fue difícil identificar detalladamente las relaciones y dependencias entre las fuentes y los temas de conocimiento. Esto dificultó la definición de la estructura de fuentes y tipos de conocimientos, así como su clasificación. Otro factor que se dificultó fue el análisis de la secuencia de actividades a lo largo del proceso. Todo lo anterior hizo difícil el análisis detallado de flujos de conocimiento, por ejemplo de una actividad a otra o

de una fuente a otra. Esta fue la razón por la que decidimos proponer un segundo lenguaje de modelado que apoyara en la realización de un análisis más detallado del proceso, lo cual fue uno de los principales objetivos de la segunda etapa de investigación bibliográfica.

Herramientas que intervienen en el flujo del conocimiento. Al inicio de este trabajo, la identificación de herramientas que intervienen en el flujo del conocimiento no fue tomado en cuenta como un factor importante, sino más bien como algo secundario. Sin embargo, al analizar el caso ejemplo, nos percatamos de que dichas herramientas pueden llegar a ser un factor sumamente relevante, ya sea facilitando u obstaculizando el flujo del conocimiento. Esta observación fue lo que nos llevó a considerar este aspecto como esencial para ser integrado con mayor cuidado en la metodología, de lo que se derivó la necesidad por identificar los elementos a tomar en cuenta al analizar herramientas de AC, aspecto que fue central en la segunda fase de investigación bibliográfica.

I.4.3 Experimento

Durante la segunda fase de investigación bibliográfica, específicamente la realizada para definir la segunda propuesta de modelado de procesos, observamos que podría ser recomendable utilizar un lenguaje dirigido al tipo de proceso a analizar. Esto nos llevó a elegir un lenguaje de modelado de procesos diseñado para procesos de ingeniería de software. La suposición inicial fue que debido a su enfoque, el lenguaje no debería ser difícil de entender, interpretar y analizar por las personas especialistas en el área de la ingeniería del software, particularmente aquellas que se dedican a realizar las actividades del desarrollo y mantenimiento. Lo anterior es importante si consideramos que normalmente se requiere de las personas involucradas en el proceso para que vayan analizándolo y dando su punto de vista sobre el modelo. De esta forma, se facilita definir modelos que realmente reflejen la forma de trabajar de las personas que participan en el proceso. Por lo tanto, comprobar si nuestra suposición era correcta podría resultar un factor importante para reforzar el uso de dicho lenguaje de modelado como parte de la metodología. Con el fin de obtener datos para verificar si la suposición era correcta, se planteó un experimento controlado. Este experimento nos ayudó a obtener algunas

propuestas de mejora, principalmente en la forma de utilizar los elementos incluidos en los modelos para representar el conocimiento y sus fuentes. Estas propuestas fueron posteriormente tomadas en cuenta para la realización de los modelos usados en el caso de estudio final, y permitieron crear modelos más legibles y fáciles de entender.

I.4.4 Aplicación en un caso real

El paso final de este trabajo lo constituyó la realización de un caso de estudio, realizado para verificar que la metodología propuesta cumple con el objetivo de apoyar en la identificación de requerimientos para diseñar sistemas o estrategias de AC enfocadas a mejorar el flujo del conocimiento, considerando el trabajo que realmente desempeñan los involucrados en el proceso, y las herramientas tecnológicas que éstos usan para realizar sus actividades. Por medio de este caso de estudio se demuestra el cumplimiento del objetivo de este trabajo de investigación. Adicionalmente, el caso de estudio sirvió para obtener una idea general de los posibles beneficios que puede aportar la aplicación de la metodología propuesta para el estudio de los procesos de las organizaciones de software. Así mismo, el caso de estudio permitió obtener algunas ideas para mejorar la metodología y las técnicas que la apoyan.

I.4.5 Algunos comentarios finales sobre la evolución de la investigación

La metodología de trabajo anteriormente expuesta presenta una clara división entre las distintas actividades realizadas. Sin embargo, en la realidad esta división no fue tan clara. Además, las interacciones entre las distintas fases tampoco fueron tan suaves y marcadas como se presentan en esta sección. La metodología se ha descrito de esa manera para darle orden al trabajo realizado. Sin embargo, como en toda investigación, existieron eventos a lo largo del proceso que nos hicieron modificar el rumbo en varias ocasiones. Difícilmente un proyecto de investigación sigue un rumbo predefinido desde su inicio hasta su culminación. Ya lo dijo Albert Einstein alguna vez: *“Si hubiéramos sabido qué era lo que estábamos haciendo, no se le habría llamado investigación”*, o Wernher von Braunⁱⁱⁱ con otras palabras: *“investigación es lo que estoy haciendo cuando no se qué estoy*

haciendo". Esta sección describe dos sucesos principales que influyeron en la definición del rumbo final que tomó esta investigación.

Primeramente es preciso señalar que en un inicio, el trabajo de investigación se enfocó principalmente en el desarrollo de sistemas de AC para procesos de mantenimiento de software que tuvieran un cierto grado de automatización en el manejo del conocimiento. Afortunadamente, desde una etapa temprana nos percatamos de que para lograr lo anterior, primero era necesario saber exactamente para qué podría requerirse dicho sistema. Esto nos llevó a darnos cuenta de que antes de proponer soluciones, era necesario conocer las necesidades de conocimiento de los grupos encargados del mantenimiento. Debido a la carencia de un enfoque que nos sirviera para realizar dicho tipo de estudios se propuso el desarrollo de una metodología que estudiara los procesos con un enfoque de administración del conocimiento. Fue en este punto que comenzamos a encontrar en la literatura que esto no era sólo una observación nuestra, sino una necesidad que comenzaba a surgir en varias áreas. Como un dato que refuerza la importancia de esta observación podemos decir que el proyecto de investigación inició en 2003, la decisión de enfocarnos al desarrollo de la metodología se tomó en 2004, las propuestas que conocemos que sigan el enfoque de la AC basada en procesos de trabajo se comenzaron a publicar a partir de 2002 (ver [Maier y Remus, 2002; Woitsch y Karagiannis, 2002]), siendo el año 2004 cuando se empieza a dar mayor relevancia al estudio de las necesidades de conocimiento de las personas que desempeñan los procesos de trabajo (ver [Dalkir, 2005: p. 21; Scholl et al., 2004; Stenmark y Lindgren, 2004; Wiig, 2004]), éstos trabajos nos confirmaron que nuestra decisión no estaba errada.

El segundo evento fue después de la etapa comprendida durante el caso de estudio ejemplo, cuando observamos la necesidad de dar más importancia a las herramientas que usan las personas que desempeñan el proceso, como apoyo para sus actividades. Esto nos llevó a decidir darle más importancia a la identificación de dichas herramientas, de forma tal que pudieran ser tomadas en cuenta como parte de los sistemas o estrategias de AC derivadas del estudio de los procesos. Este evento sucedió durante el año 2005. Al realizar la segunda fase de investigación bibliográfica, no pudimos encontrar algo parecido a lo que

proponíamos, aún cuando ya hacía tiempo que había autores que sostenían la necesidad de considerar la infraestructura tecnológica de las organizaciones como un factor importante para lograr el éxito de las iniciativas de AC, sobre todo en empresas pequeñas y medianas [Sparrow, 2001; Wong, 2005]. Más importante aun, el integrar la AC en las herramientas de trabajo diario está empezándose a ver como la manera de integrar la AC en el trabajo cotidiano de las empresas [Davenport, 2007], esto último ha sido definido como el principal reto de la AC en el presente y futuro cercano [Scholl et al., 2004]. Lo anterior ha reforzado nuestra idea de aumentar la importancia de considerar las herramientas de trabajo como una parte fundamental del estudio de un proceso desde el punto de vista de la AC, y nos ha dado razones para pensar que este último cambio en el rumbo de la investigación fue acertado.

I.5 Contribuciones principales de esta tesis

Las siguientes son las principales contribuciones de esta tesis:

- *Dos propuestas de modelado de procesos con enfoque en el modelado del conocimiento, sus fuentes y su flujo.* Durante la realización de la tesis se observó la necesidad de dos niveles de modelado para el proceso, uno inicial más enfocado a la captura del proceso, y uno final para el modelado detallado del mismo y más enfocado al análisis del proceso y con una orientación más centrada en el tipo de proceso a analizar, de lo que se obtuvieron:
 - Un enfoque de modelado de procesos con orientación al modelado del conocimiento y sus fuentes en procesos organizacionales en general, basado en un lenguaje de modelado ampliamente usado y que por su flexibilidad y generalidad puede ser aplicado en diversos contextos.
 - Un enfoque de modelado de procesos con orientación al modelado del conocimiento, sus fuentes y su flujo, basado en un lenguaje de modelado de procesos formalmente definido y restringido que está dirigido especialmente para organizaciones de desarrollo de software.

- *Una metodología para el análisis de flujos de conocimiento*, la cual define lineamientos para ayudar a centrarse en la principal información que debe ser buscada durante el análisis de flujos de conocimiento en procesos organizacionales en lo general, así como un conjunto de técnicas que apoyan cada uno de los pasos propuestos por la metodología.
- *Un marco de trabajo para el análisis de herramientas desde el punto de vista de su intervención en el flujo del conocimiento de los procesos donde son usadas*. Para dar respuesta a algunas de las preguntas de investigación planteadas, y apoyar el cumplimiento del objetivo de esta tesis, se identificaron y definieron un conjunto de lineamientos para ser tomados en cuenta con el fin de analizar herramientas de trabajo que puedan estar interviniendo, ya sea de forma positiva o negativa, en el flujo del conocimiento dentro de una organización.
- *Un ejemplo del uso de la metodología en un caso real*. Se realizó un caso de estudio donde la metodología desarrollada en esta tesis fue aplicada con el fin de verificar su utilidad, y analizar sus fuerzas y debilidades, con el objetivo de establecer lineamientos para su mejora, así como de observar y mostrar los posibles beneficios que pueden surgir de su aplicación.
- *Uso de la metodología para apoyar otros proyectos de investigación*. Como aportación adicional, la metodología ha sido aplicada en un caso distinto al de esta tesis para realizar una investigación en un contexto independiente al nuestro. Esto se describe con mayor detalle en el Capítulo VI.

I.6 Contenido del documento

El resto de este documento está dividido en seis capítulos y cinco apéndices, el contenido de cada uno de ellos se compone de la siguiente manera:

El **Capítulo II** (Marco teórico) presenta una introducción a las distintas áreas que comprenden el marco teórico en el que se basa esta tesis. En particular, se da una introducción al área de la AC, haciendo énfasis en la AC con enfoque en procesos. Posteriormente se describen trabajos relacionados con la aplicación de la AC en procesos u

organizaciones de ingeniería de software. Para finalizar con la identificación de los requerimientos necesarios para estudiar el proceso de mantenimiento de software desde el punto de vista de la AC, donde se observa la necesidad de que dicho enfoque esté centrado en los flujos de conocimiento dentro del proceso.

En el **Capítulo III** (KoFI: una metodología para el estudio de flujos de conocimiento) se detalla la metodología propuesta para apoyar en el estudio de flujos de conocimiento en procesos organizacionales. Específicamente, se describen las fases y etapas que componen la metodología, así como las técnicas propuestas para la realización de cada etapa.

El **Capítulo IV** (Modelado de procesos con enfoque en el flujo de conocimiento) presenta los enfoques de modelado de procesos que se proponen para apoyar a la metodología. El capítulo consiste en un análisis de algunos trabajos existentes en la literatura, de los cuales se deriva la propuesta de dos niveles de modelado, uno inicial para la captura del proceso que consiste en un enfoque genérico y flexible, y otro para el análisis detallado del proceso, consistente en un enfoque más formalmente definido y restringido.

El **Capítulo V** (MAHAC: Un marco de trabajo para el análisis de herramientas de apoyo a la administración del conocimiento) describe un marco de trabajo definido para apoyar durante la metodología en el análisis de herramientas de apoyo a los procesos, desde el punto de vista del papel que dichas herramientas pueden jugar como sistemas de AC.

El **Capítulo VI** (Resultados: aplicación y utilidad de la metodología) presenta los resultados de la tesis. Estos resultados están divididos en dos aspectos, el primero es un experimento que se definió para evaluar preliminarmente el lenguaje de modelado de procesos propuesto para la etapa de análisis de la metodología. El segundo aspecto consiste en la aplicación de la metodología a un caso real con el fin de mostrar su aplicabilidad, y de verificar que cumple su propósito, es decir, ayudar a definir estrategias o sistemas de AC que consideren las actividades que realizan los involucrados en el proceso, y las herramientas que éstos últimos utilizan para realizarlas.

El **Capítulo VII** (Conclusiones, aportaciones y trabajo futuro) presenta las conclusiones y reflexiones derivadas de este trabajo; las principales aportaciones, entre las que se encuentran las publicaciones derivadas de esta tesis; así como algunas oportunidades de trabajo futuro que surgieron durante el desarrollo de la misma.

El **Apéndice A** (Experimento de evaluación de las extensiones a SPEM) presenta un reporte detallado del experimento que se realizó para evaluar el lenguaje de modelado propuesto para la etapa de análisis de la metodología.

El **Apéndice B** (Caso de estudio) detalla el caso de estudio realizado para mostrar la aplicabilidad de la metodología en un caso real, y verificar que apoya en la definición de estrategias o sistemas de AC que consideren los procesos de trabajo diario y las herramientas que los soportan.

El **Apéndice C** (Cuestionario) presenta un cuestionario que fue aplicado a una persona que aplicó la metodología propuesta en esta tesis con el fin de diseñar un portal de conocimientos para una empresa del ramo industrial. Este cuestionario da una idea de la utilidad, fuerzas y debilidades de la metodología, desde el punto de vista de alguien ajeno a su desarrollo.

El **Apéndice D** (Modelo del uso de la metodología) propone un modelo general para el uso de la metodología siguiendo un enfoque visual.

El **Apéndice E** (Ontologías de fuentes y tipos de conocimiento) muestra un ejemplo de una ontología que fue definida con base en el caso de estudio realizado para esta tesis.

ⁱ William Henry (1774-1836) fue un químico inglés.

ⁱⁱ Albert Einstein (1878-1955) premio nobel de física en 1921.

ⁱⁱⁱ Wernher von Braun (1912-1977) fue uno de los líderes en el desarrollo de las tecnologías para cohetes en Alemania y Estados Unidos.

CAPÍTULO II

Marco teórico

“Un poco de conocimiento en acción es infinitamente más valioso que mucho conocimiento inactivo” Kahlil Gibran (1883-1931)^{iv}

En este trabajo se presenta una propuesta de una metodología para apoyar en la identificación, modelado y representación del conocimiento, su flujo dentro de procesos de software, con el fin de apoyar en el diseño de sistemas o estrategias de administración del conocimiento. La metodología propuesta en esta tesis está enfocada a facilitar el entendimiento de los flujos de conocimiento dentro de procesos altamente basados en conocimiento, con el fin de identificar oportunidades para la mejora de dichos flujos. Si bien la metodología ha sido definida para procesos genéricos, se ha hecho énfasis en su aplicación en procesos de ingeniería de software. En este sentido, el trabajo está fundamentado en aspectos teóricos de distintas áreas, que incluyen: administración del conocimiento, ingeniería de procesos y en particular su integración con la administración del conocimiento, y la aplicación de la administración del conocimiento en procesos de ingeniería de software, con especial énfasis en la etapa de mantenimiento. En este capítulo se presenta una introducción a éstos temas.

II.1 Administración del conocimiento

En la actualidad no existe un consenso sobre lo que significa el concepto de administración del conocimiento (AC) [Serenko y Bontis, 2004]. Debido a la naturaleza multidisciplinar de la AC [Dalkir, 2005: p. 6], ésta ha sido considerada desde muy diversos puntos de vista. Dalkir [2005: p. 4] identifica tres enfoques principales: 1) desde el punto de vista de las actividades del negocio, donde es considerada como parte de las estrategias, políticas y prácticas del negocio; 2) desde un enfoque de la ciencia cognoscitiva o ciencia del conocimiento, donde el conocimiento es visto como un recurso; y 3) desde el enfoque

de tecnologías y procesos, donde es vista como un conjunto de actividades y sistemas tecnológicos que apoyan esas actividades, por ejemplo, en el manejo del uso de la información, con el fin de facilitar el flujo del conocimiento hacia las personas que lo requieren, cuando lo requieren.

Desde el punto de vista de la tercera perspectiva, la AC puede ser definida como una disciplina sistemática, y un conjunto de propuestas que apoyan en el crecimiento y flujo del conocimiento [Rao, 2005b: p. 3]. Dado que el enfoque de este trabajo es dentro de esta perspectiva, entenderemos la administración del conocimiento como *una disciplina que provee de métodos, técnicas y herramientas encaminadas a apoyar las actividades necesarias para facilitar el flujo del conocimiento hacia donde se requiere, con el fin de realizar de mejor manera el trabajo necesario para completar procesos específicos dentro de una organización*. En este sentido, existen tres conceptos básicos que es necesario definir: ¿qué se entenderá como conocimiento?, ¿qué constituye el flujo del conocimiento?, y ¿cuáles son las actividades involucradas en el flujo del conocimiento, es decir, qué actividades constituyen el proceso de administración del conocimiento?

II.1.1 Definición de conocimiento

Davenport y Prusak [2000:p. 1] consideran que para entender el concepto de conocimiento es necesario entender los conceptos de datos e información; idea compartida por otros autores [Dalkir, 2005:p. 7; Tiwana, 2000:p. 56]. Estos tres conceptos tienen relaciones entre ellos que pueden ser usadas para considerarlos como una jerarquía. Más aun, hacer una distinción clara entre datos, información y conocimiento no siempre es posible en la práctica, por lo que puede ser más fácil verlos como un flujo continuo [Davenport y Prusak, 1997:p. 9]. Sin embargo, existen elementos que distinguen a estos tres conceptos.

- **Datos.** Los datos pueden ser vistos como los átomos de la información, es decir, hechos simples que pueden agruparse para crear información, pero que por si solos carecen de significado [Davenport y Prusak, 2000:p. 2].

- **Información.** Davenport y Prusak [2000:p. 3] sostienen que la información son datos qué hacen una diferencia. Puede verse como un conjunto de datos que tienen un significado, y cuyo objetivo es informar a quien los recibe, con el fin de tener un efecto en su juicio o en la forma en que percibe algo. Desde esta perspectiva, quien decide qué datos son realmente información, es la persona que determina su significado y pertinencia dependiendo de sus propios intereses o información previa.
- **Conocimiento.** El conocimiento es información con valor agregado [Davenport y Prusak, 1997:p. 9], es decir, cuando la persona que obtiene esa información le da más valor, en el sentido de que agrega algo, por ejemplo, la interpreta, la pone en algún contexto, o agrega algún significado tomando su propia experiencia, intereses, ideas, etc. Davenport y Prusak [2000:p. 5] definen el conocimiento como *“una mezcla fluida de experiencias enmarcadas, valores, información contextual, y pericia que provee de un marco para evaluar e incorporar nuevas experiencias e información. Esto se origina y es aplicado en la mente de quienes conocen. En las organizaciones, con frecuencia viene embebido no sólo en documentos o repositorios, sino también en rutinas, procesos, prácticas o normas de la organización”*.

El modelo jerárquico Datos-Información-Conocimiento (D-I-C) propone que existe un flujo unidireccional, en el cual el conocimiento es creado a partir de información compuesta de un conjunto de datos. Sin embargo, el proceso inverso también existe. Tuomi [1999] propone una jerarquía inversa argumentando que los datos existen como una solución a un problema práctico, donde el conocimiento se descontextualiza y estructura de acuerdo a una semántica definida en entradas de una base de datos, independientes y aisladas. En este proceso, el conocimiento se articula en un contexto conceptual, lo que permite verbalizarlo o textualizarlo para generar información. Posteriormente, los significados de esta información deben ser representados y estructurados de tal forma que se permita su manejo automático, para lo que esta información debe ser estructurada en “átomos”, los cuales no tienen significado por sí solos, es decir, que se generan datos. Analizando las dos representaciones de la jerarquía, podemos definir un modelo que las agrupe como un proceso cíclico, como se muestra en la Figura 2.

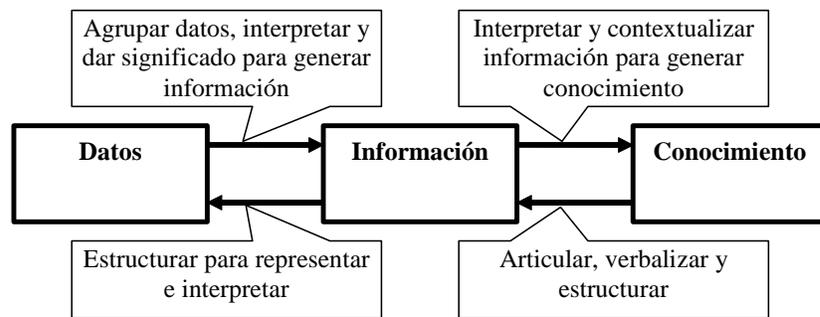


Figura 2. Modelo jerárquico del conocimiento.

No todo el conocimiento puede ser fácilmente representado como información y/o datos, ya que, si bien existe conocimiento representado en forma de documentos, registros de una base de datos, etc., existe otro que se encuentra en la mente de las personas. Es decir que el conocimiento puede ser explícito o tácito [Nonaka y Takeuchi, 1995]. El *conocimiento explícito* es el que está expresado de una manera formal (palabras, números, etc.), y por lo tanto puede ser fácilmente comunicado y transferido (datos, documentos, etc.); mientras que el *conocimiento tácito* es el conocimiento personal utilizado para realizar las tareas y para obtener un entendimiento del medio ambiente; por ejemplo, la experiencia y las habilidades individuales de cada persona. Por lo general este tipo de conocimiento es difícil de articular de una manera formal, por lo que resulta difícil de transferir.

Tomando en cuenta cómo el conocimiento tácito se almacena en la mente de las personas, éste puede ser separado en dos dimensiones: técnica (saber cómo) y cognoscitiva (saber qué, dónde, cuándo, por qué) [Choo, 1999; Robillard, 1999]:

- 1) *La dimensión técnica, o conocimiento procedural*, es dinámica y tiene que ver con la pericia para llevar a cabo una tarea; puede verse como la experiencia o habilidad para realizar alguna actividad.
- 2) *La dimensión cognoscitiva*, es estática y se refiere a esquemas, modelos mentales, creencias y percepciones sobre conceptos (objetos, personas o eventos), sus propiedades y las relaciones entre ellos. Hay autores que definen distintos tipos de

conocimiento que pueden ser agrupados en la dimensión cognoscitiva, tales como: declarativo (saber qué), causal (saber por qué), condicional (saber cuándo) y relacional (conocer las relaciones entre los conceptos de conocimiento) [Alavi y Leidner, 2001].

En la dimensión cognoscitiva, el conocimiento puede ser dividido en dos formas:

- a) *Tópico o semántico*. Se refiere al significado de las palabras, como definiciones de un diccionario o libro de texto. La memoria tónica está hecha de todas las estructuras culturales de un ambiente y da soporte a la organización del conocimiento sobre ese ambiente. Estos ambientes existen a varios niveles, incluyendo el social, personal, profesional y técnico [Robillard, 1999].
- b) *Episódico*. Consiste en la experiencia de uno mismo con el conocimiento [Robillard, 1999]. Es conocimiento que se aprende por medio de la experiencia en la aplicación del conocimiento tónico y técnico o procedural.

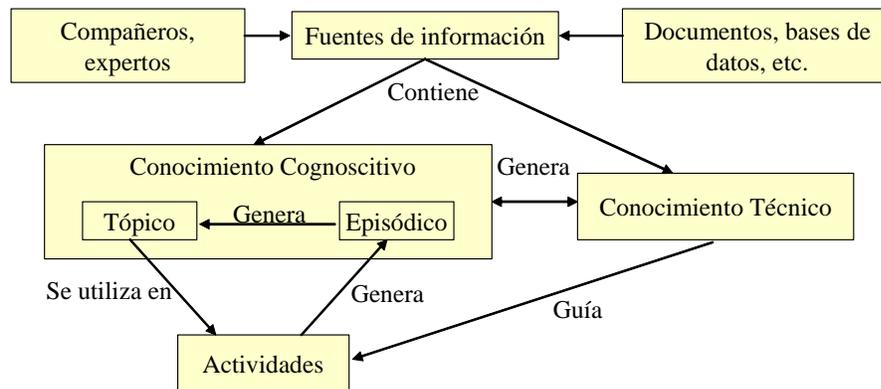


Figura 3. Modelo conceptual de las relaciones entre los tipos de conocimiento.

Como se muestra en el modelo de la Figura 3, las fuentes de información o conocimiento (explícito y tácito) pueden ser utilizadas para obtener conocimiento cognoscitivo y técnico. Así también, el conocimiento tónico es utilizado en la realización de actividades, por ejemplo, para saber qué debemos hacer, qué datos o información requerimos, cuándo se debe hacer, dónde, etc. La realización de estas actividades es guiada por el conocimiento técnico, con el cual se determina cómo llevar a cabo esa actividad. La aplicación del conocimiento tónico en la realización de actividades, genera a su vez

conocimiento episódico o experiencias, que a la larga se convierte en pericia o conocimiento técnico. Finalmente, las experiencias obtenidas en la realización de actividades pueden también incrementar nuestro conocimiento tópico, por ejemplo, identificando qué información es más adecuada para realizar una determinada tarea de una forma más rápida o eficiente.

Es necesario considerar las distintas dimensiones y tipos de conocimiento a la hora de identificar y entender la manera en que el conocimiento fluye dentro de una organización; por ejemplo, la forma en que es almacenado o estructurado en información y datos; la manera en que estos datos son luego recuperados para generar información y conocimiento; la forma en que son manejadas las fuentes de conocimiento (o conocimiento explícito); la manera en que se transmite el conocimiento tácito, etc. La siguiente sección presenta algunas propuestas de modelos que tratan de definir la forma en que el conocimiento fluye dentro de las organizaciones, de lo cual se establece un marco conceptual para analizar y entender el flujo del conocimiento tanto dentro de un contexto del individuo, como el de un grupo o una organización.

II.1.2 El flujo del conocimiento

Desde un punto de vista organizacional, la importancia del conocimiento radica en su aplicación [Alavi y Tiwana, 2002], por lo que el principal objetivo de una estrategia o sistema de AC debe enfocarse en facilitar y mejorar el flujo del conocimiento [Borghoff y Pareschi, 1998] hacia donde se requiere. Por lo tanto, un sistema de AC debe apoyar una o más de las actividades requeridas para lograr este flujo, por lo que es importante identificar cuales son estas actividades o procesos involucrados. A continuación se describen diversos modelos que han sido propuestos para explicar el flujo del conocimiento en las organizaciones, de los cuales partimos para definir un modelo propio que integra varias de las ideas propuestas por la literatura.

II.1.3 Modelos de flujo del conocimiento

En la literatura es posible identificar tres enfoques principales con respecto al flujo del conocimiento: el primero considera la forma en que el conocimiento pasa entre sus estados tácito y explícito, conocido como el proceso de creación de conocimiento. El segundo se enfoca en ver el flujo del conocimiento dentro de un contexto organizacional, considerando el conocimiento como un recurso que nos da la capacidad de utilizar información con un fin práctico, como la toma de decisiones. El tercero se refiere a las actividades que componen el proceso de AC. A continuación se describen dos modelos relacionados con los dos primeros enfoques. Posteriormente se abordan diversas propuestas de modelos de procesos de AC.

II.1.3.1 El proceso de creación del conocimiento

El proceso de creación del conocimiento de [Nonaka y Takeuchi, 1995] (o modelo SECI por las siglas de los mecanismos que propone), considera dos dimensiones para la transferencia y creación del conocimiento, la epistemológica y la ontológica. La dimensión epistemológica define mecanismos por medio de los cuales el conocimiento tácito se vuelve explícito y viceversa. La combinación de estas dos categorías del conocimiento hace posible cuatro mecanismos o patrones de conversión del conocimiento: socialización, exteriorización, combinación e interiorización. La Figura 4 muestra el funcionamiento de estos mecanismos.

La **socialización** es el proceso por medio del cual se comparte conocimiento tácito entre individuos [Nonaka y Konno, 1998]. A través de este proceso los individuos transmiten entre ellos sus experiencias.

La **exteriorización** involucra la expresión del conocimiento tácito y su translación hacia formas comprensibles que pueden ser entendidas por otros [Nonaka y Konno, 1998]. Este proceso permite convertir el conocimiento tácito en conocimiento explícito, al plasmarlo en medios formales, en forma de metáforas, analogías, modelos o anécdotas, etc. [Choo, 1999], a través de palabras, gráficos, imágenes, video, audio, etc.

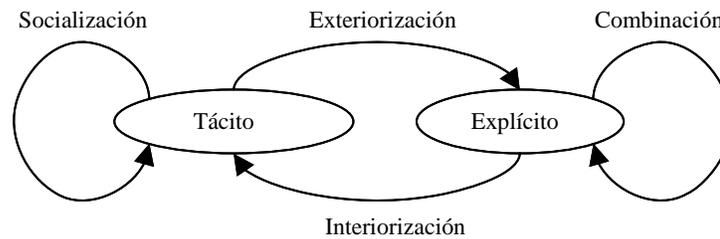


Figura 4. Mecanismos de conversión del conocimiento

La **combinación** involucra la conversión de conocimiento explícito en conjuntos más complejos del mismo [Nonaka y Konno, 1998]. Mediante este mecanismo es posible generar conocimiento explícito consultando fuentes formales de conocimiento, y combinándolas o reconfigurándolas para producir nuevo conocimiento explícito [Choo, 1999].

La **interiorización** es el mecanismo por medio del cual el conocimiento explícito se vuelve tácito [Nonaka y Konno, 1998]. Es el proceso de aprender y socializar al hacer reiteradamente una tarea de modo que el conocimiento explícito de los principios y procedimientos aplicados llegue a ser absorbido como el conocimiento tácito del estilo y el hábito del individuo [Choo, 1999]. La interiorización permite a los individuos incrementar su experiencia mediante la consulta reiterada de conocimiento explícito en la realización de sus actividades.

Por otra parte, la dimensión ontológica está relacionada con el conocimiento que es compartido desde un punto de vista del individuo, los grupos y la organización. En este esquema, la socialización permite la transferencia de conocimiento tácito entre individuos, la exteriorización del conocimiento tácito a un grupo, la combinación entre conocimiento explícito de distintos grupos, y la interiorización permite generar conocimiento tácito del conocimiento explícito existente en una organización o un grupo.

Como se muestra en el modelo de la Figura 5, los procesos de exteriorización e interiorización del proceso de creación de conocimiento, se empatan con el modelo D-I-C.

Dado que el modelo D-I-C involucra la formalización del conocimiento al convertirlo en información y datos para su almacenamiento y manejo, y la interiorización al recuperar estos datos e información para aplicarlos en la solución de algún problema.

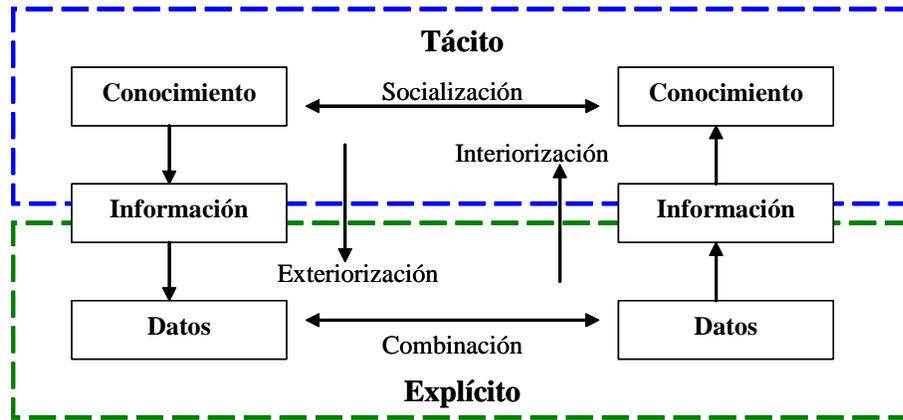


Figura 5. Integración de la jerarquía D-I-C con el proceso de creación de conocimiento.

II.1.3.2 El ciclo del conocimiento

Choo [1999] propone que existen tres procesos estratégicos de información en las organizaciones que se combinan en un ciclo continuo de aprendizaje y adaptación llamado el ciclo del conocimiento (Figura 6). Estos procesos son: percepción, creación de conocimiento y toma de decisiones.

En este ciclo de conocimiento, la percepción sirve para realizar una representación del medio ambiente; esta percepción se obtiene mediante el uso de la información y los flujos de experiencias. Esto lleva a la creación de significados compartidos o modelos mentales que son utilizados para planear y tomar decisiones. El proceso de creación de conocimiento ayuda también en la toma de decisiones, así como a generar nuevos significados y experiencias que pueden ser compartidos. Finalmente, el proceso de toma de decisiones genera a su vez nuevo conocimiento que puede ser capitalizado por los individuos, así como compartido con el resto de los miembros de la organización.

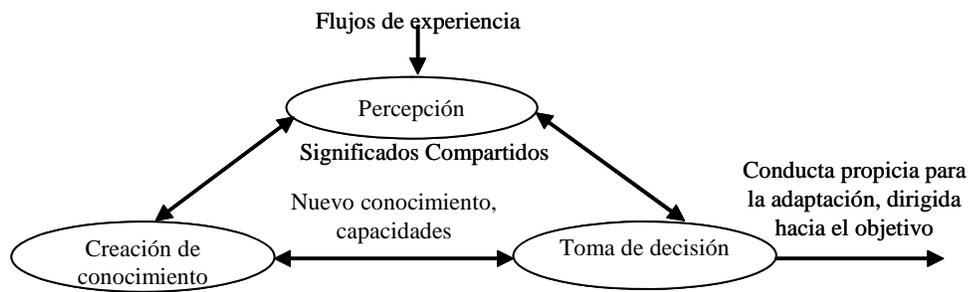


Figura 6. El ciclo del conocimiento [Choo, 1999].

El modelo del ciclo del conocimiento da una idea de la relación que existe entre el conocimiento de una organización y la toma de decisiones, donde la información existente en la organización es utilizada en procesos mentales orientados a obtener conocimiento que posteriormente permita tomar decisiones. Desde esta perspectiva, el conocimiento puede verse como un recurso que nos da la capacidad de usar información y aplicarla en la solución de un problema, realización de alguna tarea o toma de alguna decisión [Carlsson, 2002].

Con base en este enfoque centrado en la aplicación del conocimiento, podemos observar que cuando un trabajador requiere emprender alguna acción, por ejemplo resolver un problema, utiliza su propio conocimiento así como las fuentes de información que puedan ser relevantes. Esta información puede ser útil para obtener el conocimiento que dé los elementos necesarios para decidir qué solución dar al problema en cuestión. Para esto, el trabajador se apoya en los mecanismos de conversión del conocimiento (socialización e interiorización). Así también, durante los procesos de análisis de información, o al aplicar el conocimiento en la toma de decisiones o la realización de alguna acción, el trabajador genera nuevo conocimiento que posteriormente puede ser compartido, lo cual también se hace utilizando los mecanismos de conversión del conocimiento (exteriorización, combinación, socialización). Por tanto, el modelo SECI se encuentra embebido en este ciclo del conocimiento, como se ilustra en la Figura 7.

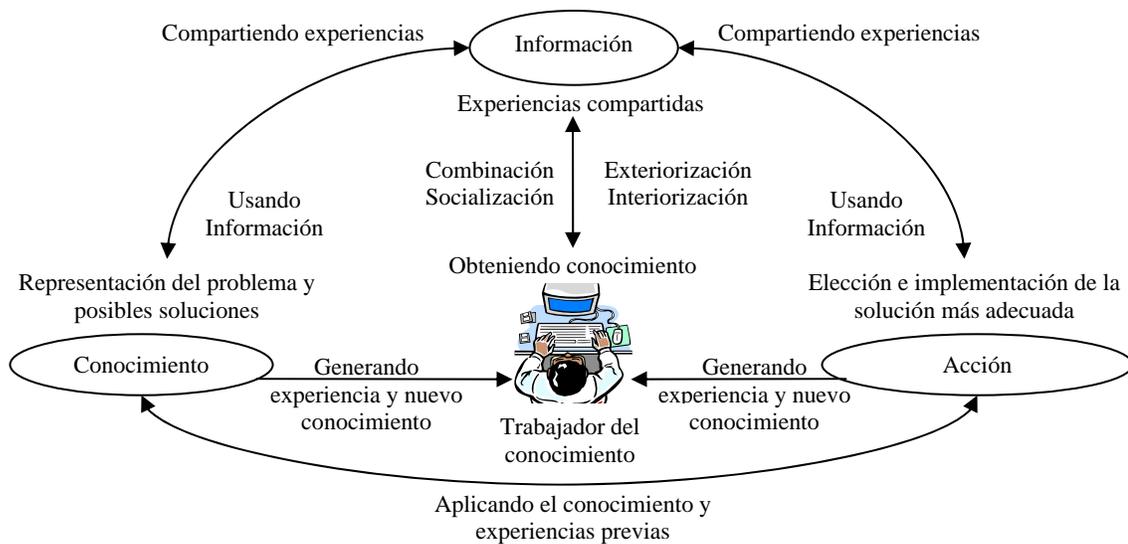


Figura 7. Extensión del modelo del ciclo del conocimiento de Choo [1999].

El modelo de la Figura 7 sugiere que las etapas de creación y aplicación de conocimiento no son procesos aislados uno de otro, sino que se encuentran interrelacionados, ya que la aplicación del conocimiento permite también incrementar el existente. Asimismo, la generación de nuevo conocimiento se da debido a la necesidad de aplicar éste a la solución de algún problema dado. Esto hace resaltar la importancia que tiene la etapa de aplicación dentro del flujo del conocimiento, ya que mientras el conocimiento sea aplicable, seguirá siendo utilizado. Por lo tanto, podemos decir que la aplicación del conocimiento es la etapa que define su ciclo de vida.

II.1.4 El proceso de administración del conocimiento

En la literatura existe una amplia variedad de propuestas de modelos de procesos de administración del conocimiento o del ciclo de vida del conocimiento. Algunos de ellos han sido analizados y comparados con otros en [Dalkir, 2005; Holsapple y Joshi, 1999; Nissen, 2002; Qureshi *et al.*, 2004]. Las propuestas de actividades que constituyen un proceso de AC difieren en varios aspectos, entre los que se encuentra, el enfoque, el proceso seguido para su definición, el tipo de recurso de conocimiento que se quiere manejar, y el objetivo que se pretende lograr con la implementación de dicho proceso [Holsapple y Joshi, 1999].

Peachey [2005] ha identificado que el principal enfoque en las investigaciones en AC publicadas en revistas relacionadas con los sistemas de información, es hacia la transferencia de conocimiento, siguiendo el almacenamiento/recuperación, después la aplicación y finalmente la creación de conocimiento. Para este estudio, Peachey se basa en el modelo de proceso de AC de Alavi y Leidner [2001], el cual sólo considera las cuatro actividades mencionadas.

La Tabla I presenta algunos de los modelos de actividades de AC que han sido propuestos, donde se ven las diferencias con respecto a las etapas que propone cada uno. Algunos consideran mayor detalle en la distribución de las distintas etapas que componen el proceso. Sin embargo, existen similitudes que pueden ser empleadas para definir un proceso que busque considerar los distintos enfoques propuestos. Tomando en cuenta el análisis de los diferentes modelos, hemos definido un proceso formado por siete etapas mostradas en la Tabla I.

En el proceso propuesto se ha buscado considerar las distintas etapas del resto de los modelos, algunas actividades de algunos modelos las hemos agrupado en una sola etapa, mientras que otras las hemos separado en dos o más. Sin embargo, las etapas principales han sido mantenidas. Por ejemplo, actividades como organización, codificación y captura de conocimiento las hemos agrupado dentro de la etapa de formalización. Esto es debido a que la mayoría de los sistemas de AC se enfocan en el manejo de conocimiento explícito, y comúnmente consideran aspectos relacionados con la codificación, organización y captura del mismo [Borghoff y Pareschi, 1998]. Por otra parte, se ha considerado la etapa de filtrado e interpretación, la cual no se encuentra de manera explícita en el resto de los modelos. Esto es debido a que dar soporte a esta actividad es uno de los principales retos de la AC [Marwick, 2001], por lo que hemos decidido tomarla como una etapa separada para que pueda ser estudiada y entendida a mayor detalle. Finalmente, algunos modelos proponen una etapa de actualización o evolución del conocimiento. En nuestro caso hemos decidido no agregarla dado que creemos que esta etapa puede ser tomada como el resultado de la transición entre las etapas de aplicación y creación de conocimiento, ya que al aplicar el conocimiento, se crea o se adquiere nuevo conocimiento, o se incrementa o complementa

el conocimiento existente, por lo que se puede considerar que la evolución se da dentro del mismo proceso de creación de conocimiento.

Tabla I. Modelos del proceso de administración del conocimiento.

Modelo	Etapa 1	Etapa 2	Etapa 3	Etapa 4	Etapa 5	Etapa 6	Etapa 7
[Wiig, 1993]	Construir	Capturar	Transferir	Aplicar			
[Davenport y Prusak, 2000]	Adquirir	Codificar/ Coordinar	Transferir				
[Tiwana, 2000]	Adquirir	Compartir	Utilizar				
[McElroy, 2000]	Producir	Integrar	Difundir	Aplicar			
[Alavi y Leidner, 2001]	Crear	Almacenar/ Recuperar	Transferir	Aplicar			
[Meehan y Richardson, 2002]	Crear	Almacenar	Compartir	Aprovechar			
[Rus y Lindvall, 2002]	Originar/ Crear	Capturar/ Adquirir	Transformar / Organizar	Distribuir/ Accesar	Aplicar		
[Edwards, 2003]	Crear/ Adquirir	Retener	Compartir/ Transferir	Usar	Refinar/ Actualizar		
[Kuczaj et al., 2001]	Crear	Coleccionar/ Almacenar	Identificar necesidad	Compartir	Actualizar		
[Qureshi et al., 2004]	Crear	Coleccionar	Organizar	Comunicar	Usar		
[Nissen, 2002]	Crear	Organizar	Formalizar	Distribuir	Aplicar	Evolucionar	
[Dalkir, 2005]	Capturar/ Crear	Valorar	Compartir/ Diseminar	Contextualizar	Adquirir/ Aplicar	Actualizar	
[Ward y Aurum, 2004]	Crear	Adquirir	Identificar	Adaptar	Organizar	Distribuir	Aplicar
Proceso propuesto	Crear/ Adquirir	Externalizar	Almacenar	Transferir/ Compartir	Internalizar Recuperar Interpretar/ Filtrar		Aplicar

Un aspecto común en los modelos de proceso de AC, es que las etapas de la AC son vistas como actividades independientes y aisladas unas de otras. Holsapple y Joshi [1999] consideran que para tener un entendimiento común de la AC, es necesario considerar las interrelaciones de las distintas actividades que la constituyen, y no ver éstas como actividades aisladas. En la siguiente sección se hace un ejercicio de análisis para integrar los distintos modelos presentados anteriormente, en un mismo modelo cíclico basado en las actividades de AC propuestas.

II.1.5 Modelo integrado

Desde el punto de vista de una organización, el valor del conocimiento radica en su aplicación, más que en el conocimiento por sí solo [Grant, 1996], por lo que la aplicación del conocimiento se vuelve la etapa más importante dentro del proceso de AC [Alavi y Tiwana, 2002; Rus y Lindvall, 2002]. Desde este enfoque, la AC debe ser conducida para facilitar el flujo del conocimiento hacia donde se requiere su aplicación [Alavi y Tiwana, 2002; Borghoff y Pareschi, 1997]. Para lograr lo anterior se necesitan las distintas actividades que constituyen el proceso de AC.

La Figura 8 presenta un modelo que integra las distintas actividades como un flujo centrado en la aplicación y la transferencia del conocimiento. En el modelo la transferencia del conocimiento es vista desde dos enfoques, la transferencia de conocimiento tácito, y la transferencia de conocimiento explícito. En el primer caso se hace uso principalmente de la socialización, mientras que en el segundo, la transferencia es habilitada por un conjunto de actividades encaminadas a facilitar la formalización del conocimiento, su almacenamiento y posterior recuperación. Así mismo, es posible observar en el modelo de la Figura 8, la interrelación de las distintas etapas del proceso de AC, con los distintos mecanismos de creación del conocimiento definidos por el modelo SECI [Nonaka y Takeuchi, 1995].

Equiparando el modelo de la Figura 8 con el proceso de AC propuesto en la Tabla I, vemos que la creación y adquisición de conocimiento se relaciona estrechamente con la aplicación del mismo. A su vez, la externalización del conocimiento es llevada a cabo por medio de procesos de captura y posterior organización y codificación para que pueda ser almacenado. Finalmente, la internalización del conocimiento requiere de la recuperación del conocimiento de donde se encuentra almacenado, para posteriormente filtrar e interpretar lo que realmente será de utilidad para quien lo aplica. A continuación se da una descripción de las etapas propuestas como proceso de AC, desde el punto de vista del modelo de interrelación de la Figura 8.

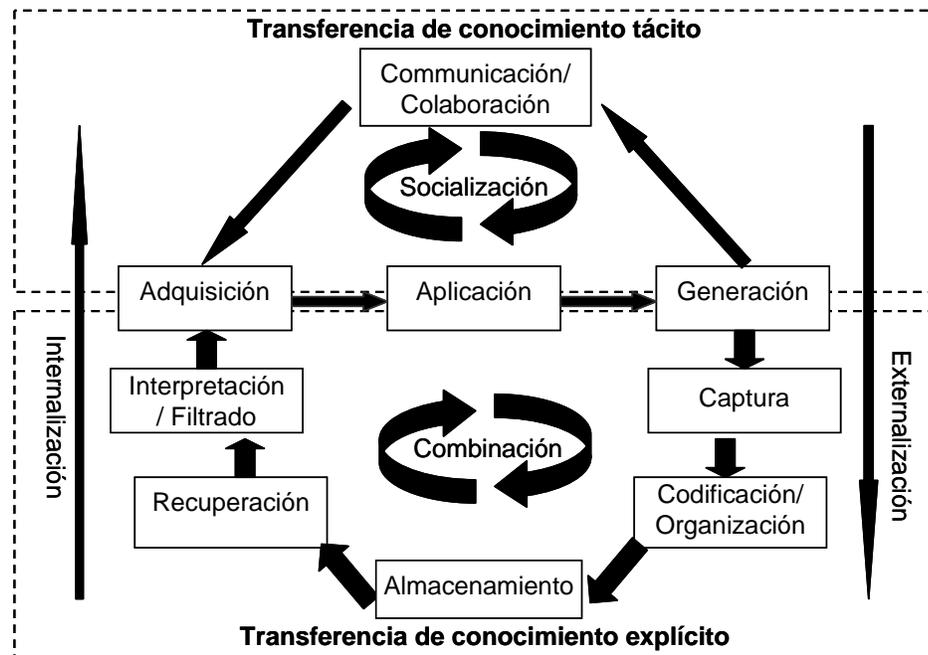


Figura 8. Modelo de interrelación de actividades de AC centrado en la aplicación y transferencia del conocimiento.

II.1.5.1 Creación/Adquisición del conocimiento

En esta etapa se consideran las actividades que permiten tanto la generación de conocimiento por parte de los individuos u organizaciones, así como su adquisición de fuentes externas [Davenport y Prusak, 2000]. La adquisición de conocimiento no se refiere a nuevo conocimiento, sino al que es nuevo para la organización que lo requiere. Por tanto, su enfoque es hacia la obtención de conocimiento existente que puede estar fuera de la organización. Por su parte, la generación de conocimiento está enfocada a la creación de nuevas ideas, como métodos, conceptos, productos, etc. Por lo general en las organizaciones existen recursos dedicados a la generación de nuevos conocimientos por medio de proyectos de investigación y desarrollo.

Existen distintas formas en que puede darse la creación o adquisición de conocimiento [Meehan y Richardson, 2002; Rus y Lindvall, 2002]. Por ejemplo, por medio de procesos de aprendizaje, ya sea de manera formal mediante cursos o capacitación, o por

medio de la práctica, es decir, aprender haciendo. Otras formas de crear conocimiento pueden darse por medio de la innovación o creatividad de los individuos, donde éstos generan ideas a partir de su conocimiento previo, experiencia, intuición, intereses, necesidades, etc. Considerando el modelo SECI [Nonaka y Takeuchi, 1995] descrito anteriormente, la creación de conocimiento también se da durante las etapas de formalización, donde se crea conocimiento explícito del conocimiento tácito; durante la interpretación y filtrado (internalización), al crear conocimiento tácito del conocimiento explícito; durante la socialización o consulta con otras personas; y al combinar fuentes de conocimiento explícito. La creación y adquisición de conocimiento tiene una fuerte relación con los procesos donde éste es requerido o aplicado, dado que cuando no se cuenta con el conocimiento suficiente para solucionar un problema o realizar alguna actividad es necesario crearlo o adquirirlo de alguna manera [Kucza et al., 2001].

II.1.5.2 Externalización del conocimiento

La etapa de externalización agrupa todas las actividades encaminadas a permitir capturar, organizar, estructurar, representar, codificar, etc. el conocimiento con el fin de facilitar su manejo [Davenport y Prusak, 2000]. El primer paso en la formalización del conocimiento tácito es su captura. Una vez capturado, éste puede ser codificado y organizado con el resto del conocimiento explícito. Un primer paso hacia la captura del conocimiento es su identificación. Es necesario identificar el conocimiento que requiere ser capturado, para poder definir los mecanismos por medio de los cuales realizar su captura; por ejemplo, por medio de entrevistas a quienes cuentan con ese conocimiento, o proporcionando herramientas para que las personas puedan expresar su conocimiento con facilidad, de forma que éste pueda ser almacenado para que no se pierda.

Wiig [1993] sostiene que para que el conocimiento sea de utilidad, éste debe estar organizado. En este proceso de organización, se deben considerar las fuentes donde se localiza el conocimiento y las interrelaciones entre los distintos tipos de conocimiento y fuentes. Es importante que los esquemas de codificación empleados, así como el contexto, sea compartido con quienes harán uso de éste conocimiento [Dalkir, 2005: p. 66].

Uno de los medios más usados en la AC para organizar y codificar conocimiento son las ontologías [Abdullah *et al.*, 2002; Fensel, 2001; Maedche *et al.*, 2003]. Las ontologías permiten definir de manera explícita los principales conceptos involucrados en las actividades que realiza la organización, así como las relaciones existentes entre estos, de forma que se puede especificar, estructurar y organizar explícitamente el conocimiento de la organización por medio de ontologías. De hecho, las ontologías han sido definidas como un medio para compartir conocimientos [Gruber, 1995]. Otras formas de codificación y organización de conocimiento son las taxonomías, que permiten definir estructuras jerárquicas entre los tipos de fuentes de conocimiento, o de los conocimientos mismos [Dalkir, 2005: p. 96].

Existen distintos medios para facilitar la formalización de conocimiento. Un ejemplo es la estandarización de los documentos que se generan durante las actividades que se realizan en la empresa. Se puede definir la estructura y el contenido que deben tener y proporcionar plantillas para facilitar la generación de los mismos. Sin embargo, no todo el conocimiento puede ser fácilmente capturado y codificado. Particularmente el conocimiento más valioso, como la experiencia y habilidades de los empleados. La codificación y organización de este conocimiento puede centrarse en definir esquemas para facilitar la identificación de las fuentes de donde puede ser obtenido. Por ejemplo, por medio de la definición de mapas de conocimiento, los cuales son guías que apuntan hacia donde está el conocimiento, mas no lo contienen [Davenport y Prusak, 2000: p. 72].

II.1.5.3 Almacenamiento/Mantenimiento del conocimiento

El almacenamiento del conocimiento se da cuando los individuos lo guardan en forma de información o datos en repositorios de documentos, bases de datos, etc. [Meehan y Richardson, 2002], esto sólo puede darse una vez que el conocimiento ha sido formalizado. Al almacenar el conocimiento se evita que se pierda y se facilita su reutilización, ya sea por el mismo individuo u organización que lo crea, o por otros que pudieran requerirlo. En esta etapa se deben considerar los medios de almacenamiento; su alcance, es decir, si son de acceso personal, grupal, organizacional, o inter-organizacional;

los mecanismos utilizados para almacenar los datos y fuentes de información, así como el tipo de información y conocimiento que se almacena en estos medios. Por lo general, el conocimiento se encuentra almacenado en medios muy diversos, por lo que es necesario proveer mecanismos que faciliten el acceso a esta gran diversidad de fuentes. Los sistemas de manejo de la memoria organizacional han propuesto avances significativos en este sentido [Borghoff y Pareschi, 1998].

Por otra parte, un aspecto importante que debe ser tomado en cuenta al manejar conocimiento almacenado, es su mantenimiento. Es importante que el conocimiento se mantenga actualizado para evitar confusiones; por ejemplo, al utilizar información o conocimiento obsoleto. Puede darse el caso de la existencia de conocimiento que tenga una utilidad temporal [Dalkir, 2005: p. 16], y una vez pasado cierto tiempo, éste deje de ser útil, por lo que deberá ser posible eliminarlo del repositorio de conocimiento, o almacenarlo en algún otro lugar. Otro ejemplo de la importancia del mantenimiento puede darse si se manejan repositorios de conocimiento distribuido, por lo que debe ser posible seguir las versiones más actualizadas, en caso de que existan copias en varias localidades.

II.1.5.4 Transmisión del conocimiento

El objetivo de la AC es que el conocimiento fluya hacia donde se requiere [Borghoff y Pareschi, 1997], por lo que los mecanismos o procesos que habilitan esa transferencia de conocimiento se vuelven de vital importancia para la AC. El proceso de transmisión puede verse desde dos perspectivas: la transmisión de conocimiento tácito y la transmisión de conocimiento explícito.

La transmisión del conocimiento tácito se da principalmente a través de procesos de socialización [Davenport y Prusak, 2000: p. 88]. Por medio de estos procesos, los individuos comparten sus experiencias con otros, con el fin de apoyarlos, ya sea en la realización de alguna actividad, o simplemente para que incrementen su conocimiento. Este proceso de socialización puede ser fomentado al participar en grupos de trabajo, grupos de discusión o comunidades de práctica. Las comunidades de práctica son un buen mecanismo

para agilizar y facilitar la transmisión de conocimiento, debido a que son formadas por grupos de personas con intereses comunes, con el fin de compartir el conocimiento de los distintos miembros y aprender juntos [Wenger, 1998]. Las herramientas que apoyan o facilitan la socialización y el trabajo en grupos son de particular interés para apoyar los procesos para transferir y compartir conocimientos tácitos entre personas.

El conocimiento explícito puede transmitirse al almacenarlo en medios compartidos, como bases de datos, repositorios de documentos, memorias organizacionales, etc. A través de estos medios, el conocimiento explícito puede ser aprovechado por otros miembros del equipo, de la organización, e incluso por otras organizaciones. Sin embargo, si las personas no tienen idea del conocimiento existente en un repositorio, es posible que no lo consulten, por lo que también pueden considerarse mecanismos que permitan informar a las personas sobre los nuevos conocimientos que se van agregando, permitiendo la difusión del mismo. Otro problema relacionado con la transferencia de conocimiento explícito, es la dificultad para encontrar lo que se busca dentro de un repositorio. Los trabajadores a cargo de actividades altamente basadas en conocimiento pasan gran parte de su tiempo buscando información o conocimiento, o ayudando a otros a hacerlo. Debido a esta dificultad para encontrar lo que necesitan, muchos de ellos prefieren acudir con otras personas antes que acceder a alguna fuente antipersonal, o a un sistema de AC [Dalkir, 2005: p. 111]. Por lo tanto, un sistema de este tipo no solo debe proveer medios para almacenar el conocimiento, sino también proveer mecanismos que faciliten la recuperación del mismo de una forma rápida y eficiente.

II.1.5.5 Internalización del conocimiento

En la internalización del conocimiento intervienen dos procesos principales, la recuperación de éste desde los medios donde se encuentra almacenado, y su posterior interpretación y filtrado.

Recuperación del conocimiento. En esta etapa se consideran los mecanismos utilizados y procesos seguidos para recuperar datos, información o conocimiento que se encuentran

almacenados en medios formales. Los mecanismos pueden incluir motores de búsquedas en repositorios de documentos, sistemas de consultas a bases de datos, etc. Entre los problemas más comunes al recuperar información, se encuentra el que las consultas de búsqueda por lo general recuperan una gran cantidad de documentos que no son relevantes para el usuario [Marwick, 2001]. Además, es posible que sólo parte de la información de un documento dado sea relevante, e incluso, que deba ser interpretada y adaptada para que sea útil. Por lo tanto, una vez recuperada la información es necesario un proceso de interpretación y filtrado para obtener lo que realmente se requiere.

Interpretación y filtrado del conocimiento. El proceso de interpretación y filtrado es el que permite interiorizar el conocimiento explícito recuperado en forma de documentos o datos, y convertirlo en conocimiento tácito para su aplicación en la realización de alguna actividad. En este proceso de interpretación y filtrado interviene principalmente el factor humano, ya que es el mismo conocimiento previo el que permite determinar qué información es importante para una persona en un momento determinado. Así también, es la misma persona la que asimila la información para convertirla en conocimiento que pueda ser aplicado. Aún así, pueden existir mecanismos que faciliten estos procesos. Por ejemplo, existen trabajos que utilizan agentes de software que, con base en el perfil del usuario y el contexto en el que se encuentra (actividad que se encuentra realizando, tipo de información que está consultando, etc.), apoyan en el filtrado de la información, disminuyendo el sobre flujo de la misma en el usuario [Borghoff y Pareschi, 1998; Maes, 1994; Tiwana, 2000:137]. Sin embargo, el proveer tecnología para apoyar a los usuarios a lograr una mejor y más rápida apreciación, entendimiento y asimilación del conocimiento explícito para convertirlo en tácito, es uno de los principales retos dentro de la AC [Marwick, 2001].

II.1.5.6 Aplicación/Evolución del conocimiento

El valor del conocimiento como una ventaja competitiva para una organización, no reside en el conocimiento en sí, sino en la forma en que éste es aplicado [Alavi y Leidner, 2001], ya sea para mejorar el rendimiento de los miembros de la organización, mejorar sus procesos productivos, tomar mejores decisiones, adaptarse rápidamente a los cambios del

mercado, etc. Por lo tanto, la aplicación del conocimiento es de suma importancia para entender cómo el conocimiento fluye dentro de la organización [Rus y Lindvall, 2002]. El proceso de aplicación del conocimiento integra al resto de las etapas, dado que al momento de realizar actividades, tomar decisiones, etc., las personas encargadas deben de obtener la información y conocimiento que les permita llevar a cabo sus tareas. Esta necesidad de conocimiento es la que dispara el resto de los procesos del flujo del conocimiento. Por ejemplo, la obtención del conocimiento necesario para tomar una decisión puede generar procesos de búsqueda de información o socialización; la necesidad de generar un escrito para documentar una actividad, genera un proceso de formalización del conocimiento que se tiene sobre esa actividad, así como un proceso de almacenamiento del documento generado en algún repositorio, etc. Más aun, el proceso de aplicación de conocimiento por si mismo genera nuevo conocimiento. Incluso podemos decir que parte del conocimiento más valioso de una organización es generado precisamente durante el proceso de aplicación, dado que aplicando el conocimiento es como los individuos obtienen experiencia. Por lo tanto, la aplicación del conocimiento permite que evolucione, incrementándose o haciéndose más especializado.

Aun cuando la aplicación del conocimiento es el fin último de la AC, son pocos los trabajos que pueden encontrarse en la literatura que explícitamente se centran en esta actividad. Quizá el problema sea la dificultad para facilitar el proceso de aplicación del conocimiento. Si consideramos la aplicación del conocimiento a nivel individual, es posible encontrar que diferentes personas, bajo la misma situación, aplicarán conocimientos distintos y de formas distintas. Esto se debe a que las características de cada individuo que busca conocimiento para aplicarlo juegan un rol importante a la hora de buscar, encontrar, entender, y usar ese conocimiento [Dalkir, 2005: p. 149]. Sin embargo, actualmente existen sistemas que pueden ayudar a solventar este problema, por ejemplo, al mantener un perfil de las preferencias o conocimiento del usuario con base en el cual personalizar los contenidos que se le presentan, adelantar búsquedas de información, u organizar los resultados de las mismas.

II.2 Administración del conocimiento con enfoque en procesos organizacionales

Hasta este punto hemos dado una descripción general del significado de la administración del conocimiento, del conocimiento mismo, y hemos presentado modelos para explicar cómo este fluye en las organizaciones. Con base en el análisis de lo anterior, se presentó también un modelo de la interrelación de las distintas actividades de la AC, con lo que se definió un proceso de AC y se dio una descripción de cada una de las etapas de este proceso. Es importante tener en cuenta que este proceso de AC no debe ser un proceso aislado dentro de la organización, sino que debe estar embebido dentro de los procesos de trabajo de la misma.

Con frecuencia las iniciativas de AC no son exitosas debido a que están desconectadas del trabajo real de las organizaciones [Stewart, 2002]. Por ejemplo, algunas de estas estrategias están orientadas a proveer herramientas que requieren que los empleados realicen trabajo extra, lo que resulta en una barrera para su adopción [Desouza, 2003]. Por lo tanto, es importante considerar los procesos de trabajo reales de la organización, así como la infraestructura tecnológica que apoya estos procesos, con el fin de poder definir estrategias de AC que sean exitosas [Jennex y Olfman, 2005; Maier y Remus, 2002]. Incluso, en un estudio realizado en 2002, se detectó que los expertos en el área de AC consideran que el aspecto más importante en el futuro de la misma, está en lograr una mejor integración de la AC con los procesos de la organización [Scholl *et al.*, 2004]. Al parecer, la mejor forma de lograr esto es buscar integrar la AC dentro de las tecnologías que los trabajadores del conocimiento usan para hacer su trabajo [Davenport, 2007]. En este sentido, el uso de técnicas de análisis de procesos puede ser beneficioso a la hora de definir estrategias de AC que estén integradas al trabajo real de las organizaciones, y que consideren las necesidades de conocimiento reales de las mismas [Maier y Remus, 2002]. Un medio útil para la identificación de estas necesidades es el modelado de procesos para analizar cómo fluye el conocimiento a través de los mismos [Bera *et al.*, 2005; Hansen y Kautz, 2004; Rodríguez *et al.*, 2004b]. El modelado de procesos puede apoyar en la identificación del conocimiento que entra y sale de las actividades de una organización, las

fuentes donde este conocimiento es obtenido o almacenado, y los flujos del conocimiento a través de las actividades, personas, u otro tipo de fuentes. [Abdullah et al., 2002; Davenport y Prusak, 2000]. Además, a través del modelado es posible detectar los problemas que puedan estar afectando al flujo del conocimiento, facilitando, a su vez, la definición de estrategias para mejorar este flujo.

Con base en lo anterior es que en este trabajo se propone una metodología basada en técnicas de ingeniería de procesos para estudiar flujos de conocimiento en procesos de ingeniería de software, con el fin de identificar requerimientos para el diseño de sistemas que ayuden a mejorar estos flujos, considerando las tecnologías que son usadas para desempeñar las actividades del proceso. El objetivo de la definición de esta metodología, fue el facilitar el estudio de flujos de conocimiento en grupos de mantenimiento de software, debido a que se identificó una falta de trabajos encaminados a proveer métodos para facilitar el diseño de sistemas de AC para este tipo de grupos. En particular, métodos centrados en la identificación de requerimientos de diseño sistemas o estrategias de AC basados en el estudio de flujos de conocimiento en procesos de software. Como se describe en las siguientes secciones.

II.3 Administración del conocimiento en ingeniería de software

La ingeniería de software es un proceso altamente basado en conocimiento, de ahí el creciente interés en organizaciones de desarrollo de software, por proveer métodos que ayuden a una buena administración del mismo [Aurum *et al.*, 2003]. Entre los beneficios potenciales que la AC puede traer a las organizaciones de desarrollo de software se encuentran: la reducción de costos, mejoras en la calidad de los procesos y productos, incremento en la satisfacción de los desarrolladores y mejor toma de decisiones, entre otros [Dingsøyr y Conradi, 2002; Rus y Lindvall, 2002].

Durante las distintas actividades de las fases del ciclo de vida del software, se genera conocimiento que es importante para otras fases [Edwards, 2003] que comúnmente

son llevadas a cabo por personas distintas. Es importante asegurar que este conocimiento sea accesible para estos últimos en el momento que lo requieran. Sin embargo, algunos problemas frecuentes en las organizaciones de desarrollo de software representan también problemas en el flujo del conocimiento, por ejemplo, la falta de documentación, o una constante rotación de personal [Dart *et al.*, 1993; Lethbridge *et al.*, 2003; Lientz, 1983; Seaman, 2002; Singer, 1998]. La documentación es una de las fuentes más importantes para transferir conocimiento entre los desarrolladores del software y los encargados de su mantenimiento [Curtis *et al.*, 1988]. Sin embargo, durante el mantenimiento, es común tener problemas por falta de documentación, o porque esta no esté actualizada, aunque, según se cita en la literatura, la documentación no actualizada puede seguir siendo útil en muchos casos [Lethbridge *et al.*, 2003]. Por otro lado, gran parte del conocimiento más valioso para una organización de desarrollo de software se pierde debido a la rotación constante del personal [Rus y Lindvall, 2002]. Este problema se agudiza si consideramos que, por lo general, los que toman los puestos de los empleados que dejan la organización o que son asignados a otros proyectos, son desarrolladores sin experiencia [Thomsett, 1998].

Con el fin de buscar mecanismos para solucionar los problemas relacionados con la pérdida y desaprovechamiento del conocimiento, es importante entender la situación actual de la organización. Este estudio debe enfocarse a identificar aspectos tales como el conocimiento requerido y generado en las principales actividades realizadas, los canales a través de los cuales ese conocimiento fluye, qué conocimiento está realmente fluyendo, y cual no, etc. [Aurum *et al.*, 2003]. Todo esto con el fin de definir posibilidades para mejorar el flujo del conocimiento a través de la organización.

Existen diversos trabajos centrados en aplicar la AC en organizaciones de desarrollo de software, la mayoría de los cuales se enfocan en la reutilización de experiencias, tales como mejores prácticas o lecciones aprendidas, con el fin de mejorar la calidad de los nuevos productos, del proceso de desarrollo, o para facilitar la reutilización de artefactos de software [Althoff *et al.*, 2002; Althoff *et al.*, 1999; Basili y Caldiera, 1995; Basili *et al.*, 1994; Dingsøyr y Conradi, 2002; Komi-Sirviö *et al.*, 2002; Kucza y Komi-Sirviö, 2001; Kucza *et al.*, 2001; Schneider *et al.*, 2002; Seaman *et al.*, 2003]. Comúnmente, este tipo de

trabajos se enfoca en la captura de conocimiento relacionado con el desarrollo de nuevos productos, dejando de lado el mantenimiento del software actual.

Gran parte de las investigaciones relacionadas con la aplicación de la AC en la ingeniería de software, no distinguen entre desarrollo y mantenimiento. Quizá esto se deba a que los desarrolladores y encargados del mantenimiento requieren conocimientos de dominios similares, como del dominio de la aplicación, o conocimientos técnicos, por ejemplo, diseño, programación, pruebas, etc. [Curtis et al., 1988; Klint y Verhoef, 2002; Lethbridge, 2000; Robillard, 1999; Rus *et al.*, 2001; Tiwana, 2004; Walz *et al.*, 1993]. Sin embargo, el mantenimiento y el desarrollo de software son diferentes debido a que los encargados del primero deben trabajar sobre sistemas existentes que con frecuencia son complejos, por lo que puede ser difícil analizar y entender estos sistemas. Como consecuencia, los encargados del mantenimiento requieren conocimientos y herramientas distintas a las requeridas por los desarrolladores [Dart et al., 1993; de Sousa *et al.*, 2004].

Parte del conocimiento requerido durante el mantenimiento es generado durante el proceso de desarrollo, al igual que a través de las distintas fases de mantenimiento por las que puede pasar un sistema durante su vida. Aunado a esto, el conocimiento generado durante el mantenimiento puede también ser útil a la hora de desarrollar nuevos sistemas [Edwards, 2003]. Por lo tanto, al igual que cualquier otro proceso de software, el mantenimiento es una actividad altamente basada en conocimiento. Incluso, el mantenimiento puede ser un trabajo aun más demandante en conocimiento que otros procesos de software [Chapin, 1987]. Si bien muchas de las actividades del desarrollo y mantenimiento de software son las mismas, los encargados del mantenimiento también requieren conocer y entender la estructura y funcionalidad de los sistemas que modifican [Glass, 2001], así como los posibles efectos que estos cambios puedan tener en otras partes del sistema [Oliveira *et al.*, 2003]. Si consideramos los problemas relacionados con el acceso a fuentes de información y conocimiento durante el mantenimiento de software [Lethbridge et al., 2003; Lientz, 1983; Seaman, 2002; Singer, 1998], entonces tenemos que el mantenimiento es una actividad que se basa fuertemente en la experiencia de quienes la

realizan, por ejemplo, en su experiencia y habilidad para analizar código fuente, o para reproducir errores en el sistema [Singer, 1998].

Debido a esta alta dependencia del conocimiento de quienes realizan el mantenimiento, los problemas relacionados con la pérdida y desaprovechamiento del conocimiento conducen a un bajo nivel de madurez en los procesos de mantenimiento de software [Chapin, 2003]. Por tanto, es importante buscar mecanismos para ayudar a los grupos encargados de este proceso a manejar de una mejor manera el conocimiento con el que cuentan, y facilitarles el acceso al conocimiento que requieren para sus actividades. Si a esto agregamos la importancia que tiene el mantenimiento con respecto a los recursos que exige de las organizaciones de desarrollo de software [Polo et al., 2003b], entonces el proceso de mantenimiento de software se vuelve un caso interesante para investigar cómo la AC puede aplicarse para abordar parte de los problemas relacionados con la pérdida y desaprovechamiento del conocimiento en este proceso.

II.4 Requerimientos para el estudio del proceso de mantenimiento de software con enfoque en la AC

Del estudio de la literatura, resulta evidente que aplicar técnicas de AC en grupos de desarrollo y mantenimiento de software puede aportar diversos beneficios. Sin embargo, no existe una idea clara de cómo esto debe hacerse [Aurum et al., 2003: p. ix].

Por otra parte, para definir estrategias de AC que sean exitosas, es necesario tomar en cuenta aspectos, tales como:

- *Basarse en el trabajo y necesidades reales de la organización donde se pretende implementar estas estrategias o sistemas [Stewart, 2002]. Es importante entender los procesos de trabajo de la organización y cómo relacionar estos procesos con las actividades de la AC [Maier y Remus, 2002].*
- *Buscar facilitar la aplicación del conocimiento por medio del mejoramiento de los flujos de conocimiento a través de los individuos, fuentes de conocimiento, y actividades. Por*

lo que es importante entender cómo fluye el conocimiento en la organización antes de definir estas estrategias [Nissen, 2002].

- *Centrarse en lo que es importante para los trabajadores del conocimiento; aquellos que lo contienen o lo requieren para sus actividades.* Es importante que las estrategias estén centradas en facilitar las actividades de los trabajadores del conocimiento [Wiig, 2004]. Para esto, se debe hacer un análisis de necesidades para entender cómo la AC podría beneficiar a individuos específicos, a grupos, y a la organización como un todo [Dalkir, 2005: p. 9].
- *Buscar aprovechar lo más posible la infraestructura tecnológica existente,* con el fin de facilitar la adopción de la AC por parte de los encargados de los procesos. Si se proporciona una infraestructura tecnológica totalmente nueva a la hora de implantar la AC, es posible que esto requiera cambios en la forma de trabajo de las personas que provoque que la AC no sea aceptada [Aurum et al., 2003: p. xii; Desouza, 2003]. Con frecuencia, en las organizaciones existen herramientas de uso común que de alguna manera dan soporte a una o varias de las actividades de la AC [Bontis *et al.*, 2003; Lindvall y Rus, 2003], así mismo, la mejor forma de integrar el conocimiento en los procesos de trabajo es a través de las tecnologías comúnmente usadas para realizarlos [Davenport, 2007]. Por lo tanto, si logramos integrar estas herramientas como parte de la estrategia o sistema de AC, es posible que se facilite su asimilación y aceptación por parte del personal encargado de los procesos.

En este sentido, para aplicar la AC en un grupo de desarrollo o mantenimiento de software, vemos la necesidad de un enfoque metodológico que permita estudiar los procesos que realizan con el fin de identificar las necesidades de conocimiento reales de los grupos a cargo de estos procesos, con el propósito de buscar mejorar el flujo del conocimiento, de tal forma que al facilitar este flujo, se facilite a su vez la aplicación del conocimiento. Entre las preguntas que se requiere contestar por medio de la aplicación de un enfoque metodológico con las características descritas, podemos definir las siguientes [Aurum et al., 2003: p. xii; Oliveira et al., 2003]:

- ¿Qué conocimiento es importante para los encargados del proceso?,
- ¿Cuáles son las fuentes que utilizan para obtener ese conocimiento?,
- ¿Qué mecanismos utilizan para consultar las fuentes de información y conocimiento?
- ¿Cuál es el conocimiento y las fuentes envueltos en las actividades que componen el proceso?
- ¿Cuál es la infraestructura tecnológica que interviene en el flujo del conocimiento?
- ¿Cuáles son los problemas que pueden estar afectando al flujo del conocimiento durante el proceso?
- ¿Qué tipo de apoyo puede ayudar a reducir estos problemas con el fin de facilitar, mejorar e/o incrementar el flujo de conocimiento?

Con base en los requerimientos presentados fue que se definió la metodología para la identificación y modelado de flujos de conocimiento propuesta en este trabajo. Esta metodología, así como las técnicas empleadas por la misma, son descritas en los capítulos siguientes.

II.5 Resumen del capítulo

En este capítulo se ha dado una descripción de los principales conceptos teóricos involucrados con el trabajo realizado y reportado en este documento. Como parte de los resultados del análisis de la literatura realizado, tenemos los siguientes;

- Se definió un modelo que integra las actividades de AC como un ciclo centrado en la aplicación y transferencia del conocimiento. Así mismo, se definieron modelos que muestran la interrelación de las principales teorías y modelos de flujo del conocimiento presentes en la literatura.
- Se definió un modelo de proceso de AC que ha sido tomado como base en el análisis de procesos con enfoque en el flujo del conocimiento que se da durante los mismos.
- Se definieron una serie de requerimientos que deben ser cubiertos por una metodología enfocada a apoyar el estudio de procesos de software con enfoque en el flujo del

conocimiento. Esto con el fin de obtener requerimientos de diseño para sistemas o estrategias centrados en mejorar los flujos de conocimiento en dichos procesos.

En el siguiente capítulo se presenta la metodología que fue definida como resultado de los requerimientos obtenidos del análisis de la literatura.

^{iv} Referencia tomada de (Dalkir, 2005: p. 25). Kahlil Gibran fue un ensayista filosófico, poeta y artista libanés-americano, quizá su libro más famoso sea “EL Profeta”.

CAPÍTULO III

KoFI: una metodología para el estudio de flujos de conocimiento

*“Es mucho más fácil sugerir soluciones cuando no conoces el problema” **Malcolm Forbes (1919-1990)**”*

En el capítulo anterior se definió un conjunto de requerimientos para el estudio de procesos de ingeniería de software desde un enfoque orientado a la administración del conocimiento. Como resultado de estos requerimientos, en este trabajo definimos una metodología¹ para el análisis e identificación de flujos de conocimiento en procesos organizacionales. El objetivo de la metodología está centrado en apoyar en la obtención de requerimientos de diseño para sistemas enfocados a facilitar el flujo del conocimiento en los procesos de trabajo. Este capítulo describe la metodología; que se ha denominado KoFI (Knowledge Flow Identification). Se presenta una descripción general de KoFI, así como los detalles de las distintas etapas junto con algunas de las técnicas empleadas en cada etapa. Debido a su extensión, algunas de las técnicas propuestas son detalladas en capítulos posteriores, en particular una propuesta de modelado de procesos descrita en el Capítulo IV, y un marco de trabajo para el análisis de herramientas de apoyo al flujo del conocimiento descrito en el Capítulo V.

¹ El diccionario de la Lengua Española, 22ª Edición, define metodología como la ciencia del método, o como un conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal. Por su parte, el Cambridge Advanced Learner's Dictionary, define el término metodología (Methodology en inglés) como un sistema de formas de hacer, enseñar o estudiar algo.

III.1 Descripción general de KoFI

Del estudio de la literatura presentado en el capítulo anterior, identificamos que para implementar la AC, es necesario tener un enfoque centrado en el trabajo llevado a cabo en la organización donde se quiere utilizar, así como en el conocimiento que los empleados de la misma requieren para poder desempeñar este trabajo. Para lograr lo anterior, se observó que se requiere de un enfoque metodológico que nos ayude a estudiar y analizar los procesos de trabajo con el objetivo de contestar preguntas tales como ¿Qué conocimiento es importante para los encargados del proceso?, o ¿Cuál es el conocimiento y las fuentes envueltos en las actividades que componen el proceso?, entre otras.

Con base en lo anterior fue que se definió la metodología KoFI, la cual propone una guía que ayuda a la identificación y entendimiento de flujos de conocimiento en procesos organizacionales, por medio del modelado de estos flujos utilizando un enfoque de ingeniería de procesos. KoFI se orienta a la obtención de información que apoye el diseño de sistemas de AC que den soporte al flujo del conocimiento. Sin embargo, no se pretende que la metodología sea la guía hacia el desarrollo de sistemas de software, sino hacia la propuesta de alternativas de solución a problemas específicos. Como mencionan Cox *et. al.* [2005b], los modelos de un proceso deben ayudar a informar requerimientos sin asumir un enfoque de diseño particular. Por tanto, las alternativas de solución a los problemas observados no deberán enfocarse exclusivamente al desarrollo de sistemas de software, sino que podrán consistir en cambios, ya sea en los procesos de trabajo, en el modo de uso de la infraestructura tecnológica de trabajo, en la infraestructura misma, o la adquisición o desarrollo de nuevas herramientas.

Son tres los aportes principales hacia los que se enfoca KoFI:

- 1) Obtener información que ayude a estructurar una base de conocimientos para un grupo u organización determinada. Como se puede observar en [Rao, 2005b], uno de los pasos importantes en la definición de estrategias y sistemas de AC, es hacer una identificación y clasificación de los conocimientos de la organización, así como las fuentes de donde

se obtiene o donde se almacena. Esto puede lograrse por medio de la definición de taxonomías que permitan clasificar y organizar el conocimiento y sus fuentes. Es importante que estas taxonomías reflejen la manera en que los trabajadores realizan sus actividades [Rao, 2005b: p. 39], por lo tanto, al definir estas taxonomías es importante considerar el conocimiento y sus fuentes que están involucrados en las actividades más relevantes de los procesos de trabajo.

- 2) Identificar la infraestructura tecnológica que apoya los procesos de trabajo y que interviene en el flujo del conocimiento, así como la forma en que la misma apoya o dificulta dicho flujo. La infraestructura tecnológica es un aspecto esencial para las estrategias de AC, ya que ésta puede ser un factor determinante en el éxito o fracaso de las mismas [Jennex y Olfman, 2005].
- 3) Obtener requerimientos que ayuden en el diseño de sistemas de AC centrados en dar soporte al flujo de conocimiento dentro del grupo u organización. Los sistemas o estrategias de AC deben enfocarse a mejorar el flujo del conocimiento relevante para las actividades que deben ser realizadas [Borghoff y Pareschi, 1997; Maier y Remus, 2002], por lo que es importante estudiar estos flujos para entenderlos e identificar fallas y debilidades, con el fin de proponer soluciones [Nissen, 2002].

Para la obtención y análisis de información, KoFI hace uso de técnicas de modelado de procesos. Existen diversas metodologías a seguir para el modelado de procesos, las cuales buscan objetivos diversos [Curtis *et al.*, 1992]. En nuestro caso hemos elegido la metodología de análisis y diseño de procesos (PADM por sus siglas en inglés) [Wastell *et al.*, 1994], debido a que es una metodología que conocemos y que ha sido ampliamente usada en nuestro ámbito local. Además, considera aspectos técnicos y sociales dentro de los procesos, lo que es importante a la hora de estudiar flujos de conocimiento, ya que estos dependen de las personas involucradas, por lo que pueden verse afectados de manera importante por aspectos sociales, como lo reportan diversos autores en el área [Dalkir, 2005; Davenport y Prusak, 2000; Tiwana, 2000]. De hecho, Karl Wiig [2004], uno de los pioneros en el área de AC, sostiene que los nuevos enfoques de AC deben estar centrados en los trabajadores que contienen y hacen uso del conocimiento. Es decir, que es necesario

entender las necesidades de estos, con el fin de proponer herramientas que les faciliten su labor.

La metodología PADM toma en consideración dos enfoques en el estudio de procesos, el primero es lograr un entendimiento del proceso en términos generales, para posteriormente comparar esta representación con el proceso real, a fin de acercarnos más éste en cada iteración. PADM también considera a los procesos como sistemas socio-técnicos, donde es necesario encontrar la manera en que interactúa el subsistema técnico con el subsistema social. Desde el punto de vista de los flujos de conocimiento, esto sugiere que es necesario estudiar la forma en que tanto las personas, como las herramientas de apoyo que usan, contribuyen o intervienen en el flujo del conocimiento. En este sentido, la metodología se ha complementado con técnicas para ayudar en el análisis de flujos de conocimiento tanto entre personas, como entre actividades y herramientas involucradas en el proceso.

PADM define cuatro etapas principales para el estudio de procesos:

- 1) *Definición del proceso*, que incluye identificar las entradas y salidas, así como los actores que interactúan y los roles que juegan dentro del mismo.
- 2) *Captura y representación*, comprende el modelado del proceso; la realización de una representación gráfica del mismo.
- 3) *Evaluación*, se centra en la identificación de las fallas y debilidades del proceso.
- 4) *Rediseño*, comprende el diseño de un nuevo proceso a seguir.

En la metodología KoFI, las dos primeras etapas de PADM son contempladas por medio de la identificación de flujos de conocimiento con base en el modelado de procesos, mientras que las dos últimas son abordadas por medio de la definición de escenarios que muestran los problemas que afectan al flujo del conocimiento, así como posibles alternativas de solución, como se detalla a continuación.

III.2 Fases de la metodología KoFI

KoFI se descompone en tres fases principales (como se ilustra en la Figura 9): la primera encargada del modelado del proceso con un enfoque en los flujos de conocimiento. La segunda es una fase de análisis compuesta de cuatro etapas; las dos primeras se orientan a la identificación de las fuentes de información y conocimiento, y los tipos de conocimiento que pueden aportar. Las siguientes dos etapas, se centran en identificar la forma en que el conocimiento fluye dentro del grupo u organización estudiada, así como los problemas que puedan estar afectando el flujo de conocimiento. Lo anterior con el fin de obtener requerimientos para proponer mejoras al flujo de conocimiento. Finalmente, la tercera fase de KoFI se centra en la identificación de las herramientas de trabajo que pueden estar involucradas en el flujo del conocimiento. Esto con el fin de considerar estas herramientas dentro de las propuestas de mejora, buscando integrar las estrategias de AC con la infraestructura usada para apoyar el proceso.

Dado que cada fase puede generar información útil para complementar al resto, la aplicación de la metodología debe ser un proceso iterativo e incremental, en el cual es posible regresar de una fase a cualquiera de las anteriores. Esto es igual en las etapas de análisis. Así, por ejemplo, si durante la identificación de flujos de conocimientos encontramos la existencia de algún tipo de documento que no ha sido considerado, podemos regresar a la primera etapa para agregarlo al resto, y posteriormente identificar el tipo de información y conocimientos que pueden ser obtenidos en ese tipo de documentos. En las siguientes secciones de este capítulo se detallan las distintas fases y etapas propuestas en KoFI así como algunas de las técnicas empleadas en cada una. Por su parte, el Apéndice D presenta un modelo general de las actividades a seguir durante el uso de la metodología, así como los productos principales involucrados en la misma, con el fin de que sirvan como una guía visual para el uso de la metodología. Este modelo del uso de KoFI se desarrolló siguiendo la técnica de modelado descrita en el siguiente capítulo.

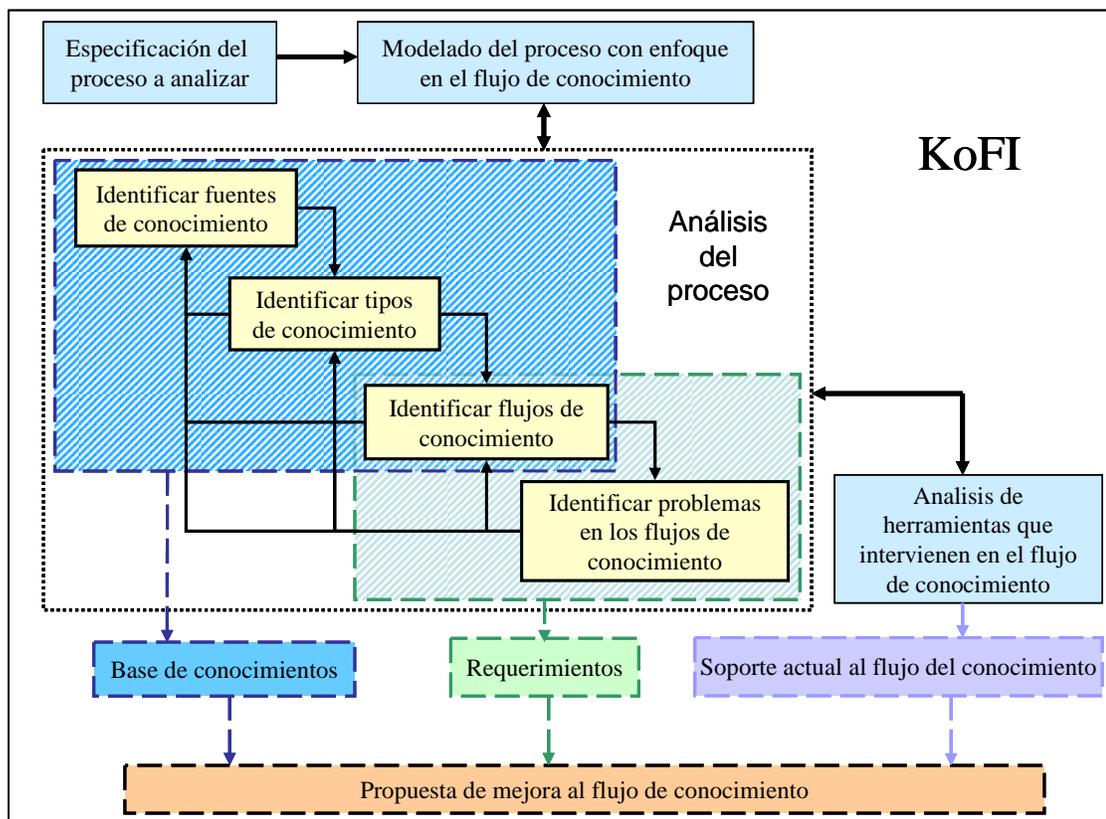


Figura 9. Vista general de la metodología KoFI.

III.3 Modelado del proceso con enfoque en el flujo de conocimiento

Una de las principales técnicas para el modelado de procesos es la representación gráfica del mismo [Curtis et al., 1992]. Existen distintas técnicas diagramáticas para este fin. Algunas de ellas son de propósito general, es decir para distintos tipos de procesos organizacionales, mientras que otras se centran en tipos específicos de procesos. Un lenguaje genérico, si bien puede ser adaptado a distintos contextos, por su misma generalidad, en ocasiones no facilita el análisis de aspectos específicos de ciertos tipos de procesos. En cambio, un lenguaje centrado en un tipo particular de proceso, podría presentar la ventaja de que en su definición integra aspectos que son importantes en ese tipo de proceso. Así mismo, el manejar elementos que son comunes a las personas que

participan dentro del proceso podría ayudar en la obtención de retroalimentación por parte de estas personas. Esto último es importante si consideramos que comúnmente un modelo de un proceso se realiza de forma escalonada e iterativa; es decir, se genera un modelo general que se va refinando por medio de la información proporcionada por las personas que participan en dicho proceso. Esta información es en ocasiones derivada del análisis que dichas personas hacen sobre los modelos previos del proceso. Por lo tanto, la inclusión explícita de los elementos de un proceso en el modelo del mismo, así como el uso de un lenguaje que sea entendible a quienes participan en dicho proceso, facilita su análisis [Conradi y Jaccheri, 1999].

Por otro lado, los lenguajes de modelado de procesos pueden agruparse en flexibles y formales. Los flexibles son aquellos que pueden ser adaptados fácilmente según las necesidades del modelador del proceso, mientras que los formales definen mecanismos estructurados y un lenguaje bien definido por medio de los cuáles el modelador estructura el proceso siguiendo una especificación formalmente predefinida. Los lenguajes flexibles son comúnmente usados en etapas iniciales del modelado del proceso, con el fin de obtener una visión global del mismo [Monk y Howard, 1998]. Sin embargo, suele ser más fácil realizar un análisis detallado del proceso al utilizar lenguajes formalmente definidos [Curtis et al., 1992].

Con base en todo lo expuesto anteriormente, para la realización de la fase de modelado de la metodología KoFI, se propone dividir el modelado gráfico de los procesos en dos niveles de abstracción. El primero para identificar los procesos en términos generales y el segundo para definir los procesos con un mayor nivel de detalle. Para el primer nivel de abstracción recomendamos usar un lenguaje flexible y de propósito general, de manera que pueda adaptarse según las necesidades de quien realice el modelo. El segundo nivel de abstracción se encarga de identificar los detalles con respecto a los distintos elementos que participan en dichos procesos. Para esta etapa, proponemos usar un lenguaje formalmente definido, que permita distintas vistas del proceso, y, de ser posible, que esté enfocado al tipo de proceso específico que se quiere analizar.

Con base en las observaciones anteriores, y considerando que el área de aplicación de la metodología propuesta son organizaciones de desarrollo de software, en este documento se proponen dos enfoques de modelado, el primero es por medio de una adaptación de la técnica de Gráfica Rica [Checkland y Scholes, 1999; Monk y Howard, 1998], con el fin de ser usado en la primera etapa del modelado. El segundo enfoque es una adaptación del Metamodelo para Procesos de Ingeniería de Software (SPEM) [OMG, 2002], con el fin de que sea aplicado durante la segunda etapa de modelado, y durante el análisis detallado del proceso. Debido a la extensión de la descripción de dichas propuestas, ambas técnicas son descritas en el siguiente capítulo (Capítulo IV).

III.4 Análisis de flujos de conocimiento

Una vez que se ha obtenido un modelo del proceso se inicia la etapa de análisis del mismo. Este análisis se centra principalmente en los flujos de conocimiento siguiendo un proceso compuesto de cuatro etapas: 1) la identificación de fuentes de conocimiento, 2) la identificación de tipos de conocimiento, 3) la identificación de flujos de conocimiento, y 4) la identificación de problemas en el flujo de conocimiento.

Cabe destacar que dentro de las dos primeras etapas se considera la clasificación de tipos y fuentes de conocimiento, mientras que en la tercera, se considera la relación entre estas fuentes y tipos de conocimiento con las actividades que componen el proceso estudiado. Para facilitar lo anterior, se ha definido un metamodelo de fuentes y tipos de conocimientos que sirva de guía para definir las relaciones entre los tipos y fuentes de conocimiento, y las actividades de un proceso, mostrado en la Figura 10.

El metamodelo mostrado en la Figura 10 se compone de dos elementos principales: conceptos de conocimiento (KConcept) y definiciones de trabajo (WorkDefinition). Una definición de trabajo es un concepto extraído del metamodelo para la ingeniería de procesos de software (SPEM) [OMG, 2002], el cual se refiere a una operación, o conjunto de estas, que deben ser realizadas para completar un proceso. Estas definiciones de trabajo pueden ser flujos de trabajo como subprocesos, actividades, tareas específicas o la toma de alguna

decisión. Las definiciones de trabajo a tomar en cuenta deben tener una razón de existir, es decir, debe estar bien definida la meta u objetivo para el cual dicha definición de trabajo debe ser realizada.

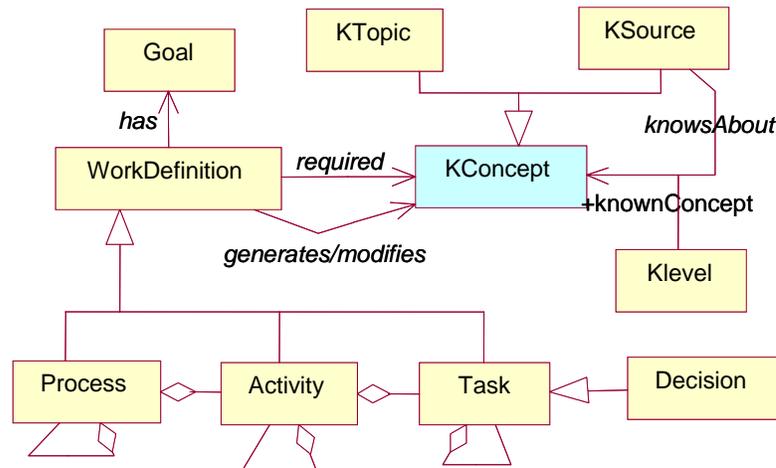


Figura 10. Vista general del metamodelo de fuentes y tipos de conocimiento.

Por otro lado, los conceptos de conocimiento son usados para agrupar, tanto a las fuentes (KSource) como a los tipos (KTopic) de conocimiento que son requeridos, modificados o generados en las definiciones de trabajo. Como se muestra en el metamodelo, cada fuente puede tener un cierto nivel de conocimiento (KLevel) a cerca de uno o más conceptos, ya sea sobre temas o áreas de conocimiento específicas, o sobre alguna otra fuente.

La identificación de fuentes y tipos de conocimiento se hace siguiendo y extendiendo el metamodelo descrito. Estas etapas, junto con las otras dos de la fase de análisis de la metodología KoFI se describen a continuación. Para la ejemplificación de las etapas de análisis se hará uso de un escenario que se resume a continuación.

III.4.1 Escenario ejemplo

El escenario trata sobre la creación de un plan para un proyecto de desarrollo de software. Este plan es realizado entre el jefe del departamento de desarrollo junto con el

responsable del proyecto. El plan de proyecto debe contener información de contacto del cliente solicitante del sistema, la descripción del sistema, la funcionalidad principal esperada, referencias a los documentos asociados (principalmente la especificación de requerimientos), una descripción detallada de las actividades principales que deberán ser realizadas para culminar el proyecto; el tiempo que se estima lleve la realización del proyecto, el costo, personal que participará, recursos requeridos, y una relación detallada de la secuencia de actividades y el personal asignado a cada una.

El objetivo en las siguientes secciones será, entonces, usar el escenario para ejemplificar los pasos de la fase de análisis de la metodología, es decir: identificar fuentes, tipos y flujos de conocimiento, así como problemas en dichos flujos.

III.4.2 Identificación de fuentes de conocimiento

La primera etapa de la metodología consiste en la identificación de las fuentes de información y conocimiento que utilizan los miembros del grupo bajo estudio. En esta etapa se inicia, también, la identificación del conocimiento existente en cada fuente, debido a que se analiza el tipo de información que tiene cada una de ellas, lo que sirve de transición para el inicio de la siguiente etapa.

Los elementos principales que es necesario considerar durante esta etapa son:

- Documentos utilizados o generados dentro del grupo (libros, manuales, tutoriales, reportes, etc.).
- Elementos que componen los productos generados por el mismo. Por ejemplo, en una organización de desarrollo de software los sistemas de software generados son una de las principales fuentes de información y conocimiento (tanto el código fuente como los sistemas ejecutables) [Seaman, 2002; Singer, 1998].
- Personas involucradas en los procesos; los miembros del grupo; personas externas que son consultadas por los miembros del grupo; 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, dentro de una organización de desarrollo de software pueden existir ambientes de desarrollo de software, sistemas de control de versiones, analizadores de código, etc.

De la descripción del escenario ejemplo, podemos identificar varios tipos de fuentes. Primeramente dos tipos de **documentos**: el plan y los documentos asociados, como el caso de la especificación de requerimientos; así mismo, las **personas** involucradas son el jefe del departamento y el responsable del proyecto. Sin embargo, supongamos que para la definición de actividades y asignación de personal los encargados consultan a los posibles participantes del proyecto para pedirles información sobre su disponibilidad, o consultarlos sobre el tiempo que les podría llevar determinadas actividades, con el fin de hacer las estimaciones de tiempo del proyecto. En tal caso, dichas personas consultadas también deberán ser consideradas como fuentes de conocimiento.

En el caso del escenario no se describe el uso de sistemas o herramientas, no obstante, supongamos que para la asignación de personal a las actividades los encargados hacen uso de un sistema de manejo de recursos humanos que contiene información sobre el trabajo asignado a cada miembro de la organización, así como su perfil de experiencia, de forma tal que dicha información sea usada como base para asignar personal a las actividades. En tal caso el sistema podrá ser considerado como una fuente de conocimiento en la categoría de **sistemas de información**. Por otro lado, si los encargados se apoyan con el uso de herramientas, por ejemplo, en la estimación de tiempos y costos de los proyectos tomando como base proyectos previos (un sistema de razonamiento basado en casos, por ejemplo), dicho sistema podría considerarse dentro de la categoría de **herramientas** que apoyan el proceso de desarrollo.

La etapa de identificación de fuentes también considera la definición de un esquema para clasificar y organizar los distintos tipos de fuentes de conocimiento, así como sus relaciones con otros elementos de los procesos llevados a cabo por el grupo en cuestión. Dos técnicas pueden ser usadas para clasificar y organizar las fuentes: taxonomías y

ontologías [Dalkir, 2005: p. 40]. Las taxonomías son una técnica ampliamente usada en sistemas de AC, dado que facilitan la organización del conocimiento y sus fuentes [Rao, 2005b]. Una vez definida dicha taxonomía, la estructuración de la base de conocimientos puede llevarse a cabo por medio de la definición de una ontología [Gruber, 1995]. Así, se puede usar dicha ontología para definir la forma de describir y catalogar las fuentes de conocimiento identificadas, lo que posteriormente puede usarse para crear la base de conocimientos [O'Leary, 1998]. En el Apéndice E se presenta un ejemplo de una ontología obtenida de un proceso de mantenimiento de software donde se aplicó la metodología.

Para ayudar en la clasificación y descripción de fuentes de conocimiento, se agregaron algunos elementos al metamodelo de la Figura 10 (anterior), y se definió una plantilla para la descripción de fuentes de conocimiento que son detallados a continuación.

III.4.2.1 Clasificación de fuentes de conocimiento

Para la clasificación de fuentes de conocimiento se definió un esquema basado en categorías y tipos de fuentes. Este esquema se ha dejado abierto para permitir la definición de categorías y tipos según las necesidades de cada estudio particular. Sin embargo, en un primer nivel, la clasificación de fuentes de conocimiento debe considerar categorías con respecto a la naturaleza de la fuente, por ejemplo, personas, documentos, sistemas de información, herramientas de apoyo a los procesos de trabajo, productos generados durante los procesos de trabajo, etc. Posteriormente, en cada categoría se pueden definir tipos y subtipos dependiendo del nivel de detalle al que se quiera llegar. Por ejemplo, en algunos casos podrá ser suficiente con definir un tipo para los empleados de una organización, mientras que en otros sea necesario separarlos en departamentos, roles, o jerarquías.

El esquema de clasificación se ha integrado en el metamodelo para la descripción de fuentes de conocimiento que ha sido la base para la definición de la estructura de clasificación y definición de dichas fuentes. La Figura 11 muestra la parte del metamodelo relacionada con las fuentes de conocimiento. En particular, cada fuente deriva del concepto base para los tipos y fuentes de conocimiento, denominado KConcept. Cada uno de estos

conceptos está organizado por medio de elementos de organización (KSourceOrganizationElement), que en el caso de las fuentes son categorías (KSourceCategory), y tipos (KSourceKind). Asimismo, cada fuente tiene una o más localizaciones, que definen el lugar donde la fuente puede ser localizada. Para cada localización se define su tipo (e. j. localidad física, electrónica, teléfono, email, etc.), y se asocia a un tipo de formato y soporte físico en caso de que aplique. Finalmente, cada soporte físico puede definir un conjunto de formatos posibles.

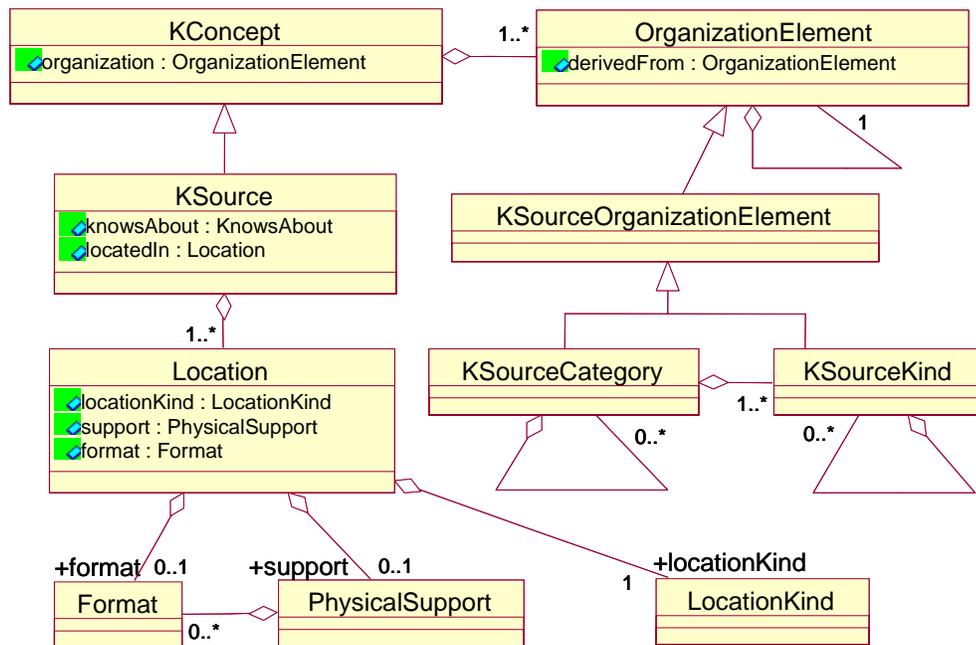


Figura 11. Memamodelo de fuentes de conocimiento.

La Figura 12 muestra un esquema que ejemplifica el uso del metamodelo para definir la estructura de clasificación de las fuentes del escenario ejemplo. Como se observa, se han definido cuatro categorías que se extrajeron durante la identificación de fuentes en la descripción del escenario que hemos estado analizando: documentos, personas, sistemas de información y herramientas de apoyo. Dentro de cada categoría se han definido tipos para agrupar las fuentes específicas descritas en la sección anterior. Por ejemplo, se ha definido una categoría para los documentos, dentro de la cual se tienen documentos de sistema que a

su vez pueden corresponder con planes de proyecto o especificaciones de requerimientos de los sistemas desarrollados o en desarrollo.

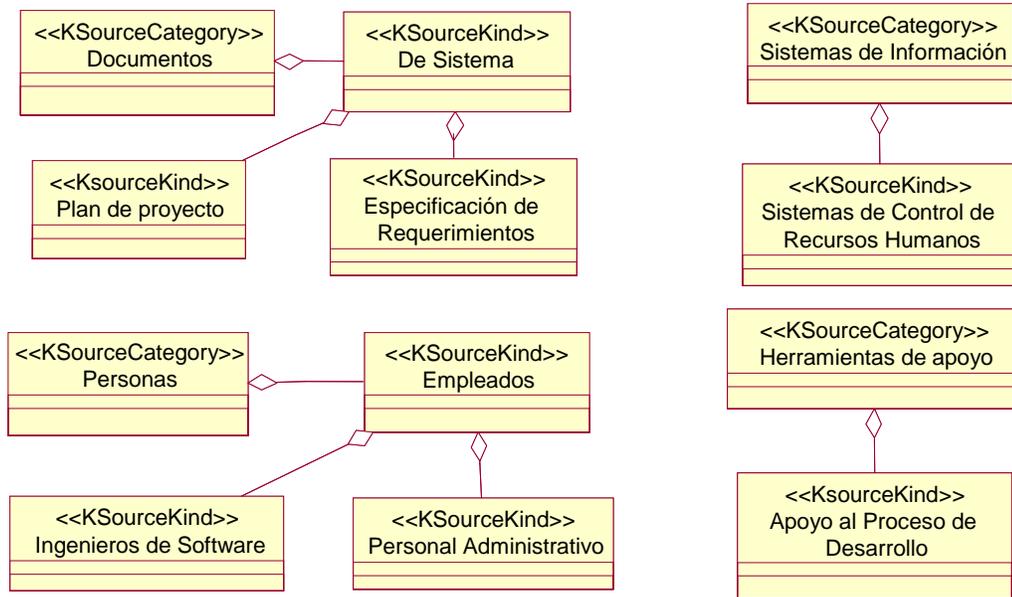


Figura 12. Ejemplo del uso del metamodelo de fuentes de conocimiento. Presenta un esquema de clasificación para el escenario ejemplo.

El metamodelo aquí descrito puede ser tomado como base para la definición tanto de un esquema de clasificación de fuentes, como para la definición de la estructura de una ontología de fuentes de conocimiento. La identificación de los elementos definidos en el metamodelo es un paso importante en la definición de cada fuente específica, para posteriormente integrar estas fuentes como parte de la base de conocimientos. A continuación se presenta un esquema que apoya en la definición de fuentes de conocimiento específicas.

III.4.2.2 Descripción de fuentes de conocimiento

La plantilla mostrada en la Tabla II se ha definido para facilitar la descripción de fuentes de conocimiento con base en la información especificada en el metamodelo. En ella se captura información para identificar las fuentes, su localización y el tipo de

conocimiento que puede ser obtenido de ellas. La tabla presenta información sobre una fuente hipotética, extraída del escenario ejemplo. A continuación se describe cada uno de los campos de la plantilla.

Tabla II. Plantilla para la definición de fuentes de conocimiento. Muestra información sobre un tipo de fuente hipotética extraída del escenario ejemplo.

Nombre (id):	Plan de proyecto		
Tipo:	De sistema		
Categoría:	Documentación		
Descripción:	Tipo de documento que contiene información sobre los proyectos a desarrollar		
Localización:			
Tipo	Descripción	Soporte físico	Formato
Electrónica	Los documentos de este tipo se encuentran almacenados una carpeta compartida en la máquina personal del responsable de proyecto. También están accesibles a través del portal organizacional bajo la categoría de Departamento de Desarrollo->Proyectos	No aplica	Word
Conoce a cerca de:			
Concepto		Nivel (Experto, avanzado, medio, etc)	
Documentos asociados		Avanzado	
Costo del proyecto		Avanzado	
Tiempo de duración del proyecto		Avanzado	
Actividades a realizar		Avanzado	
Pesonal asignado		Medio	

Nombre. El valor de este campo es el que identifica a la fuente de información en cuestión, por lo que debe ser único (Ej. Plan de mantenimiento del sistema X).

Tipo. Este campo indica el tipo de fuente de información o conocimiento.

Categoría. Indica la categoría a la que pertenece la fuente.

Descripción. Sirve para dar una descripción que permita conocer más sobre la fuente de conocimiento.

Localización. Indica las distintas formas en que puede ser localizada la fuente de conocimiento.

- **Tipo.** Indica el tipo de localización, por ejemplo: correo electrónico, localización física, teléfono, dirección electrónica, base de datos, algún repositorio de documentos, etc.

- **Descripción.** Indica la forma específica para localizar la fuente según el tipo de que se trate. Por ejemplo, la dirección de correo, la dirección física, el número de teléfono, el url, la base de datos, etc.
- **Soporte físico.** Este campo indica el elemento físico en el que se encuentra la fuente de información (Ej. Papel). Pueden darse casos en que una misma fuente cuente con más de un soporte físico, por ejemplo, que se tenga un documento con una copia en papel y otra en formato electrónico. Algunos de los tipos de soporte que pueden considerarse son: no aplica (para las personas este campo no aplicaría), papel, electrónico, audio, vídeo, etc.
- **Formato.** Este campo se puede emplear para especificar un tipo de formato específico para el tipo de soporte físico. Ejemplos de esto podrían ser los siguientes: electrónico (Documento Word, Excel, PowerPoint, PDF, PostScript, HTML, etc.), fotografía (Papel fotográfico, microfilm), audio (analógico o digital; WAV, MP3, WMA, etc.), vídeo (analógico o digital; VHS, SuperVHS; MPEG, AVI, etc.).

Conoce a cerca de. Indica los principales conocimientos que pueden ser obtenidos por medio de ésta fuente.

- **Concepto.** Nombre del concepto sobre el que la fuente tiene información. Por ejemplo, para indicar que en la fuente se puede encontrar información sobre el proceso de solicitud de modificaciones a un sistema, el nombre del concepto podría ser: Proceso de solicitud de modificaciones.
- **Nivel.** Indica el nivel de conocimiento que puede ser obtenido por medio de la fuente. La definición de estos niveles debe realizarse con base en las necesidades de la organización donde se vaya aplicar. Por ejemplo, en el caso de las personas, Wiig [1993] (referenciado en [Dalkir, 2005: p. 64]) ha propuesto un modelo de 5 grados de experiencia: 1) novato, se refiere a personas que no están concientes del conocimiento que tienen a mano, y cómo puede ser usado; 2) principiante, sabe que el conocimiento existe y dónde obtenerlo, pero no puede razonar con el; 3) competente, sabe a cerca del conocimiento, puede usarlo y razonar a cerca del conocimiento obteniendo ayuda de bases de conocimiento externas, como documentos y personas; 4) experto, sabe a cerca del conocimiento, lo contiene en su memoria, entiende dónde aplica, razona a cerca del mismo sin necesidad de ayuda externa; 5) maestro, internaliza totalmente el

conocimiento, tiene un entendimiento profundo con una completa integración con el valor, juicios, y consecuencias de usar ese conocimiento. Con respecto a las fuentes de conocimiento explícito, podrían clasificarse dependiendo de la cantidad de tópicos de conocimiento que cubren sobre un tema específico, o por ejemplo, dependiendo de si el conocimiento es tópico (descripciones; saber qué), episódico (ejemplos de situaciones similares; saber qué, cuándo, dónde), procedural (descripción de procedimientos, técnicas, etc.; saber cómo), o causal (saber por qué).

La plantilla para la definición de fuentes de conocimiento, permite asociar estas fuentes con los principales conocimientos o información que puede ser obtenida de ellas. En una primera iteración, los temas de conocimiento definidos pueden ser genéricos. Estos temas pueden después clasificarse, y conforme se avance en la clasificación de los distintos tipos de conocimiento, se pueden modificar las plantillas para reflejar estos cambios. Finalmente, es importante que las taxonomías de fuentes y tipos de conocimiento estén alineadas con las actividades del proceso estudiado [Rao, 2005b: p. 57], por lo que deberán de irse refinando conforme se avance en el análisis del proceso. Es importante tener en cuenta que las plantillas pueden ser usadas tanto para definir fuentes concretas como tipos de estas, como en el caso del ejemplo mostrado.

III.4.3 Identificación de tipos de conocimiento

La segunda etapa de la fase de análisis de la metodología es la identificación del conocimiento involucrado dentro del grupo estudiado. En esta etapa también se hace uso de taxonomías para definir un esquema para clasificar los distintos tipos de conocimiento, y de ontologías para definir la manera de describir los distintos tipos de conocimiento (en el Apéndice E se presenta un ejemplo de una taxonomía y ontología de tipos de conocimiento). Entre los aspectos que deben ser considerados en esta etapa se encuentran:

- 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.

Rao [2005b: p. 36] hace una distinción de una serie de atributos que hay que considerar con respecto al análisis del conocimiento que se quiere administrar en una organización. Entre los principales aspectos se encuentran:

- Complejidad. Se refiere a la complejidad en el manejo del tipo de conocimiento, por ejemplo, si es tácito es más difícil de manejar que si es explícito. A su vez, el conocimiento explícito puede tener diversos grados de complejidad dependiendo de que tan estructurado es [Valente y Housel, 2002].
- Dominio. Se refiere al dominio del conocimiento, por ejemplo, si es sobre tecnología, aspectos organizacionales, etc.
- Enfoque. Referente a si el enfoque del conocimiento es estratégico u operacional.
- Tiempo de relevancia. Referente al tiempo en que el conocimiento pierde su utilidad, por ejemplo, inmediato, o de mediano o largo plazo.
- Granularidad. Indica que tan atomizado está el conocimiento, por ejemplo, si es muy general (amplio) o muy específico (fino).
- Fuente. Se refiere al tipo de fuente del conocimiento, por ejemplo, personas (individuo, grupo, organización, dominio público), procesos, repositorios, etc.
- Medio de transmisión. Referente al medio por el cual se transmite el conocimiento, tales como oral, escrito, texto, gráficos, multimedia, digital, etc.
- Audiencia. Relaciona al tamaño de la audiencia a la que va dirigido el conocimiento. Uno a uno, uno a muchos, muchos a uno, muchos a muchos.
- Importancia. Referente a la importancia del conocimiento (relevante, útil, crítico, indispensable, único, irrelevante).

Por otro lado, los temas de conocimiento pueden ser vistos como unidades cognitivas siguiendo cinco propiedades:

- *coherencia*, que indica que todos los componentes del tema surgen de una idea central y claramente identificada;
- *precisión*, que indica que el tema está descrito claramente y sin ambigüedad;
- *puede ser examinado*, indica que el tema incluye elementos que facilitan identificar cuando una persona conoce el concepto;
- *generalizable*, que los tópicos incluidos son de interés general; y
- *especificidad*, el alcance del tema es lo suficientemente pequeño para ser cubierto en una o pocas clases de un curso o tutorial, en un capítulo de un libro o secciones de un documento, etc.

A su vez, es importante considerar algunos componentes para cada tema de conocimiento [Meyer, 2006], en nuestro caso consideramos que los más relevantes son:

- Nombre;
- Nombres alternativos;
- Dependencias: otros temas que requieran ser conocidos con anterioridad;
- Resumen: una descripción corta del tema.

Siguiendo el escenario ejemplo, los temas corresponderían a aquella información o conocimiento que se requiere para desarrollar el plan del proyecto, incluyendo: la información que éste debe contener, el formato del documento, las fuentes de dónde se puede obtener la información que se captura en el proyecto, métodos para calcular el tiempo o costo que requerirá el proyecto, información para definir las actividades a realizar, información sobre el personal que podría participar en el proyecto, etc.

Conforme se identifican los principales temas de conocimiento requeridos en las actividades más relevantes del proceso, así como las fuentes de donde pueden ser

obtenidos, deben clasificarse y describirse, para posteriormente analizar cómo estos temas de conocimiento y sus fuentes intervienen a lo largo del proceso.

III.4.3.1 Clasificación de tipos de conocimiento

Para la clasificación de tipos de conocimiento hemos definido un esquema basado en la estructura utilizada en [Hilburn *et al.*, 1999a]. La Figura 13 presenta la integración de este esquema al metamodelo de conceptos de conocimiento que ha sido definido. Cada uno de los distintos elementos de la estructura se ha definido como tópicos de conocimiento (KTopic). La estructura de clasificación separa los niveles de abstracción en categorías de conocimientos (KCategory), las cuales están compuestas por áreas de conocimiento (KArea), que a su vez pueden contener sub-áreas y/o temas (KSubject), mismos que pueden estar constituidos de sub-temas o temas más especializados.

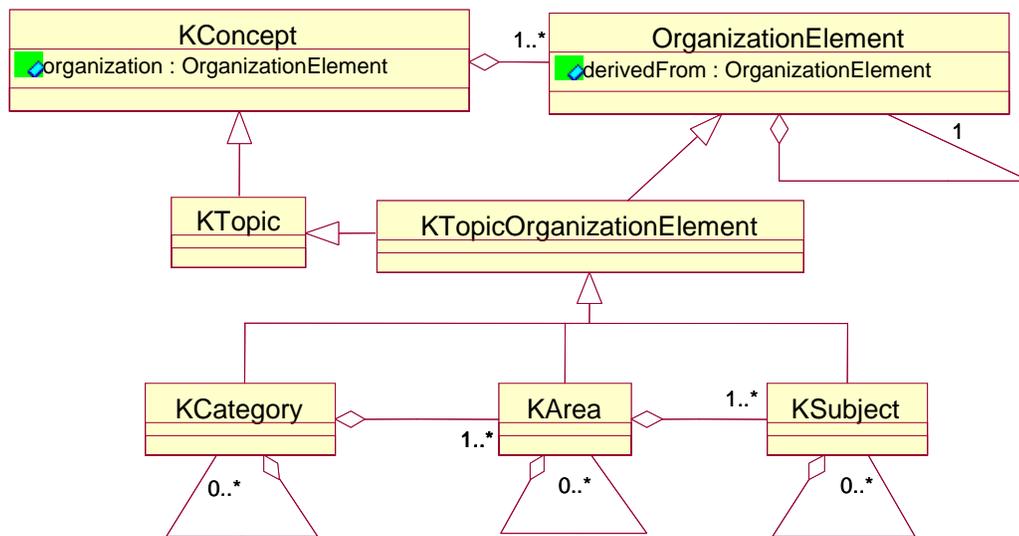


Figura 13. Niveles de abstracción propuestos para clasificar tipos de conocimiento.

Las *categorías de conocimiento* son elementos estructurales de alto nivel de abstracción, usadas para organizar y clasificar áreas de conocimiento relacionadas de alguna manera. Las *áreas de conocimiento* son subdivisiones de las categorías que están lógicamente relacionadas con la categoría, ya sea por herencia, agregación o composición,

es decir, que sean una subdivisión de la categoría o parte de los componentes de la categoría. Las áreas de conocimiento se pueden constituir a su vez de áreas más especializadas, o de temas de conocimiento específico. 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 específicos.

Las áreas de conocimiento donde los temas son clasificados, al igual que los temas, deben estar claramente identificadas por medio de un nombre, así como la identificación de nombres alternativos; una descripción, y la identificación de otras áreas que pudieran estar relacionadas, por ejemplo, que correspondan a conocimientos que son requeridos para comprender el área en cuestión.

Un aspecto importante a considerar en esta etapa es el nivel de abstracción al que se llegará en la clasificación y detallado de los distintos tópicos de conocimiento que serán considerados. La forma en que se utilice la estructura de clasificación para los tipos de conocimientos específicos de una organización depende del detalle y nivel de abstracción que se requieran. Por ejemplo, supongamos que una organización define un área que corresponde a lenguajes de programación y con el lenguaje C++ como tema. Si queremos llegar a un nivel más bajo de abstracción, podríamos definir como temas de conocimiento más específicos: la sintaxis del lenguaje C++, su semántica o las librerías que proporciona.

Continuando con el escenario, supongamos que se identifica que dentro de la organización existen métodos predefinidos para la estimación de tiempos y costos de los proyectos y se decide clasificar dichos métodos. Dado que éstos corresponden a métodos utilizados para la realización de actividades requeridas en el proceso, se crea una categoría y una secuencia de áreas dentro de la misma hasta llegar a los métodos de estimación de costos y tiempos. Sin embargo, debido a que ambos tipos de métodos se usan dentro de la actividad que corresponde con el desarrollo del plan de proyecto, se decide también crear

un área para agrupar los conocimientos relacionados con dicha actividad. De esta manera, los métodos de estimación quedarían dentro de dos áreas, la que corresponde con métodos de estimación, y la del área de desarrollo del plan del proyecto. La Figura 14 presenta un ejemplo de la forma en que el metamodelo puede ser usado para representar este escenario.

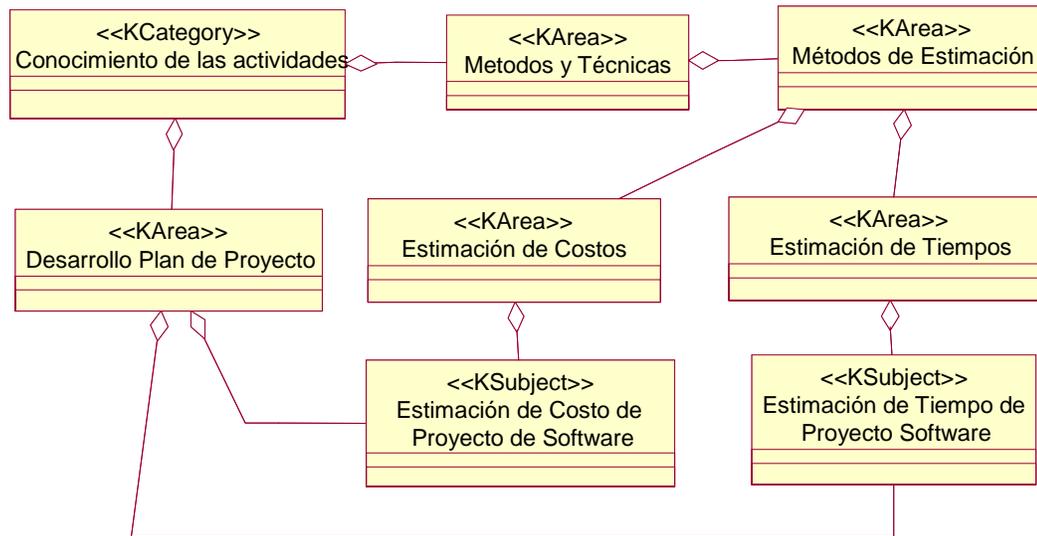


Figura 14. Ejemplo de clasificación de temas de conocimiento. Presenta un esquema de clasificación para un caso en el escenario ejemplo.

III.4.3.2 Descripción de tipos de conocimiento

Para la descripción de tipos de conocimiento hemos tomado en cuenta, tanto el enfoque propuesto por Meyer [2006] presentado anteriormente en este capítulo, como el modelo del flujo del conocimiento presentado en el Capítulo II, sección II.1.1. Este modelo describe cómo el conocimiento tópico utilizado durante la realización de las actividades, genera conocimiento episódico, y la aplicación reiterada de éstos genera a su vez conocimiento técnico o procedural. Es decir, que para realizar alguna actividad es necesario saber qué se debe hacer, cuándo, dónde, con qué, por qué, etc. El manejo del conocimiento desde esta perspectiva, puede ser tomado como un indicador del nivel del manejo de conocimiento, siendo el saber qué el primero, seguido del saber cómo y el saber por qué [Tiwana, 2000: p. 73]. Cada vez que se aplican conocimientos tópicos obtenemos

experiencias de la manera en que los conceptos se pueden aplicar para resolver algún caso práctico. Finalmente, la aplicación reiterada de estos conceptos en casos prácticos genera la experiencia del saber cómo adaptar los conocimientos adquiridos para resolver casos similares. Asimismo, existen fuentes donde estos conocimientos pueden ser obtenidos. Estas fuentes pueden ser desde documentos hasta la misma experiencia de las personas encargadas de realizar las actividades. Siguiendo este modelo conceptual es posible identificar los componentes cognoscitivo y técnico de los distintos tipos de conocimientos involucrados en las actividades realizadas por el grupo, así como el existente en las distintas fuentes de conocimiento.

En la Tabla III se presenta una plantilla que puede ser empleada para la captura de esta información. Para ejemplificar su uso tomaremos el caso de la estimación de tiempos de un proyecto del escenario anterior. En este caso, se podría identificar que para la estimación de tiempos se requiere saber qué actividades se realizará, así como información sobre quien las realizará, por ejemplo la experiencia en la realización de ese tipo de actividades, si tiene otras tareas distintas a las asignadas, etc. Así mismo, se requiere conocer sobre los riesgos que podrían influir en el tiempo del proyecto, la frecuencia de los mismos y el tiempo probable que requerirían en caso de presentarse. Finalmente, se necesitaría conocer sobre el procedimiento para estimar el tiempo. Con el fin de clarificar todos los temas relacionados con la estimación del tiempo, se decide emplear la plantilla para especificar los qué (tópico) y los cómo (procedural) de dicho tema, así también los momentos que podrían generar experiencia o conocimientos episódicos sobre el tema.

Los campos usados en la plantilla se describen a continuación.

Nombre. Este campo identifica al concepto de conocimiento. Este nombre debe ser único para cada concepto.

Nombres alternativos. Lista los nombres alternativos existentes para referirse al mismo concepto de conocimiento.

Tabla III. Plantilla para de descripción de temas de conocimiento. La información mostrada corresponde con un caso hipotético extraído del escenario ejemplo.

Nombre:	Estimación de tiempos de proyectos de software	
Nombres alternativos:	Cálculo de tiempo de proyectos de software	
Tipo:	Métodos de estimación de tiempos	
Clasificación:	Conocimiento de las actividades/Métodos de estimación	
Casificaciones secundarias:	Conocimiento de las actividades/Desarrollo de plan de proyecto	
Descripción:	Se refiere al conocimiento relacionado con el método de estimación de tiempos para los proyectos de desarrollo y mantenimiento de software	
Descomposición del tema		
Tópico:	¿Qué información se requiere para estimar el tiempo del proyecto? ¿Dónde se puede obtener dicha información? ¿Qué problemas podrían afectar el tiempo del proyecto? ¿Cuál es el procedimiento para estimar el tiempo?	
Episódico:	Estimación del tiempo en proyectos previos Apoyo en la estimación del tiempo en proyectos previos	
Procedural:	¿Cómo organizar la información para la estimación del tiempo? ¿Cómo calcular el tiempo requerido de presentarse problemas? ¿Cómo calcular el tiempo de cada actividad? ¿Cómo aplicar el método de estimación de tiempo?	
Afecta en:	Proceso/Actividad	Descripción (requerido, generado)
	Planeación del proyecto	Requerido
Localizado en:	Nombre de fuente	Grado de conocimiento (experto, avanzado, medio, bajo)
	Responsable del proyecto	Avanzado
Conocimiento relacionado	Nombre del concepto	Tipo de relación (requerido, agregado, complemento)
	Estimación de riesgos	Requerido

Tipo. Especifica el tipo de concepto de conocimiento (categoría, área, tema). Es decir, el elemento de la ontología al que corresponde el concepto que se define.

Clasificación. En este campo se define la ruta dentro de la estructura de áreas de conocimiento a la que pertenece el concepto de conocimiento definido.

Clasificaciones secundarias. En ocasiones, un tema puede pertenecer a más de un área o categoría de conocimiento. Para estos casos, en este campo se pueden listar todas aquellas áreas donde el tema también podría estar incluido.

Descripción. Si el nombre no es suficiente para expresar claramente el conocimiento asociado al concepto, se puede utilizar este campo para tal efecto.

Descomposición del tema. Aquí se describen los componentes tópico, episódico y procedural del concepto de conocimiento que se está describiendo.

- **Tópico.** Dentro de este campo es donde se describe el componente declarativo-tópico del concepto que se está describiendo. Los conceptos principales involucrados, por ejemplo, el saber qué o con qué.
- **Episódico.** Este campo es donde se define el componente declarativo-episódico del concepto. Es decir, las distintas situaciones que pueden generar experiencias en la aplicación del componente tópico del conocimiento. Es algo parecido al saber cuándo o por qué es aplicado el conocimiento. Mediante la definición de este campo se pueden determinar las situaciones que pueden generar experiencia en la aplicación del conocimiento que se define.
- **Procedural.** Define el componente técnico o procedural asociado al concepto. En este campo se especifica el conocimiento requerido para saber cómo aplicar el concepto.

Afecta en. Dentro de este campo se definen los procesos y/o actividades donde el concepto de conocimiento está involucrado, así como la forma en que se relaciona el concepto con estas actividades o procesos.

- **Proceso/Actividad.** Define la actividad o proceso donde es utilizado el concepto.
- **Descripción.** Una descripción de la manera en que se relaciona el concepto dentro de la actividad o proceso. Este campo puede indicar si el conocimiento que se está definiendo es creado, utilizado, incrementado, requerido, etc. por la actividad o proceso en cuestión.

Localizado en. Dentro de este campo se definen las fuentes de información donde se puede obtener conocimiento sobre el concepto.

- **Nombre de fuente.** Indica el nombre de la fuente de información.
- **Grado de conocimiento.** Indica el grado de conocimiento, sobre el concepto, que puede ser obtenido en la fuente (por ejemplo: experto, avanzado, intermedio, básico).

Conocimiento relacionado. Sirve para identificar qué otros conceptos de conocimiento puedan estar relacionados con el descrito. Por ejemplo, qué conocimientos se requieren para poder aplicar el que se está describiendo, o en qué otros conceptos se requiere conocer a cerca del que se está describiendo.

- **Nombre del concepto.** El nombre del concepto con el que se relaciona el descrito.

- **Tipo de relación.** Indica el tipo de relación que existe entre los conceptos. Ejemplos de tipos de relaciones que podría existir son las de dependencia (el conocimiento de un concepto depende del conocimiento que se tenga sobre el otro), de agregación (que un concepto de conocimiento es parte o componente de otro más amplio), de clasificación (que un determinado concepto de conocimiento pertenece a una determinada clase), complemento (que el conocimiento de uno de los conceptos puede complementar el que se tenga del otro), etc.

La plantilla facilita la identificación de las asociaciones de los temas de conocimiento con las principales actividades o procesos con los que están relacionados. En una primera iteración, las actividades o procesos definidos pueden ser descritos en términos generales. Posteriormente se pueden ir especificando a mayor detalle, y conforme se avance en la definición y modelado de los flujos de conocimiento, se pueden modificar las plantillas para reflejar mejor los procesos, actividades, tareas, etc., específicos con los que se relaciona el concepto de conocimiento.

III.4.4 Identificación de flujos de conocimiento

La tercera etapa de la fase de análisis de la metodología KoFI se centra en identificar la manera en que los tipos y fuentes de conocimiento interactúan dentro de los distintos procesos y actividades que debe realizar y decisiones que debe tomar el grupo u organización que se estudia. Esto se hace con el fin de identificar el conocimiento requerido por dichas actividades, procesos o decisiones, así como las fuentes y mecanismos de que se valen los encargados de las mismas para obtener, guardar y compartir ese conocimiento.

Para la identificación de los flujos de conocimiento, se identifica primero el conocimiento que es aplicado o requerido durante la realización de actividades, toma de decisiones, etc. Se identifican las fuentes donde este conocimiento puede ser obtenido, así como los mecanismos utilizados para localizar y recuperar esas fuentes. También, se identifica el conocimiento generado durante estas actividades o toma de decisiones, si éste

conocimiento es almacenado en algún lugar, y de ser así, qué mecanismos se utilizan para estructurarlo, representarlo y almacenarlo.

Siguiendo con el escenario, podríamos identificar que durante la estimación de tiempos, el responsable del proyecto requiere obtener información sobre las personas que serán asignadas al mismo, tales como su carga de trabajo actual y futura, su experiencia, etc. De esta manera el responsable del proyecto estará en posición de asignar las actividades a las personas más adecuadas con el fin de buscar minimizar el tiempo que el proyecto podría consumir. En este caso sería relevante identificar la forma en que dicha información fluye hacia el responsable del proyecto. Podría darse el caso que existiera un sistema de información donde obtener parte de dicha información, pero podría también ser necesario consultar con todos los ingenieros de software para conocer cuál es su carga de trabajo y su experiencia en cada una de las actividades que requerirá el proyecto. En ambos casos el flujo de información es distinto tanto en forma como en el tiempo que podría requerirse para que toda la información sea obtenida.

Para una mayor clarificación, la identificación de los flujos de conocimiento se hace mediante el modelado de los mismos. Es importante que la técnica de modelado utilizada disponga de elementos que permitan representar de manera explícita el conocimiento involucrado en el proceso, al igual que las fuentes del mismo. Los modelos de procesos, además de permitir el entendimiento de dichos procesos, pueden servir de apoyo en la identificación de problemas, y con base en estos, la identificación de requerimientos para definir estrategias, o diseñar sistemas encaminados a resolver dichos problemas [Cox et al., 2005b]. Lo cual es la siguiente etapa en la fase de análisis de la metodología.

III.4.5 Identificación de problemas en el flujo del conocimiento

La cuarta y última etapa del análisis del proceso se centra en identificar problemas específicos que pueden afectar el flujo del conocimiento. El objeto de la identificación de estos problemas es buscar formas de resolverlos, por lo que la etapa también considera la proposición de posibles soluciones a los problemas que sean identificados.

En esta etapa se hace uso de *escenarios de problemas*, los cuales son una combinación de los conceptos definidos por las técnicas de marcos de problemas (problem frames) [Cox *et al.*, 2005a; Jackson, 2005], y la técnica de escenarios [Carroll y Rosson, 1992; Chin *et al.*, 1997]. Estas dos técnicas han sido definidas para la obtención de requerimientos para el diseño de sistemas de software. Los marcos de problemas buscan centrarse en el entendimiento del problema y su contexto más que en la solución. Es un marco conceptual para enfocarse en los grupos importantes de tipos de problemas, con el fin de desarrollar buenas soluciones [Jackson, 2005]. Por su parte, un escenario es una descripción textual de algún suceso o situación, una especie de historia sobre las actividades que una persona debe realizar dentro de un determinado contexto [Carroll y Rosson, 1992]. Tomando ambos conceptos, hemos definido un *escenario de problema* como una descripción textual de un tipo de problema particular, una historia que narre el problema de forma que se identifique su contexto. Una ventaja del uso de escenarios es que pueden ser usados para facilitar la participación de los usuarios durante la especificación de requerimientos [Chin *et al.*, 1997].

Los escenarios de problema se obtienen del análisis de los modelos de los flujos de conocimiento, buscando identificar los problemas que puedan afectar dichos flujos. Cada escenario de problema es especificado definiendo un nombre descriptivo del problema específico, el tipo de problema en el que es clasificado, la descripción textual del escenario, y finalmente, una o varias alternativas de solución. Las alternativas de solución son descritas como un escenario alternativo que ejemplifique la manera en que la solución propuesta modificaría el escenario de problema descrito. Estas alternativas son empleadas posteriormente para obtener los requerimientos que deberá cubrir un sistema enfocado a atacar los problemas identificados.

Como ejemplo, y continuando con el escenario, supongamos que existe un sistema donde se capturan los proyectos, el cual permite definir las actividades del proyecto y los responsables de las mismas. Una vez que se han definido dichas actividades, el sistema envía una copia de la información de las mismas a los ingenieros responsables de realizarlas, de forma que ellos puedan llevar un registro de sus actividades realizadas y

pendientes. Aun cuando los ingenieros pueden modificar dichas actividades, por ejemplo indicando cuáles fueron concluidas y el tiempo que les llevó realizarlas, dicha información no se refleja en el sistema principal, dado que no existe una conexión entre las copias locales a cada ingeniero, y la del sistema. Por lo tanto, aun cuando el responsable del proyecto utilizara el sistema para conocer los ingenieros que tienen otras actividades asignadas, tendrá que consultar con los mismos para poder actualizar la información que el sistema proporciona; lo que consume tiempo a la hora de definir los planes de proyecto.

Después de observar este problema, se podría proponer modificar el sistema de forma tal que las copias de las tareas locales estuvieran conectadas con el sistema central. De esta manera, el sistema mantendrá información más actualizada sobre las actividades pendientes de cada ingeniero, facilitándole al responsable del proyecto la asignación de personal. Adicionalmente, tener un estimado del tiempo que le lleva a un determinado ingeniero terminar cierto tipo de actividades, podría ayudar al momento de estimar los tiempos del proyecto. Incluso, podría darse el caso de que dicha información fuera utilizada para realizar una estimación preliminar de forma automatizada, tomando en cuenta la información existente en el sistema, facilitándole así aún más la tarea al responsable del sistema. De esta forma, si la organización decidiera realizar estas modificaciones, podría utilizar la información obtenida a lo largo de los pasos anteriores con el fin de conocer la información que debe almacenarse en el sistema para facilitar la estimación de tiempos, actividades y asignación de personal de los proyectos.

En la Tabla IV se presenta un formato que puede ser utilizado para la descripción de escenarios de problemas. Este formato describe el problema anteriormente descrito, así como la solución que se propuso.

Tabla IV. Ejemplo de descripción de un escenario de problema.

Problema:	Inconsistencia en la información de las actividades realizadas
Tipo:	Pérdida de información
Descripción del escenario:	
José, el responsable de uno de los proyectos de la organización, tiene que definir las actividades del proyecto y estimar el tiempo que llevarán, con el fin de determinar el tiempo total requerido por el proyecto a su cargo. José sabe que el tiempo de cada actividad depende en parte de la persona que esté asignada para realizarla. Debido a esto, antes de hacer la estimación decide hacer una asignación de las mismas entre los ingenieros a cargo del proyecto. José realiza la asignación tomando como base su conocimiento de la experiencia de cada ingeniero, con el fin de reducir el tiempo que llevará cada actividad asignando a cada actividad al ingeniero más experimentado para realizarla. Una vez hecha la estimación, José consulta con las personas asignadas a cada actividad con el fin de determinar en conjunto el tiempo que éstas llevarán. Cuando José se encuentra a mitad de dicha estimación, al llamar a uno de los ingenieros éste le hace ver que ya tiene otras actividades asignadas en otro proyecto que le impedirán dedicarse al ciento por ciento, lo que hace que Juan deba reasignar las actividades y volver a iniciar todo el proceso.	
Escenario alternativo:	
Cuando José inicia la definición de las actividades, utiliza el sistema de gestión de proyectos para identificar actividades similares y buscar determinar el tiempo promedio que llevan las mismas. Si este tiempo es similar independientemente de quien realice la actividad, José podría suponer que en esta nueva ocasión también sería similar. Si no es este el caso, entonces podría analizar si la diferencia se debe al ingeniero a cargo, y consultar con estos para determinar cuál podría ser la causa de la variación. De esta manera José podría iniciar una estimación que pueda ser independiente de los ingenieros que fueran asignados a cada actividad. Adicionalmente, una vez que José decide asignar las actividades a los ingenieros a cargo del proyecto, utiliza el sistema para identificar qué ingenieros tienen otras actividades pendientes con el fin de considerar el tiempo que éstas llevarán, para hacer una asignación de actividades que se ajuste lo más posible a dicha situación.	

En este punto de la metodología podrían surgir dos casos: primero, la decisión de desarrollar o adquirir un nuevo sistema para abordar los problemas detectados; y segundo, la de utilizar los existentes, por ejemplo modificándolos. En el primero caso, la metodología podría dejarse de aplicar en este punto y centrarse en definir los requerimientos que deberá tener el nuevo sistema. Esto último tomando como base la información obtenida del análisis, y sobre todo de los escenarios de problemas y soluciones. Sin embargo, debemos tener en cuenta que una propuesta de AC no solamente debe centrarse en el desarrollo o adquisición de nuevas herramientas. Para lograr estrategias de AC exitosas, es importante considerar también la manera en que la infraestructura tecnológica de apoyo al proceso actual puede ser integrada a dicha estrategia [Davenport *et al.*, 1998; Jennex y Olfman, 2005].

El escenario descrito en esta sección es un ejemplo de lo anterior, en el que se decide por modificar un sistema con el fin de facilitar la tarea de estimación de tiempos y

asignación de personal a los proyectos. Los requerimientos para dicha modificación pueden ser extraídos de las descripciones de problemas y soluciones, así como de los modelos de fuentes y conocimientos que sean definidos a lo largo del proceso de análisis (por ejemplo para identificar el tipo de información relacionada con la estimación de tiempos, las actividades donde se genera, las fuentes de donde se obtiene, etc.). No obstante, antes de empezar a modificar dicho sistema para incluir las nuevas opciones, es necesario primero conocer aspectos tales como el tipo de información que el sistema permite manejar, los procesos que apoya, los usuarios potenciales, la estructura que guarda la información manejada en el mismo, las actividades de AC que actualmente apoya y las que no, etc. Lo anterior con el fin de ayudar a identificar las implicaciones y el efecto que podría tener la modificación del sistema en el proceso. Para facilitar esto último, la fase final de la metodología consiste en un marco que ayuda a identificar el rol que los sistemas o herramientas usadas en el proceso estudiado juegan dentro del flujo del conocimiento. De esta manera, al proponer modificaciones a dichos sistemas se podría observar la forma en que éstas afectarían dicho rol.

III.5 Análisis de herramientas de soporte al flujo del conocimiento

Con frecuencia, las organizaciones suelen tener algún nivel de AC aun cuando estas no apliquen estrategias de AC de forma explícita. Por ejemplo, existen estudios que han observado este hecho en organizaciones de desarrollo de software [Meehan y Richardson, 2002; Ward y Aurum, 2004], y en general, lo anterior lo podríamos esperar en casi cualquier organización cuyos procesos estén altamente basados en conocimiento, como lo son las compañías tecnológicas [Bontis *et al.*, 2003], o el sector salud [Curia *et al.*, 2005]. En este tipo de organizaciones, la infraestructura tecnológica que apoya los procesos de trabajo juega un papel importante como habilitadora de la AC. Incluso, muchas herramientas y tecnologías que nacieron dentro del área de los sistemas de información, y que son comúnmente usadas por las organizaciones, han sido de gran apoyo como sistemas de apoyo a la AC [Davenport, 2007; Marwick, 2001]. Por otra parte, entre los factores a considerar para lograr el éxito de las estrategias de AC está la inclusión de los sistemas de

apoyo al trabajo diario de las organizaciones [Davenport et al., 1998; Jennex y Olfman, 2005; Maier y Remus, 2002; Stewart, 2002]. Como ya hemos mencionado, una de las necesidades principales de la AC es lograr la integración del conocimiento en los procesos de trabajo [Scholl *et al.*, 2004], siendo la integración de dicho conocimiento en las tecnologías usadas para realizar esos procesos la opción que parece ser más viable hasta el momento [Davenport, 2007].

Por lo anterior, como fase final de la metodología se propone estudiar las herramientas que apoyan el proceso a analizar, desde el punto de vista del rol que juegan dichas herramientas como facilitadoras del flujo del conocimiento durante el proceso. Los objetivos de este análisis incluyen los siguientes:

- 1) Identificar el rol que juegan las herramientas usadas durante el proceso, como facilitadoras del flujo del conocimiento.
- 2) Identificar requerimientos para mejorar el uso de dichas herramientas como habilitadoras del flujo del conocimiento.
- 3) Identificar los efectos que podrían causar en el flujo del conocimiento las modificaciones a dichas herramientas.
- 4) Identificar posibles usos de tales herramientas dentro de otros procesos o actividades.
- 5) Comparar distintas herramientas que apoyen actividades similares, desde el punto de vista de la contribución de las mismas a la AC.

Con el fin de facilitar el cumplimiento de los objetivos expuestos, como parte de la metodología se ha definido un marco de trabajo encaminado a servir como guía en el estudio de herramientas desde el punto de vista descrito. Este marco de trabajo consta de una serie de características que deben ser identificadas, incluyendo características que consideran aspectos como las actividades del proceso y las actividades de AC que son apoyadas por las herramientas. Debido a la extensión de dicho marco de trabajo, la descripción del mismo constituye uno de los capítulos de este documento, específicamente el Capítulo V.

III.6 Resumen del capítulo

En este capítulo se describieron las fases que componen la metodología propuesta en esta tesis. Esta metodología, denominada KoFI, se centra en el estudio de procesos organizacionales, y en particular procesos de ingeniería de software, desde un enfoque del flujo del conocimiento que se da en dichos procesos. El objetivo principal de la metodología, es ayudar en la identificación de posibilidades de mejora de dichos flujos, tomando en cuenta no sólo el desarrollo o adquisición de nuevas herramientas, sino también considerando la infraestructura tecnológica que soporta el proceso estudiado.

Como apoyo a la metodología, se han desarrollado una serie de técnicas, algunas de las cuales han sido descritas en este capítulo. Sin embargo, dos de estas propuestas son detalladas en los Capítulos IV y V; específicamente se presenta una propuesta para el modelado de procesos de ingeniería de software con enfoque en el flujo del conocimiento, y un marco de trabajo para el análisis de herramientas de soporte a dichos procesos desde el punto de vista del apoyo que dan a la AC. Finalmente, la metodología descrita ha sido validada por medio de su aplicación a un caso real, cuyos resultados son descritos en el Capítulo VI de este documento.

^v Malcolm Forbes fue editor de la revista Forbes después de la muerte de su padre, Bertie Charles Forbes, fundador de dicha revista.

CAPÍTULO IV

Modelado de procesos con enfoque en el flujo de conocimiento

“El análisis de procesos puede ser un buen punto de partida para diseñar e introducir sistemas de AC” [Maier y Remus, 2002]

Como parte de las técnicas que apoyan la metodología propuesta en esta tesis, se ha definido un enfoque de modelado de procesos para facilitar la identificación del conocimiento y las fuentes involucradas en dichos procesos, así como la forma en que este conocimiento y fuentes están envueltos en las actividades de los mismos. En particular, el propósito de este enfoque de modelado es que apoye en la identificación de la información que es requerida durante las etapas de análisis de la metodología KoFI. Este capítulo presenta el enfoque de modelado de procesos mencionado. El capítulo está compuesto por una revisión de literatura sobre enfoques de modelado de flujos de conocimiento, de donde se deriva la necesidad del enfoque propuesto en esta tesis. El enfoque propuesto consta de dos etapas: una de análisis preliminar del proceso con una técnica de modelado flexible, y otro de análisis detallado por medio de una técnica de modelado formal, ambas descritas en este capítulo.

IV.1 Modelado de flujos de conocimiento en procesos de ingeniería de software

Aun cuando la metodología propuesta fue desarrollada con el propósito de que sea de propósito general, debido a que nuestro interés son las organizaciones de desarrollo de software, las técnicas utilizadas en la metodología han sido definidas siguiendo este enfoque. Por lo tanto, para el lenguaje de modelado a utilizar, hemos hecho un análisis de las técnicas existentes que han sido aplicadas para estudiar organizaciones de desarrollo de software.

En la literatura sólo hemos encontrado unos cuantos trabajos que describen el uso de Lenguajes de Modelado de Procesos (LMP) para estudiar el conocimiento involucrado en procesos de software. Por lo general, estos estudios han utilizado lenguajes de propósito general para este tipo de estudios. Lo anterior puede deberse a la falta de enfoques de modelado de procesos que cuenten con elementos para representar de manera explícita el conocimiento dentro de los procesos [Bera *et al.*, 2005; Papavassiliou y Mentzas, 2003]. Por ejemplo, Hansen and Kautz [2004] han usado la técnica de Gráfica Rica para identificar flujos de conocimiento en procesos de software. La Gráfica Rica es un LMP flexible que puede ser fácilmente adaptado según las necesidades del modelador del proceso [Checkland y Scholes, 1999; Monk y Howard, 1998]. Sin embargo, es más fácil analizar detalles en modelos representados en un LMP formalmente definido y restringido, que en uno flexible y de propósito general [Curtis *et al.*, 1992].

De hecho, en una primera etapa del desarrollo de la metodología propuesta, empleamos la técnica de Gráfica Rica como lenguaje de modelado. Sin embargo, experimentamos algunos problemas. En particular, se nos dificultó la identificación y análisis de la estructura de las actividades y sub-actividades, la secuencia de las mismas, así como las dependencias entre actividades, conocimientos y fuentes. Debido a esto fue que nos decidimos por buscar otro enfoque de modelado para complementar la técnica de Gráfica Rica, la cual es una buena alternativa como técnica para hacer una representación inicial del proceso, pero que requiere ser complementada con técnicas más formales que faciliten el análisis de los detalles del proceso [Monk y Howard, 1998]. De hecho, dentro de la metodología de sistemas suaves donde se propone la técnica de Gráfica Rica, los autores proponen ésta como apoyo en la identificación de situaciones problemáticas generales en el proceso, e incluyen una etapa de construcción de modelos de actividades posterior al modelado de esta visión general del proceso, pero no definen una técnica formal para hacer esto último [Checkland y Scholes, 1999: p. A21].

Algunos de los enfoques de modelado formales utilizados para estudiar flujos de conocimiento en procesos de software hacen uso de lenguajes que no han sido diseñados para ese propósito. Por ejemplo, Woitsch y Karagiannis [2002] proponen una herramienta y

un enfoque de modelado centrado en facilitar el análisis de documentación y la implementación de sistemas de AC empresariales. Estos autores usan varios tipos de modelos para representar distintos elementos del proceso, como actividades, personas, etc. Los autores ejemplifican el uso de su propuesta por medio de un caso de estudio en una organización de desarrollo de software, presentando la manera en que ellos desarrollaron una memoria organizacional para dicha organización. Otro enfoque es el de Nissen y Levitt [2004], ellos usan una herramienta para diseñar grupos virtuales. Con esta herramienta, los autores hacen el análisis de flujos de conocimiento en un proceso de desarrollo de software. Los autores desarrollan este caso de estudio para ilustrar un modelo teórico del flujo de conocimiento que ellos proponen. Ninguno de los dos enfoques anteriores propone primitivas para la representación explícita de aspectos relacionados con el conocimiento en los procesos.

Por otra parte, Zhuge [2002] ha propuesto un enfoque exclusivamente para el análisis de flujos de conocimiento, el cual ha aplicado a un caso de estudio en una organización de desarrollo de software. En este enfoque, los procesos se descomponen en tareas y nodos de conocimiento, posteriormente se define el flujo de actividades y la secuencia de transferencia de conocimiento entre los nodos. Lo anterior se hace con base en las entradas y salidas de las tareas y los nodos de conocimiento (por ejemplo, conocimiento consumido o generado en un nodo). El enfoque propuesto por Zhuge, es sencillo, pero muy general en el sentido de que sólo propone un tipo de diagrama, el de actividades y nodos de conocimiento y sus conexiones a través de los flujos de conocimiento entre los mismos.

Otro trabajo relacionado es el de Strohmaier y Tochtermann [2005], quienes proponen un marco de trabajo para definir procesos de conocimiento, y una herramienta para facilitar el desarrollo de una “infraestructura de conocimiento”. Esta propuesta es ejemplificada por los autores por medio del desarrollo de un portal de conocimiento para una organización de desarrollo de software. Sin embargo, este enfoque se centra en el desarrollo de sistemas de AC, y no en el análisis de procesos de software.

Junto con los trabajos anteriores, existen otros que proponen enfoques de modelado de conocimiento y flujos de conocimiento en procesos organizacionales que han sido aplicados en procesos distintos a los de ingeniería de software. Entre estos está el de Bera et al. [2005] quienes adaptaron un enfoque de ingeniería del conocimiento para analizar requerimientos de conocimiento en procesos organizacionales. Estos autores ilustran el uso de su propuesta al identificar fuentes de conocimiento relacionadas con las actividades de un proceso, la manera en que estas fuentes cambian durante el proceso, y la información o el conocimiento principal que puede ser obtenido en esas fuentes. Por su parte, Kim et al. [2003] también han propuesto un tipo de diagrama específico para representar y analizar flujos de conocimiento entre procesos de una organización. Este enfoque consiste en un grafo compuesto por una serie de nodos que representan los procesos, y conexiones entre los nodos que representan los flujos de conocimiento entre dichos procesos. Finalmente, Papavassiliou y Mentzas [2003] han desarrollado un metamodelo por medio del cual estructurar flujos de trabajo diferenciando entre actividades comunes y actividades de AC, así como objetos de conocimiento de los objetos de información o datos. Sin embargo, esta última propuesta no propone un enfoque de modelado de procesos tal cual.

Finalmente, Rhem [2006] ha propuesto el uso de un lenguaje de modelado ampliamente usado y conocido en el área del desarrollo de software, como lo es UML [Jacobson *et al.*, 2000], para el desarrollo de sistemas de AC. El enfoque de Rhem se centra en el modelado de conocimiento para ser manejado por sistemas basados en conocimientos. Este enfoque no está definido para apoyar en el análisis de procesos, sino en ejemplificar la manera en que UML puede ser usado para modelar la estructura que guardará el conocimiento que será manejado por alguna herramienta de AC.

Del análisis que hemos hecho de los trabajos mencionados, podemos destacar tres observaciones:

- 1) No hemos encontrado ejemplos del uso de LMP orientados exclusivamente a los procesos de software.
- 2) Los enfoques analizados no están basados en LMP estandarizados, a excepción del de

Hansen y Kautz [2004] que usan Grafica Rica. Además, pocos enfoques presentan elementos para representar explícitamente el conocimiento involucrado en los procesos. Entre estos últimos, la mayoría propone su propio lenguaje de modelado, a excepción de Bera et al. [2005] quienes adaptan CommonKADS, una metodología desarrollada para apoyar en el desarrollo de sistemas basados en conocimiento [Schreiber *et al.*, 1994]; y Rhem [2006], quien propone el uso de UML para modelar la estructura del conocimiento manejado por sistemas de AC.

- 3) Resulta claro que LMPs generales pueden ser usados para identificar, de manera implícita, algunos aspectos relacionados con el flujo del conocimiento, tales como las fuentes de información que son requeridas, generadas, o modificadas en las actividades de un proceso, como ya ha sido observado por otros autores [Abdullah *et al.*, 2002]. Sin embargo, es importante que un LMP usado para analizar flujos de conocimiento provea primitivas para representar explícitamente aspectos relacionados con el conocimiento, tales como el conocimiento consumido o generado en las actividades, o el requerido por los roles que participan en dichas actividades, las fuentes de ese conocimiento, o las dependencias entre distintos tipos de conocimiento [Nissen y Levitt, 2004]. Esto es debido a que incluir explícitamente los elementos involucrados en un proceso en el modelo facilita su análisis [Conradi y Jaccheri, 1999].

De lo anterior, podemos concluir que una forma de abordar las carencias observadas en la literatura es adaptar un LMP estándar que esté enfocado al tipo de proceso que se pretende analizar. Siguiendo esta premisa, hemos adaptado el Metamodelo para Ingeniería de Procesos de Software (SPEM por sus siglas en inglés), el cual ha sido desarrollado por el grupo de administración de objetos (OMG: Object Management Group) [OMG, 2002].

Existen varias ventajas para usar SPEM como lenguaje de modelado para procesos de software: primero, UML es el lenguaje de modelado más ampliamente usado en las organizaciones de desarrollo de software. Lo anterior puede facilitar la asimilación de SPEM como un LMP estándar en este tipo de organizaciones. En segundo lugar, es posible usar casi cualquier herramienta de modelado en UML que apoye el uso de perfiles UML para producir modelos en SPEM [Bézivin y Breton, 2004]. Lo anterior es importante

debido a la gran cantidad de herramientas (comerciales, de libre distribución, de código abierto, etc.) que permiten el modelado por medio de UML, lo que hace que no sea necesario el desarrollo de una herramienta especial para poder realizar modelos en SPEM. Finalmente, SPEM permite el modelado de procesos de software desde distintas perspectivas, con diversos niveles de abstracción y por medio de un lenguaje formalmente definido. Las anteriores ventajas han provocado que SPEM sea visto como una alternativa de modelado de procesos de software en ambientes tan extendidos en la industria como lo es el ambiente de desarrollo Eclipse, el cual ha desarrollado el Eclipse Process Framework (<http://www.eclipse.org/epf/>), una herramienta para ingeniería de procesos de software que está basada en SPEM. Lo anterior puede contribuir a la adopción de SPEM como un LMP estándar para ambientes de desarrollo de software centrados en el proceso. Por todo lo mencionado, consideramos que SPEM es un lenguaje de modelado adecuado como la base para nuestra propuesta.

Sin embargo, la adaptación de SPEM se propone para modelar los detalles del proceso que se requieren durante la etapa de análisis del mismo. Nuestra propuesta es usar un lenguaje más flexible y genérico, por lo que también hemos realizado una adaptación de la técnica de Gráfica Rica. A continuación se da una descripción de estas dos adaptaciones.

IV.2 Adaptación de la técnica de Gráfica Rica

La Gráfica Rica es una técnica de modelado de procesos introducida por Peter Checkland como parte de su metodología para sistemas suaves [Checkland y Scholes, 1999]. Los sistemas suaves son aquellos donde el ser humano juega un papel preponderante, por lo que el sistema se puede ver afectado por aspectos propios del ser humano, tales como miedos, preocupaciones, ideas preconcebidas, creencias, cultura, etc. Si consideramos que el conocimiento es una característica inherente al ser humano, y como tal se ve afectado por aspectos propios de las personas, entonces un sistema de AC es por lo anterior un sistema suave, el cual se puede ver afectado por aspectos como los mencionados anteriormente. Por ejemplo, un sistema de AC puede ser subutilizado debido al miedo que tienen las personas por perder la ventaja que representa el conocimiento que ellos tienen,

por lo que deciden no compartirlo. Por otro lado, la decisión de usar o no cierto conocimiento puede depender de la confiabilidad que la persona que lo utilizará tenga con respecto a la fuente de donde proviene dicho conocimiento. Es por lo anterior que consideramos que el uso de la técnica de Grafica Rica es apropiado como paso inicial en el estudio de procesos con enfoque en el flujo del conocimiento. Dado que podría ser útil para identificar aspectos no técnicos que podrían estar afectando el flujo del conocimiento.

La Gráfica Rica se enfoca más a representar situaciones problemáticas, y en particular, problemas relacionados a las personas que participan en un proceso. Esto por medio de la representación general de los elementos involucrados en un proceso, sus relaciones principales, y los aspectos sociales que podrían afectar a las personas que intervienen en el proceso. Una gráfica rica normalmente es una vista general y de alto nivel del problema que se analiza. Además, es comúnmente usada durante la captura del proceso, por ejemplo, durante las entrevistas hechas a los involucrados para conocer la manera en que se realiza el proceso [Monk y Howard, 1998].

No existe una notación ni mecanismos formales para representar procesos por medio de una grafica rica. De hecho, como sus autores comentan, no existe una técnica formal o forma clásica para representar Graficas Ricas [Checkland y Scholes, 1999: p. 45]. Sin embargo, existen algunas notaciones que son comúnmente usadas al crear graficas ricas, las cuales son descritas a continuación.

IV.2.1 Descripción de la técnica de Grafica Rica

Una gráfica rica es esencialmente un representación de un proceso por medio de un dibujo tipo caricatura que identifica a todos los involucrados, sus preocupaciones, y algunas de las estructuras que definen el contexto de trabajo en el proceso [Monk y Howard, 1998]. El objetivo es utilizar estos elementos de la representación para razonar a cerca de ellos, por ejemplo la manera en que se realiza actualmente dicho proceso, los problemas que pueden estarse presentando, etc. Por lo tanto, es importante que dicha representación contenga explícitamente los elementos sobre los cuales se quiere razonar.

Normalmente, una gráfica rica contiene los siguientes elementos:

- **Actividades.** Las principales actividades que constituyen el proceso, las cuales pueden ser representadas por medio de óvalos, o nubes.
- **Roles.** Representan a las personas que intervienen en el proceso, y pueden ser representadas por cualquier imagen que se considere apropiada, por ejemplo un dibujo de una persona, o algún otro dibujo que represente mejor el papel que juega la persona dentro del proceso.
- **Preocupaciones.** Representan los aspectos humanos de las personas involucradas en el proceso, tales como sus inquietudes, miedos, intereses, etc. Estas pueden ser representadas como nubes de pensamiento que salen del rol que representa a la persona en cuestión.
- **Conexiones.** Son líneas que conectan a las actividades con los distintos elementos que intervienen en el proceso. Por lo general, los roles son conectados con las actividades con líneas sin dirección, mientras que los elementos de información son conectados por medio de flechas cuya dirección indica si dicho elemento es una salida (flecha dirigida desde la actividad al elemento de información) o entrada (flecha dirigida del elemento de información a la actividad) de la actividad, o si es modificado (flecha con doble dirección) dentro de esta.
- **Fuentes de información.** Representan los documentos o información que es usada, generada o modificada dentro de las actividades. Estos pueden ser representados con rectángulos, o con el dibujo que se considere más apropiado.

Aprovechando la flexibilidad de la técnica de Gráfica Rica, y con el fin de usarla para estudiar aspectos relacionados con el conocimiento involucrado en un proceso, hemos definido una serie de convenciones para el modelado de procesos por medio de gráfica rica, que incluyan de manera explícita el conocimiento involucrado en el proceso. Estas convenciones son descritas a continuación.

IV.2.2 Adaptaciones propuestas

Principalmente, la adaptación realizada define elementos para representar las actividades, los roles y el conocimiento involucrado en el proceso. La Figura 15 muestra estos elementos, los cuales corresponden a:

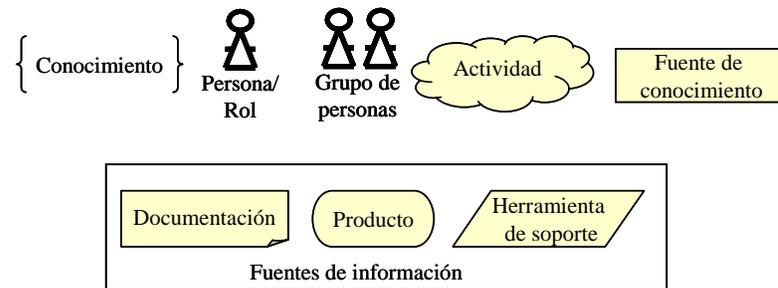


Figura 15. Elementos de la adaptación de la técnica de Gráfica Rica.

- **Personas o roles.** Usado para representar a los actores del proceso, ya sea personas específicas o roles del proceso.
- **Grupos de personas.** Usado para representar a grupos de personas (por ejemplo un equipo de trabajo) con necesidades de conocimiento comunes; o grupos de personas donde cualquiera de ellas, o todas en conjunto, pueden aportar conocimiento que es requerido para las actividades.
- **Actividades.** Igual que en la nomenclatura clásica de la gráfica rica.
- **Fuentes de conocimiento.** Se puede representar de forma similar a las fuentes de información en la nomenclatura clásica de la gráfica rica. Sin embargo, se proponen tres íconos para distinguir entre distintos tipos de fuentes: documentos, productos del proceso, y herramientas que apoyan al proceso.
- **Conocimiento.** Representado como texto encerrado entre dos llaves. Cada línea de texto corresponde con un área o tema de conocimiento.

El conocimiento puede relacionarse con los distintos elementos del modelo de distintas formas. Para ejemplificar estas conexiones, tomaremos el escenario del capítulo

anterior para presentar su gráfica rica, mostrada en la Figura 16. En dicha gráfica se presentan los distintos tipos de conexión que pueden existir entre el conocimiento y los elementos del modelo, mismos que son descritos a continuación.

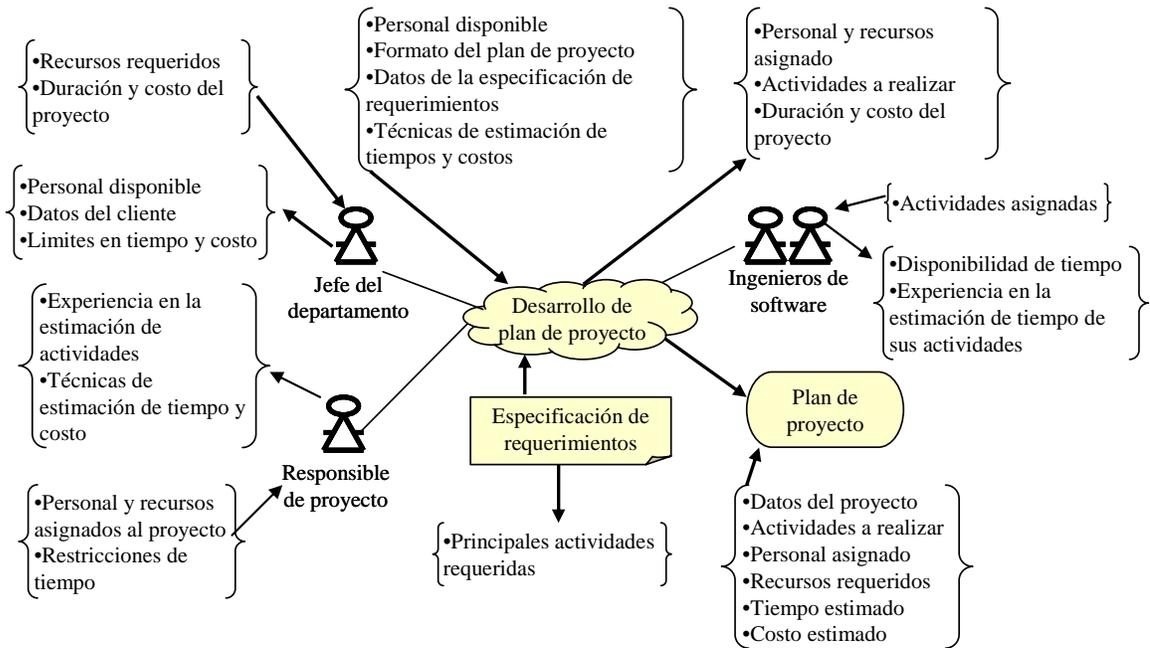


Figura 16. Modelo del escenario ejemplo por medio de una gráfica rica adaptada. Presenta los distintos tipos de conexiones entre los elementos del modelo y el conocimiento involucrado.

- Conocimiento de los roles.** Es el conocimiento que aportan las personas que intervienen en el proceso. Dichas personas pueden ser un solo individuo o un grupo de los mismos. En este caso, los temas de conocimiento se conectan al ícono que representa a la persona que lo aporta por medio de una flecha dirigida desde el rol hacia los temas de conocimiento, indicando que dicho conocimiento es obtenido de dicho rol o persona. Debido a que los diagramas se pueden volver muy grandes, es importante sólo centrarse en los temas de mayor relevancia para cada elemento del modelo. Por ejemplo, en la Figura 16, el jefe del departamento apoya con información sobre el personal disponible, los datos del cliente, y los límites de tiempo y costo a los que se debe ajustar el proyecto.

- **Conocimiento obtenido por los roles.** Es el conocimiento que obtienen las personas al participar en un proceso. En este caso, la flecha se dirige desde los temas hacia la persona, para indicar que dicho conocimiento es obtenido por el rol o persona en cuestión. Igualmente, es importante sólo centrarse en los temas más relevantes para cada elemento. Por ejemplo, en el modelo de la Figura 16 se decidió indicar que el responsable del proyecto, después de participar en la actividad, conocerá cuales son el personal y los recursos asignados al proyecto, así como el tiempo al que se deberá ajustar; mientras que los ingenieros de software sabrán cuáles son las actividades que les fueron asignadas.
- **Conocimiento requerido por las actividades.** Es el conocimiento que necesitan tener las personas que desempeñan dichas actividades. En este caso, la flecha va dirigida desde los temas de conocimiento hacia la actividad donde son requeridos. En el caso del modelo de la Figura 16 se ha querido hacer énfasis en la necesidad de conocimiento sobre el personal disponible, el formato que debe seguir el plan del proyecto, información de la especificación de requerimientos asociada al proyecto, y técnicas para la estimación del tiempo y costo del proyecto.
- **Conocimiento generado por las actividades.** Es el conocimiento que puede generarse al realizar una actividad. Para este caso, la flecha va dirigida desde la actividad hacia los temas de conocimiento. Se utiliza para enfatizar el conocimiento o información más relevante generada durante la actividad. En el caso del ejemplo de la Figura 16, se ha decidido enfatizar que lo importante es saber qué recursos y personas han sido asignadas al proyecto, qué actividades deberán ser realizadas, y cuál será la duración y costo estimado del proyecto.
- **Conocimiento obtenido de alguna fuente.** Representa el conocimiento que contienen las fuentes de información o conocimiento que son usadas durante las actividades. Para conectar la fuente con las actividades donde son usadas, se utiliza la notación de la técnica de Gráfica Rica descrita anteriormente, mientras que para conectar el conocimiento que aporta dicha fuente, se utiliza una flecha dirigida de la fuente a los temas de conocimiento, para indicar que dicho conocimiento es extraído de la fuente. Un ejemplo es la información extraída de la especificación de requerimientos en el modelo

de la Figura 16, el cual corresponde a las principales actividades que deberán ser realizadas en base a los requerimientos, lo que ayudará a estimar el tiempo y costo del proyecto.

- **Conocimiento almacenado en alguna fuente.** Representa el conocimiento que es guardado en alguna fuente de información o conocimiento que es generado o modificado en una actividad. En este caso, los temas de conocimiento se conectan por medio de una flecha que va de los temas a la fuente para indicar que dicho conocimiento es almacenado en esa fuente. La información almacenada en el plan del proyecto (el cual es un producto del proceso modelado) en la Figura 16, es un ejemplo.

De lo anterior podemos observar que por medio de las convenciones propuestas es posible comenzar a identificar los flujos de conocimiento dentro de un proceso, por ejemplo, conocimientos obtenidos por los roles, almacenados en las fuentes, generados en las actividades, etc. Sin embargo, la adaptación de la técnica de Gráfica Rica sólo se propone como un paso inicial en la identificación de los principales elementos involucrados en el proceso, en particular, el conocimiento requerido y generado durante las actividades, así como las fuentes de donde éste es obtenido o donde es almacenado.

El uso de la técnica Gráfica Rica es un paso inicial en el estudio de un proceso, y requiere ser complementada con otras técnicas que permitan identificar los detalles del proceso, así como representarlo desde diferentes perspectivas y niveles de abstracción [Monk y Howard, 1998]. Con este propósito es que se propone también, como apoyo para la segunda etapa en el análisis, la adaptación de SPEM descrita a continuación.

IV.3 SPEM-KF: Adaptación del Metamodelo para Ingeniería de Procesos de Software (SPEM)

Como ya se mencionó, para la etapa del modelado detallado del proceso se propone una adaptación de SPEM. Esta adaptación toma como base el metamodelo de tipos y fuentes de conocimiento presentado en el capítulo anterior, y lo integra a SPEM, con el objetivo de dotar al lenguaje de modelado propuesto por la especificación de SPEM con

elementos para representar explícitamente aspectos como el conocimiento involucrado en las actividades de un proceso, sus fuentes, dependencias entre tipos y/o fuentes de conocimiento entre otras cosas. A continuación se describen estas adaptaciones, pero primero se da una breve descripción de SPEM.

IV.3.1 Descripción de SPEM

SPEM es un metamodelo basado en el Lenguaje Unificado de Modelado (UML por sus siglas en inglés) [Jacobson et al., 2000], especialmente diseñado para el modelado de procesos de software. SPEM ha sido definido para describir procesos de software y sus componentes, siguiendo un enfoque de modelado orientado a objetos con base en UML. El metamodelo ha sido definido como un perfil de UML; es decir, SPEM es una variante de UML que usa los mecanismos de extensión de UML en una forma estandarizada con el propósito de modelar procesos de software.

En términos generales, un modelo de proceso en SPEM se define como un conjunto de elementos de trabajo tales como actividades, que son clasificados como definiciones de trabajo (*WorkDefinitions*). Estas definiciones de trabajo son operaciones que describen el trabajo realizado por los roles en un proceso. Las definiciones de trabajo son usadas para estructurar el proceso, por ejemplo, describiendo su ciclo de vida, sus fases, sus iteraciones o actividades. El resultado de una definición de trabajo se conoce como producto de trabajo (*WorkProduct*). Un producto del trabajo puede ser cualquier cosa producida, consumida o modificada en un proceso. Con base en los elementos anteriores, en SPEM un proceso de desarrollo de software es considerado una colaboración entre entidades activas abstractas llamadas roles del proceso (*Role*) que realizan operaciones llamadas actividades (*Activity*) en entidades tangibles y concretas llamadas productos del trabajo [OMG, 2002]. Este modelo base de los modelos de procesos de software en SPEM es mostrado en la Figura 17.

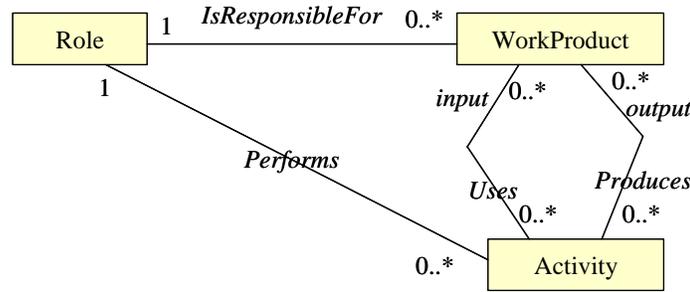


Figura 17. Modelo conceptual base del modelado de procesos con SPEM

Si bien los tres elementos de la Figura 17 son la base del modelado en SPEM, el metamodelo completo contempla una amplia variedad de entidades para definir distintos tipos de datos, tipos de elementos para los modelos de procesos, sus atributos y relaciones, la estructura del proceso, su ciclo de vida, etc. La Figura 18 muestra un diagrama con los principales elementos de la especificación de SPEM, así como las relaciones entre estos. Como se puede observar, la estructura de un proceso (*Process*) puede definirse por medio de su ciclo de vida (*Lifecycle*), sus fases (*Phase*), iteraciones (*Iteration*) y actividades (*Activity*), los cuales constituyen definiciones de trabajo (*WorkDefinition*). Una actividad puede también definirse como una serie de pasos (*Step*) a ser realizados. A su vez, una definición de trabajo puede estar restringida (*Constraint*) por medio de las metas (*Goal*) perseguidas por la misma, y sus precondiciones (*Preconditions*). Finalmente, las actividades son realizadas por roles del proceso (*ProcessRole*) los cuales son realizadores del proceso (*ProcessPerformed*) que son responsables de los productos del trabajo (*WorkProduct*).

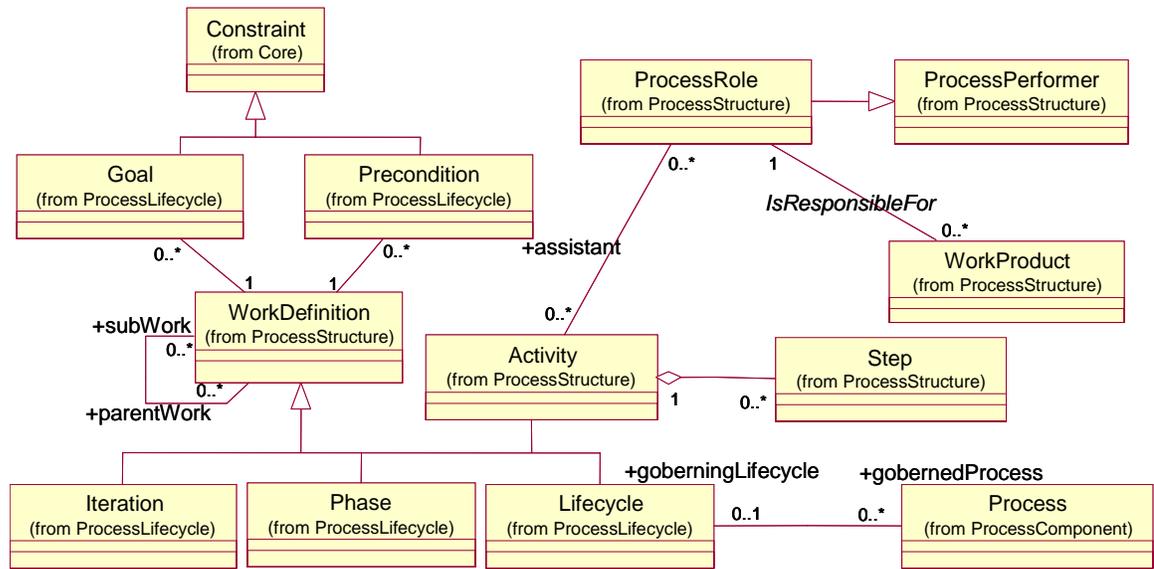


Figura 18. Elementos principales de la especificación de SPEM.

La notación de los principales elementos que componen un modelo en SPEM se ilustran en la Figura 19. Sólo para estos se ha definido una notación gráfica estándar. El resto de los elementos del metamodelo se representan con la notación de UML para los estereotipos; aunque también se puede utilizar algún otro ícono que se considere adecuado.

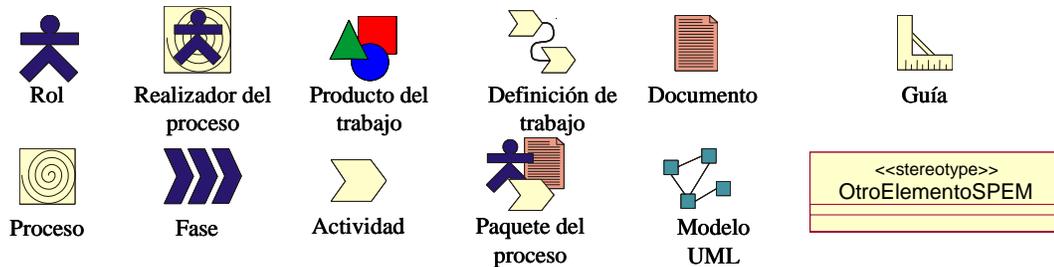


Figura 19. Íconos para los principales elementos de SPEM.

Los conceptos definidos en SPEM pueden ser usados en el siguiente conjunto de diagramas para representar modelos de procesos:

- **Diagrama de clases.** Representan la estructura de los productos del trabajo, tales como la manera en que están relacionados con los roles en el proceso, las relaciones de herencia, descomposición, dependencias, o asociaciones simples con otros productos del trabajo.
- **Diagramas de paquetes.** Ilustran la organización de un proceso, por ejemplo al agrupar los productos de trabajo, actividades, roles, sub-procesos, etc. que están relacionados.
- **Diagramas de casos de uso.** Muestran las relaciones entre los roles y las definiciones de trabajo de un proceso.
- **Diagramas de secuencia.** Describen las iteraciones entre instancias de elementos de un proceso.
- **Diagramas de estados.** Representan la vista del comportamiento de los elementos de un modelo.
- **Diagramas de actividad.** Presentan la secuencia de actividades con los productos del trabajo que entran y salen de las mismas. Este tipo de diagrama también puede mostrar el estado de los objetos que fluyen a lo largo del proceso. Así mismo, es posible definir las responsabilidades de los roles del proceso al usar líneas que delimitan las actividades que corresponden a cada rol.

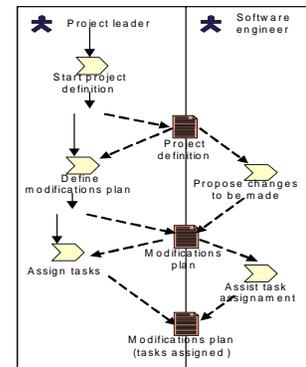
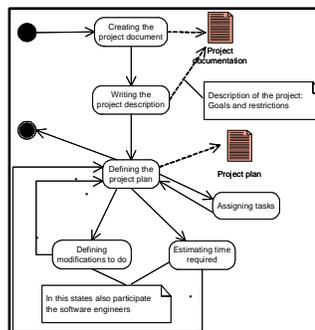
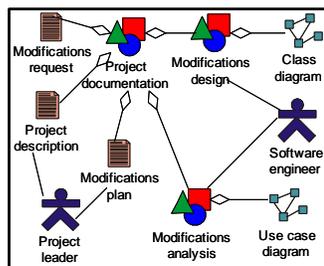
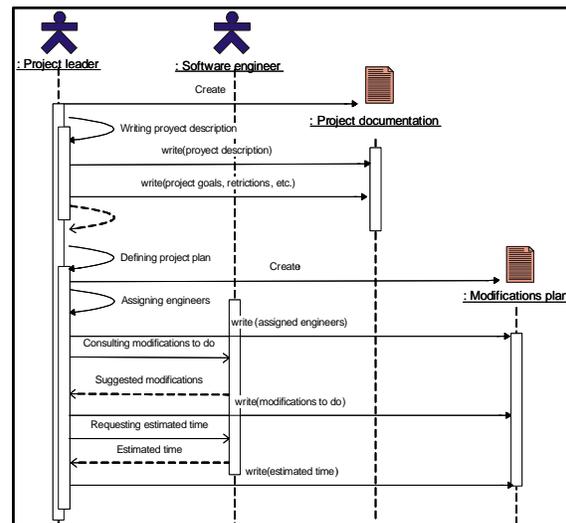
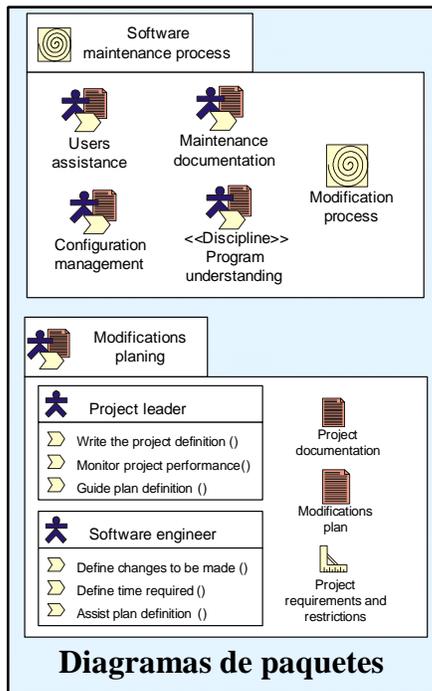
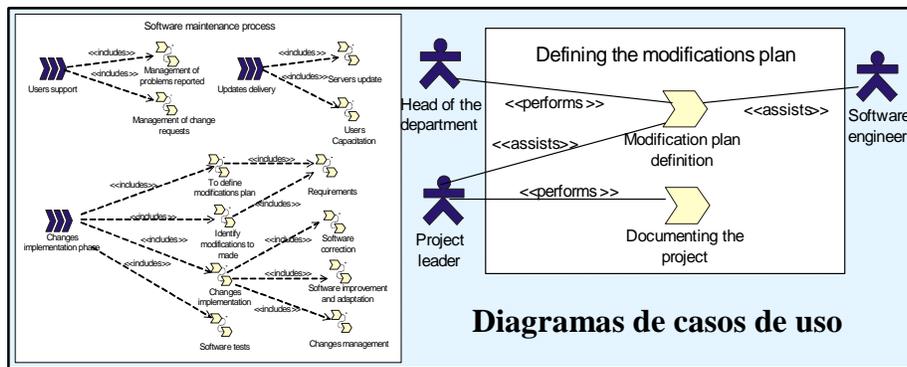


Figura 20. Ejemplos de diagramas basados en SPEM. Da una visión general de los tipos de diagramas que se pueden crear con el lenguaje de modelado propuesto en SPEM.

La Figura 20 muestra ejemplos de los diagramas que pueden ser usados para modelar un proceso con la notación propuesta por la especificación de SPEM. Algunos de estos diagramas pueden ayudar a identificar, de manera implícita, algunos aspectos relacionados con los flujos de conocimiento. Por ejemplo, los diagramas de actividad pueden mostrar los documentos que son usados, modificados o generados en las actividades. De esta manera, es posible darse una idea de la información que fluye a lo largo del proceso, al intentar intuir el tipo de información que se puede estar obteniendo o capturando en dichos documentos. Sin embargo, SPEM no provee elementos para representar de forma explícita el conocimiento involucrado en las actividades. Para solventar esta carencia se propone una extensión a SPEM.

IV.3.2 Extensión de SPEM

La extensión de SPEM que se propone en esta sección consiste en un conjunto de elementos de modelado adicionales a los propuestos por la especificación de SPEM. El propósito principal de esta adaptación se centra en tres aspectos:

- 1) Ilustrar el conocimiento y sus fuentes, que están involucrados en las actividades del proceso.
- 2) Ilustrar la manera en que fluye el conocimiento a través de las actividades, o la forma en que fuentes específicas están siendo usadas o modificadas a lo largo del proceso.
- 3) Ilustrar transferencias de conocimiento entre fuentes, y a través de las actividades.

Para lograr lo anterior, se definieron algunos conceptos y relaciones que han sido agrupados como un paquete dentro de un metamodelo de conceptos de conocimiento llamado *KnowledgeConcepts*. Este metamodelo está basado en el metamodelo de tipos y fuentes de conocimiento descrito en el Capítulo III, Sección III.4, de este documento. El metamodelo se definió como un paquete que ha sido integrado al resto de los paquetes de elementos que constituyen el metamodelo de SPEM. A continuación se describen los distintos elementos del paquete *KnowledgeConcepts*.

IV.3.2.1 El conocimiento como un producto del trabajo

Para el enfoque de modelado propuesto, los tipos y fuentes de conocimiento son considerados como entidades que son usadas, generadas o modificadas durante las actividades de un proceso. Es decir, estos tipos y fuentes de conocimiento son productos de trabajo. Considerando la definición que sobre estos últimos elementos hace la especificación de SPEM [OMG, 2002]. Con base en lo anterior, hemos incluido los conceptos de conocimiento (*KConcept* que agrupa a los tipos y fuentes de conocimiento) como tipos especiales de productos del trabajo denominados *InvolvedKnowledge* o conocimiento involucrado (véase la Figura 21). Este elemento agregado al metamodelo representa el conocimiento que está envuelto en las actividades. Este conocimiento puede ser conceptos de conocimiento específicos, cada uno representado como un elemento *KConcept*, o grupos de estos conceptos. Para definir conjuntos de conceptos de conocimientos, se ha definido un tipo de paquete denominado *GroupedKnowledge* o conocimiento agrupado.

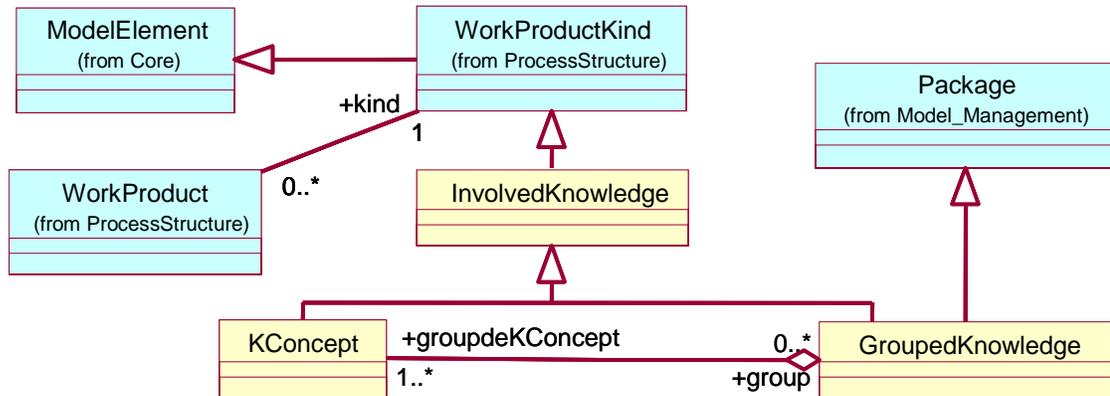


Figura 21. Integración de los conceptos de conocimiento como productos del trabajo.

La Figura 21 muestra la manera en que estos tres elementos antes mencionados se integran con la especificación de SPEM. Los cuatro elementos en la parte superior de la figura corresponden a elementos de SPEM. Como se puede observar, el elemento

InvolvedKnowledge ha sido definido como un tipo de producto del trabajo (*WorkProductKind*), así mismo, los conceptos de conocimiento (*KConcept*), y los grupos de estos conceptos (*GroupdeKnowledge*) son también de tipo *InvolvedKnowledge*, y por tanto, tipos de productos del trabajo. Finalmente, un elemento de tipo *GroupedKnowledge* es también un tipo de paquete (*Package*), el cual puede agrupar un conjunto de conceptos de conocimiento.

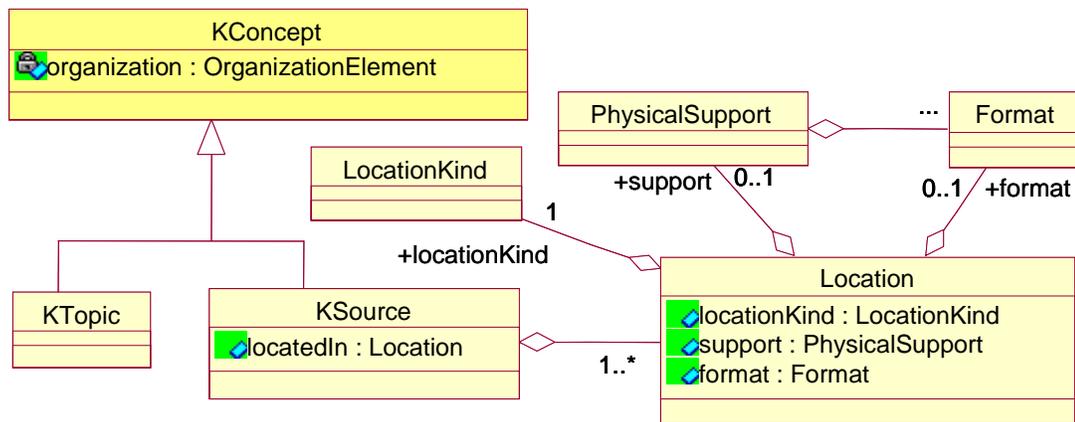


Figura 22. Metamodelo de conceptos de conocimiento (elementos principales del paquete KnowledgeConcepts).

Los conceptos de conocimiento que forman parte del paquete *KnowledgeConcepts* son mostrados en la Figura 22. Son dos los conceptos de conocimiento principales, los cuales derivan de *KConcept*: las fuentes de conocimiento (*KSource*), y los tipos o temas de conocimiento (*KTopic*). Las fuentes de conocimiento pueden tener una localidad asociada (*Location*), es decir, una descripción del lugar donde pueden ser consultadas. Esta localidad puede ser una dirección física, un correo electrónico, una dirección electrónica, etc., la cual se define como un elemento de tipo *LocationKind* que representa el tipo de localidad. A su vez, una localidad puede estar asociada a un medio físico, como un documento en papel impreso, un video, una fotografía, etc., representado por un elemento de tipo soporte físico (*PhysicalSupport*). Finalmente, dependiendo de este soporte físico, y el tipo de localidad, la fuente podría tener asociado un formato (*Format*); por ejemplo, si se trata de una fuente en

un archivo electrónico, podría a su vez tener un formato como documento Word, Excel, pdf, etc. Finalmente, una localidad física puede también tener un conjunto de posibles formatos, como se puede observar en el diagrama de la Figura 22. Con base en la definición del metamodelo descrito en la Figura 22, es posible también manejar fuentes que pudieran estar replicadas, por ejemplo, que existiera una copia en formato electrónico en un repositorio de documentos, y una en forma de papel impreso almacenado en algún repositorio de archivos. En estos casos, es posible definir la fuente una sola vez, y asociar a ella distintos formatos y o soportes físicos.

Adicionalmente, se han definido un conjunto de elementos para permitir la organización (OrganizationElement) de las diferentes fuentes o temas de conocimiento mostrados en la Figura 23. De esta manera, tanto las fuentes como los tipos o temas de conocimiento pueden ser clasificados. Por ejemplo para definir una taxonomía de tipos o fuentes de conocimiento, lo cual constituye un paso inicial en muchas iniciativas de AC [Rao, 2005b].

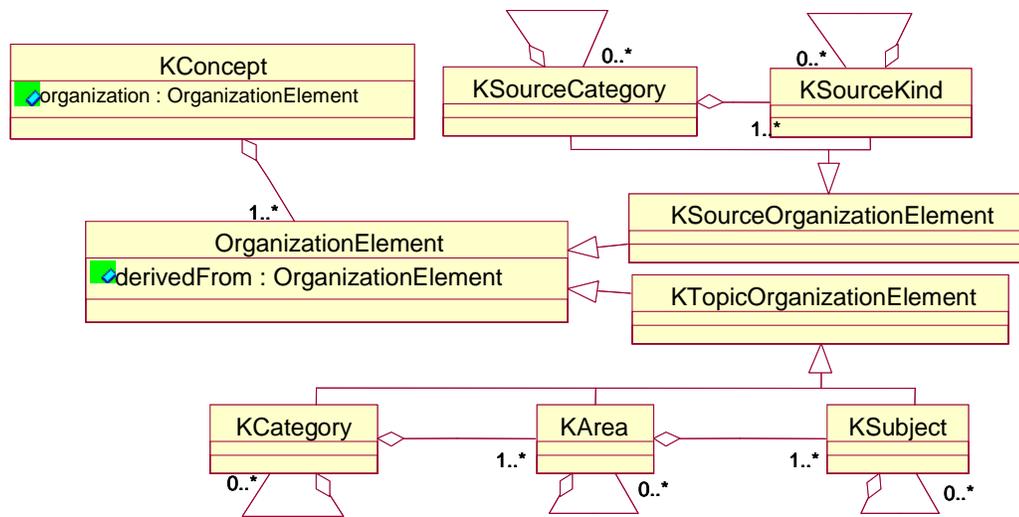


Figura 23. Elementos de organización integrados al paquete KnowledgeConcepts.

Como se puede observar en la Figura 23, los temas o tipos de conocimiento pueden ser organizados siguiendo un esquema de categorías (*KCategory*), áreas (*KArea*) y temas

(*KTopic*), algo similar al esquema usado en el SEWBOK [Abran *et al.*, 2004; Hilburn *et al.*, 1999b] para clasificar áreas y tópicos de conocimiento relacionados con la Ingeniería del Software. Las fuentes de conocimiento son clasificadas siguiendo un esquema compuesto por categorías (*KSourceCategory*) y tipos (*KSourceKind*). Al seguir el esquema propuesto, es posible identificar, a partir de una determinada fuente o tema de conocimiento, a qué tipo, área o categoría específica pertenece. Debido a que cada elemento de organización apunta a su antecesor por medio del atributo *derivedFrom*, también es posible hacer un recorrido hacia atrás para identificar, por ejemplo, a qué área o categoría pertenece un determinado tema de conocimiento. Así mismo, dado una categoría, es posible recorrer sus sub-categorías, áreas, sub-áreas, y temas específicos, para el caso de los temas de conocimiento, o sub-categorías, tipos y sub-tipos para el caso de las fuentes de conocimiento.

IV.3.2.2 Relaciones y dependencias entre los conceptos de conocimiento

Para permitir la representación de transferencias de conocimiento, así como del conocimiento que una fuente puede contener, se han definido un conjunto de relaciones mostradas en la Figura 24. Particularmente, se han definido dos relaciones (*Relationship*) denominadas *KnowledgeTransfer* y *KnowsAbout*. La relación *KnowledgeTransfer* es utilizada para representar transferencias de conocimiento entre fuentes, por ejemplo entre personas, una persona y un documento, etc. Esta es una relación múltiple, es decir, puede asociar más de dos elementos a la vez, y contiene dos propiedades principales: las fuentes que participan en la transferencia (*Knower*), y la definición de trabajo donde se realiza dicha transferencia (*WorkDefinition*). El conocimiento transferido u obtenido por una fuente se define por medio de relaciones de tipo *KnowsAbout*. Este último tipo de relación asocia a la fuente (*Knower*) con el conocimiento que contiene (*KnownKConcept*). Los elementos *KnownKConcept* pueden hacer referencia tanto a fuentes o temas de conocimiento individuales, o a grupos de estos por medio de elementos de tipo *InvolvedKnowledge*. Así mismo, la propiedad *knowledgeLevel* definida en el elemento *KnownKConcept*, permite que se indique el nivel de conocimiento (*KLevel*) que una fuente puede tener, por ejemplo, sobre un tema en particular. Los posibles niveles de

conocimiento deben ser definidos acorde con las necesidades particulares de quien haga uso del metamodelo.

Adicionalmente a las relaciones antes mencionadas, se han definido dos tipos de dependencias mostradas en la Figura 25. La primera, denominada *Required*, es para definir el conocimiento que es requerido en las actividades o flujos de trabajo de un proceso, o para representar dependencias entre áreas o temas de conocimiento, por ejemplo, para indicar que para aplicar un tema de conocimiento se requiere conocer sobre algún otro. La segunda dependencia, denominada *Affected* es usada para definir el conocimiento que es afectado por las actividades o flujos de trabajo. Cada dependencia tiene un cliente (*client*) y un proveedor (*supplier*). En el caso de la dependencia *Required*, el cliente puede ser una definición de trabajo (*WorkDefinition*) o un elemento de tipo *InvolvedKnowledge*, mientras que el proveedor deberá ser siempre un elemento de tipo *InvolvedKnowledge*. En el caso de la dependencia *Affected*, el cliente siempre deberá ser un elemento de tipo *InvolvedKnowledge*, mientras que el proveedor deberá ser de tipo *WorkDefinition*.

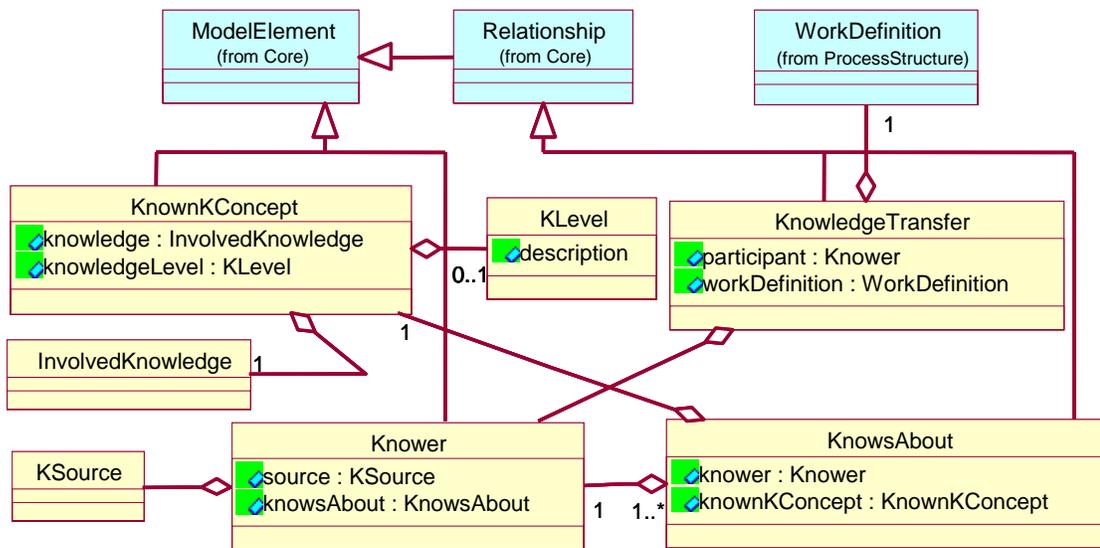


Figura 24. Relaciones definidas en el paquete KnowledgeConcepts (Los tres elementos en la parte superior pertenecen a la especificación de SPEM).

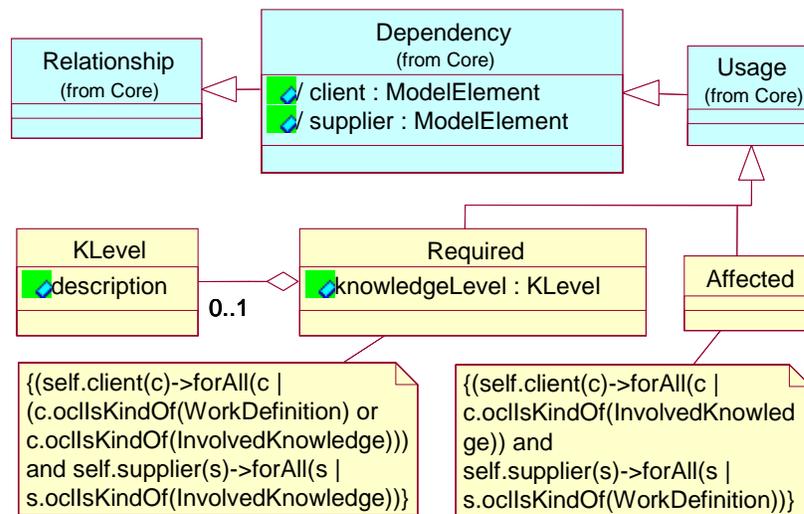


Figura 25. Dependencias que forman parte del paquete KnowledgeConcepts (Los tres elementos en la parte superior de la figura forman parte de la especificación de SPEM.)

Se propone que las dependencias descritas sean utilizadas durante la definición de casos de uso, con el fin de definir los paquetes de conocimiento que son requeridos o modificados durante los distintos flujos de trabajo. De esta manera, desde el inicio de la definición del proceso se puede comenzar a identificar los paquetes de conocimiento principales involucrados en el proceso, para posteriormente identificar las fuentes, áreas o temas específicos que formarán parte de esos paquetes, así como las actividades particulares donde dichas fuentes, áreas o temas de conocimiento son utilizados, generados o modificados. La dependencia *Required* también puede ser utilizada en los diagramas de clases para definir dependencias entre conceptos de conocimiento. A continuación se describen con mayor detalle los diagramas donde pueden ser usados los distintos elementos que componen el metamodelo de conceptos de conocimiento, pero antes se presenta la notación propuesta para los elementos principales de este metamodelo.

IV.3.2.3 Notación

La notación propuesta para representar temas, fuentes y transferencias de conocimientos en los modelos es mostrada en la Figura 26. Estos íconos son usados en

conjunto con la notación propuesta en la especificación de SPEM [OMG, 2002]. El ícono *KTopic* es usado para representar temas o tópicos de conocimiento y habilidades requeridas o generadas en las actividades, así como aquel conocimiento que poseen las personas que participan en el proceso, o el conocimiento que es almacenado u obtenido de las fuentes de conocimiento. El paquete *GroupedKnowledge* es usado para representar grupos de fuentes, temas o tópicos de conocimiento, o habilidades relacionadas. Por ejemplo, pueden ser usados para representar áreas de conocimiento, grupos de estas áreas, el conocimiento obtenido de una fuente determinada, etc. El ícono *KSource* es utilizado para distinguir las fuentes de conocimiento (tanto documentos como sistemas de información) de documentos comunes y corrientes. Para representar personas o roles, se hace uso del ícono definido en la especificación de SPEM para este fin. Finalmente, el ícono *KnowledgeTransfer* es utilizado para indicar relaciones de transferencias de conocimiento.



Figura 26. Íconos de la notación de los elementos del paquete KnowledgePackage.

Es importante añadir que los íconos propuestos son usados en los diagramas comunes definidos por la especificación de SPEM, a excepción del ícono *KnowledgeTransfer*, el cual es utilizado dentro de un nuevo tipo de diagrama propuesto en este trabajo. Este diagrama se describe en la siguiente sub-sección, donde se detalla el uso de los distintos elementos incluidos dentro de los diagramas que conforman la especificación de SPEM, así como un par de tipos de diagramas nuevos que son propuestos en este trabajo.

IV.3.2.4 Diagramas

Los diagramas definidos para su uso como apoyo en la metodología se centran en tres aspectos principales:

- 1) Apoyar en la identificación de las principales fuentes y conocimiento involucrado en las actividades del proceso.
- 2) Apoyar en la clasificación y estructuración de los tipos y fuentes de conocimiento.
- 3) Apoyar en el modelado y análisis de los flujos de conocimiento que se dan durante el proceso.

Con el propósito de lograr lo anterior, se han hecho adaptaciones de algunos de los diagramas definidos en la especificación de SPEM. En particular, se han adaptado los diagramas de casos de uso, de actividad y de secuencia. Adicionalmente se han propuesto un par de nuevos tipos de diagramas para apoyar en la identificación de flujos de conocimiento. A continuación se describen los diagramas que apoyan cada uno de los objetivos anteriormente mencionados, y se presentan ejemplos para ilustrar el uso de los diagramas adaptados y los dos nuevos tipos de diagramas propuestos.

IV.3.2.4.1 Identificación de actividades, tipos y fuentes de conocimiento

Son dos los diagramas principales que pueden ser utilizados para la identificación de las actividades, tipos y fuentes de conocimiento principales del proceso, estos son los diagramas de casos de uso y los de actividad.

Diagramas de Casos de Uso

Los diagramas de casos de uso pueden ayudar a obtener una visión general del proceso al identificar las principales actividades o flujos de trabajo, así como su relación con los roles del proceso. Para aprovechar esta característica, hemos adaptado los diagramas de casos de uso para también permitir la identificación de las relaciones entre las principales actividades, y las principales fuentes de conocimiento. Lo anterior se logra al añadir al diagrama elementos de tipo *GroupedKnowledge* que representan el conjunto de conocimientos que una fuente aporta, u obtiene de una determinada actividad o flujo de trabajo. La Figura 27 muestra un caso de uso para el ejemplo que hemos estado siguiendo. En este caso de uso, se identifican tres actividades principales que constituyen el desarrollo del plan del proyecto: la escritura del documento, la estimación de tiempo y costo, y la

definición y asignación de actividades. Así mismo, se observa que las actividades dependen del conocimiento que aportan el Jefe del Departamento, el Responsable del Proyecto, y los Ingenieros de Software, mientras que dentro de éstas se genera la información que es contenida en el documento del plan del proyecto.

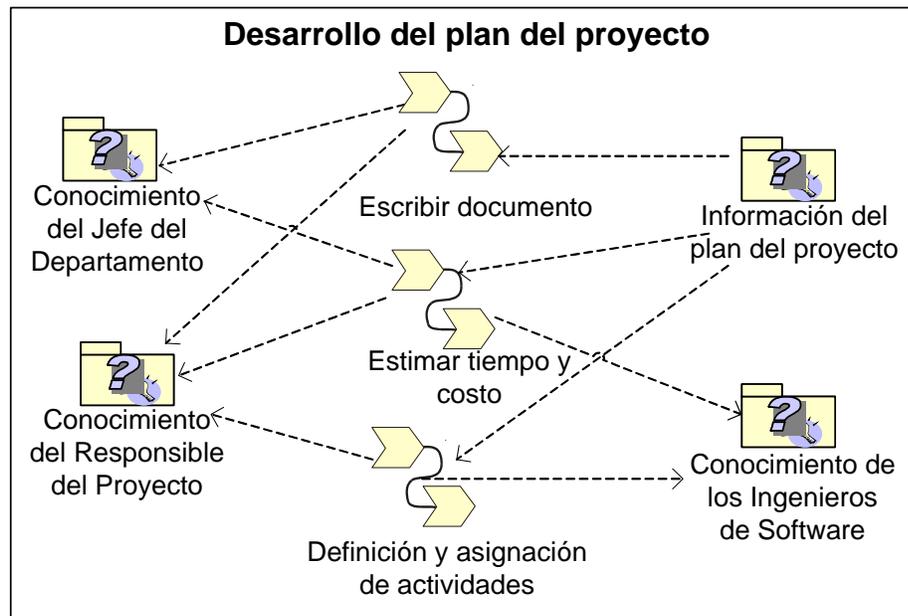


Figura 27. Ejemplo de un diagrama de casos de uso adaptado. Las carpetas representan paquetes para agrupar el conocimiento usado y generado en los flujos de actividades de un proceso. La dirección de la flecha indica si el conocimiento del paquete es generado (del paquete al flujo de actividades) o requerido (viceversa).

Los elementos tipo *GroupedKnowledge* se asocian con las actividades por medio de dependencias. Las dependencias son flechas discontinuas, donde la flecha apunta hacia el elemento del que se depende. De esta forma, una flecha apuntando de una actividad a un paquete de conocimiento indica que al menos parte del contenido de ese paquete (es decir el conocimiento) es requerido en dicha actividad (por ejemplo, el conocimiento del Responsable del Proyecto es requerido en las tres actividades en la Figura 27). Por el contrario, una flecha que apunta del paquete de conocimiento hacia la actividad, indica que parte del contenido de dicho paquete depende de la actividad, ya sea porque es generado o

modificado en ella (por ejemplo, la información que contiene el documento del plan del proyecto es generada en las tres actividades de la Figura 27). El recuadro de la Figura 27 indica los límites del proceso, sub-proceso, fase, etc., que está siendo representado por el caso de uso. De los diagramas como el anterior pueden surgir varias preguntas, por ejemplo:

- ¿Existe conocimiento generado en una actividad que es utilizado en otra? De ser afirmativa la respuesta, pueden surgir dos casos: si el conocimiento es almacenado en una fuente indicaría que ésta sirve como canal de transmisión de conocimiento entre las actividades, de lo contrario, estaríamos ante un posible problema o riesgo de pérdida de conocimiento o información por no almacenarla adecuadamente.
- Dado el primer caso, ¿el conocimiento almacenado en la fuente es el aportado por alguno de los roles o actores? Si este es el caso, entonces indicaría que la fuente podría estar siendo usada como un medio para compartir conocimiento.
- Por otra parte, ¿el conocimiento almacenado en la fuente es el requerido por alguno de los roles que participan en la actividad donde dicha fuente es usada? De ser afirmativa esta respuesta y las dos anteriores, esto indicaría que la fuente está sirviendo como canal para compartir conocimientos no solo entre sub-procesos, sino entre las personas que participan en el proceso.

Como se puede observar de lo anterior, los casos de uso pueden ser un apoyo en la identificación de aspectos específicos a analizar con mayor detalle. Sin embargo, debido a su generalidad, los casos de uso no permiten obtener detalles del proceso, como por ejemplo, la respuesta precisa a las preguntas antes expuestas. Son más un primer paso para comenzar a identificar elementos o situaciones que deben ser detallados. Estos detalles deberán ser obtenidos con el resto de los diagramas. Por ejemplo, los diagramas de paquetes descritos más adelante, podrían usarse para detallar los temas o áreas de conocimiento específicos contenidos en cada paquete identificado en los casos de uso.

Es importante también añadir que no necesariamente los paquetes de conocimiento incluidos en los casos de uso deberán corresponder con fuentes de conocimiento. Por

ejemplo, si se quiere separar el conocimiento que una determinada fuente aporta, y aquel que obtiene, se podrían tener dos paquetes separados, que posteriormente en un diagrama de paquetes podrían agruparse como parte del conocimiento completo de esa fuente.

Diagramas de Actividad

Los diagramas de actividad pueden apoyar en la identificación detallada de las actividades, su secuencia, los roles que las deben desempeñar, y los productos de trabajo que fluyen a lo largo de las actividades, es decir, los que son requeridos, generados o modificados en dichas actividades. Por lo tanto, este tipo de diagramas puede ser utilizado para identificar los temas o áreas de conocimiento específicos que son requeridos, generados o modificados en cada actividad. Si a esto agregamos que se puede especificar qué actividades desempeña cada rol, entonces este tipo de diagrama también puede ser usado para identificar el conocimiento que estos roles deben tener para poder desempeñar las actividades que tienen a su cargo. Adicionalmente, algunas dependencias de conocimientos pueden también ser identificadas por medio de este tipo de diagramas, por ejemplo, conocimientos que son generados en una determinada actividad, y que son requeridos en otras.

Para ejemplificar el uso de este tipo de diagramas, y siguiendo con el mismo escenario, consideremos que durante el proceso, el Responsable del Proyecto debe crear el documento del plan por medio de una herramienta específica y siguiendo un formato determinado. Dicho documento es posteriormente usado por el Jefe del Departamento, quien analiza la información y realiza un resumen ejecutivo de dicha información para el cliente. Durante la captura del proceso, identificamos que resulta importante que el Responsable del Proyecto conozca sobre el uso de la herramienta en la que se realiza el plan, así como el formato que debe seguir dicho plan. También, identificamos que la elección de la información del resumen ejecutivo, por lo general esta basado en la experiencia de quien lo realiza. El diagrama mostrado en la Figura 28 podría representar un modelo del escenario descrito.

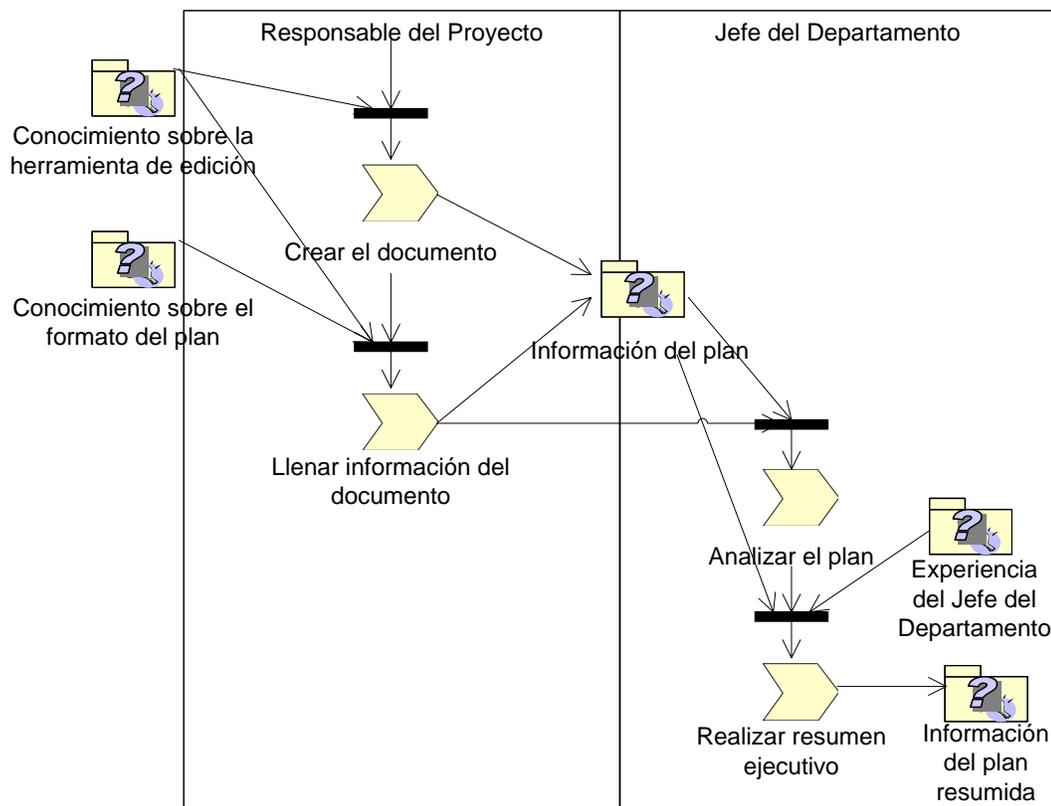


Figura 28. Ejemplo de un diagrama de actividad adaptado. Muestra los paquetes de conocimiento que son usados o generados en actividades específicas de un proceso. Las flechas indican el flujo.

Al observar el diagrama, podemos identificar que la información contenida en el plan resulta importante tanto para el Responsable del Proyecto, que requiere realizarlo, como para el Jefe del Departamento, que requiere analizarlo y resumirlo.

Supongamos ahora que el análisis del diagrama da la pauta para comenzar a buscar maneras de facilitarles las tareas a ambos roles, tanto en la generación del plan, como en la del resumen. Para esto, podríamos comenzar a identificar los temas específicos que componen cada paquete de conocimiento definido en el diagrama, las actividades donde se genera cada tema, o las fuentes de donde se obtiene la información que se almacena en el plan. De lo anterior, se podría posteriormente pensar en un sistema donde dicha

información pudiera estarse manejando de tal forma que conforme se genere, se vaya almacenando en un repositorio de donde la herramienta de edición la tome para generar de forma automática ciertas partes del plan siguiendo el formato definido. Así también, supongamos que por lo general, el Jefe del Departamento tiende a tomar cierto tipo de información de manera regular para elaborar sus resúmenes, esto podría ser útil para proponer una aplicación que le ayudara a elaborar estos resúmenes adelantándose para extraer dicha información que ya se encontraría almacenada con el formato que guarda el plan.

Del ejemplo anterior, podemos ver que para ayudar en el análisis de los flujos de conocimiento, una parte importante es identificar los temas específicos que constituyen cada paquete que se defina en los modelos, así como las fuentes de donde pueden ser obtenidos esos temas. Lo anterior puede realizarse por medio de diagramas que permiten clasificar y estructurar los tipos y fuentes de conocimiento.

IV.3.2.4.2 Clasificación y estructuración de tipos y fuentes de conocimiento

La clasificación y estructuración de tipos y fuentes de conocimiento se realiza por medio de dos tipos de diagramas: de paquetes de conocimientos y de clases. Los diagramas de paquetes de conocimientos son diagramas que muestran los sub-paquetes, o temas de conocimiento específicos que se encuentran en cada paquete de conocimientos definido en los distintos modelos. Por ejemplo, se podrían utilizar estos paquetes para definir categorías, constituidas por paquetes que representan áreas de conocimiento, las cuales se constituyen a su vez de sub-áreas o temas específicos.

Siguiendo el ejemplo presentado en el diagrama anterior (Figura 28), en la Figura 29 se presenta un ejemplo de diagramas de paquetes de conocimiento, que es usado para organizar la información relacionada con el plan. Según lo que se describió del escenario en el capítulo anterior, el plan debe contener información sobre cliente, el sistema a desarrollar, la funcionalidad esperada, descripción de las actividades, tiempo y costo estimados, personal y recursos asignados, etc.

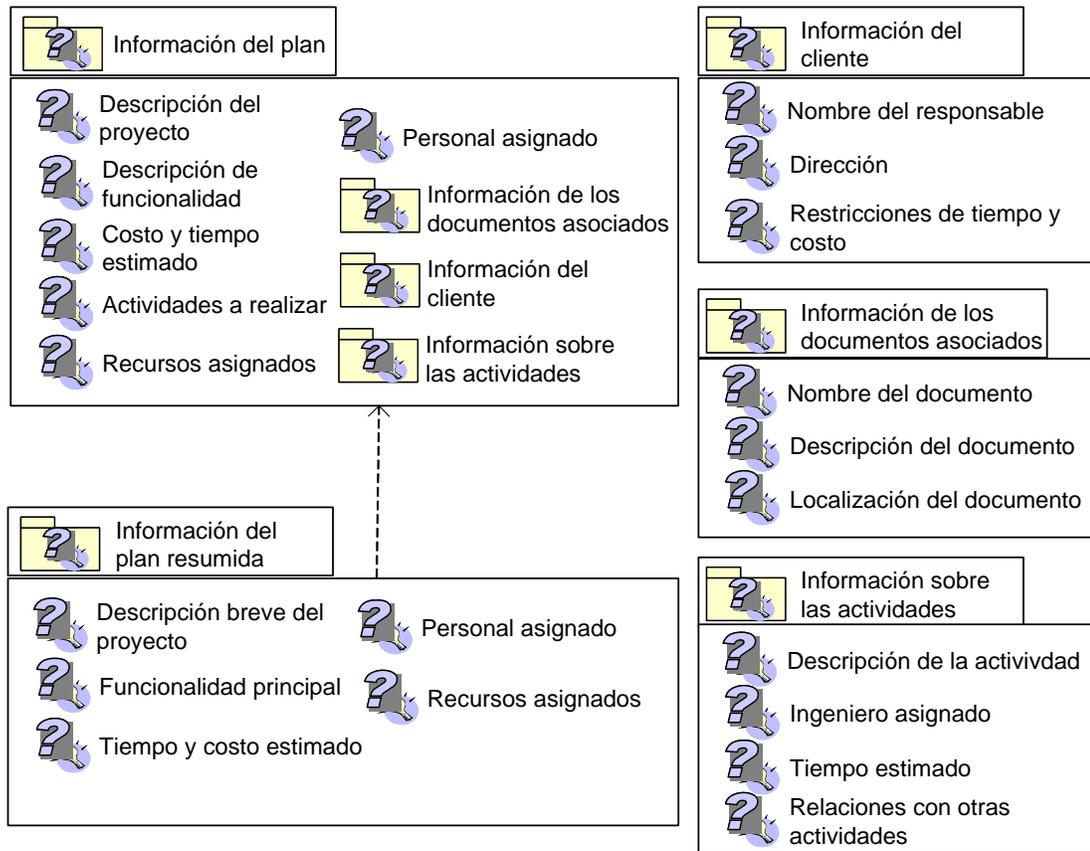


Figura 29. Ejemplo de diagramas de paquetes de conocimiento. Muestra los temas específicos así como sub-paquetes (áreas) de conocimiento que pertenecen a un paquete de conocimientos determinado. La flecha punteada indica dependencia entre paquetes.

Como se puede observar en el diagrama de la Figura 29, parte de la información del plan está agrupada en paquetes. También, es posible observar que se ha incluido la información contenida en el resumen del plan. Debido a que dicha información es obtenida del plan, se ha incluido una dependencia entre ambos paquetes para indicarlo.

Una vez identificados los principales temas de conocimiento que constituyen los paquetes de conocimiento que han sido identificados, el siguiente paso es identificar las fuentes que pueden ser usadas para obtener este conocimiento, así como la estructura que estas fuentes guardan. Lo anterior puede realizarse por medio de diagramas de clases, los

cuales muestran las relaciones de los distintos productos del trabajo y los roles encargados de los mismos, como el mostrado en la Figura 30.

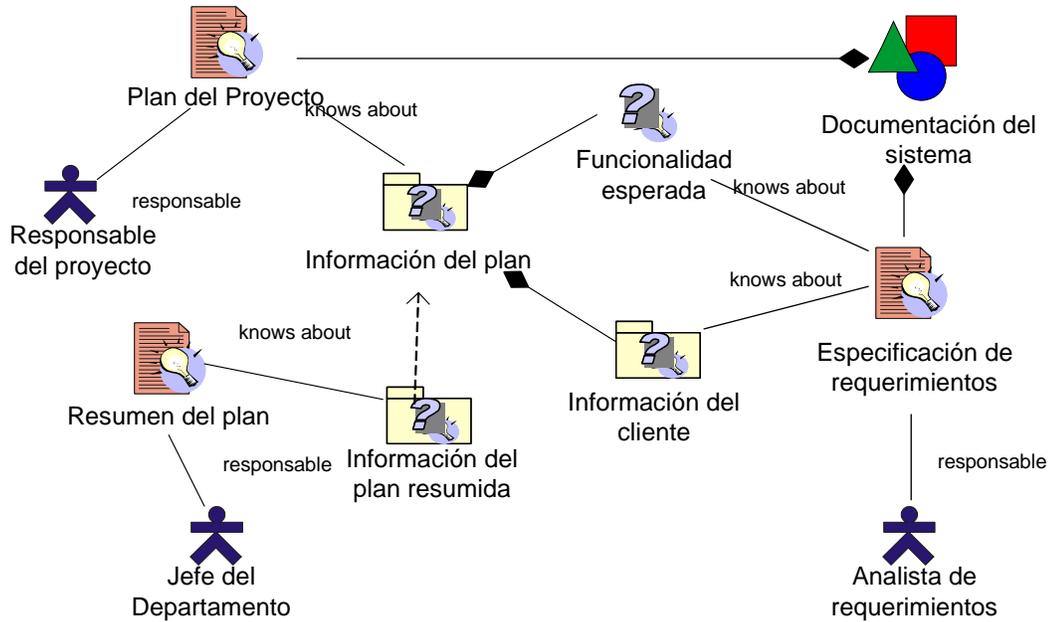


Figura 30. Ejemplo de diagrama de clases. Muestra las relaciones entre los roles, las fuentes y los paquetes de conocimiento.

En el ejemplo mostrado en la Figura 30 se identifican las principales fuentes donde se obtiene la información que posteriormente es almacenada tanto en el plan como en el resumen. Así también, se identifican a los responsables de dichos documentos que son definidos como fuentes de conocimiento. Por ejemplo, se observa que parte de dicha información es obtenida de la especificación de requerimientos, la cual es responsabilidad del analista de requerimientos. Es decir que el analista puede ser una fuente de información útil para esta actividad. Como se observa en el ejemplo, las fuentes se conectan a los temas o paquetes de conocimiento por medio de relaciones tipo *knowsAbout*. Adicionalmente, en estos diagramas se pueden comenzar a especificar categorías o tipos de fuentes con el fin de iniciar su clasificación, lo que es el caso de los documentos del plan del proyecto y la especificación de requerimientos, los cuales han sido agrupados como tipo de documentación del sistema.

En los diagramas que han sido presentados hasta este punto es posible identificar algunos flujos de conocimiento que se encuentran implícitos en los modelos, debido a que se describen conocimientos que son generados en una actividad y usados en otras, o conocimientos que son obtenidos de una determinada fuente, y guardados en otras. Sin embargo, para identificar detalles de estos flujos de conocimiento de forma explícita, se han propuesto un par de nuevos diagramas descritos a continuación.

IV.3.2.4.3 Identificación de flujos de conocimiento

De los dos tipos de diagrama propuestos para el análisis de flujos de conocimientos, el primero se centra en identificar las actividades a través de las cuales fluye el conocimiento de una fuente determinada. Por su parte, el segundo se enfoca a identificar el conocimiento que es transferido entre distintas fuentes.

Transferencia del conocimiento de una fuente específica entre actividades

La Figura 31 presenta un ejemplo del primer tipo de diagrama. Como se puede observar, el diagrama básicamente constituye un paquete dividido en tres partes que representan el conocimiento que la fuente aporta a una actividad, la actividad, y el conocimiento que la fuente obtiene de las actividades donde participa. Los temas o paquetes de conocimiento se conectan con las actividades por medio de flechas, cuya dirección, en el caso del conocimiento requerido va del conocimiento a la actividad, mientras que en el caso del conocimiento obtenido va de la actividad al tema o paquete específico. El diagrama se puede leer de la siguiente manera: si existe un tema en la parte denominada como requerido, significa que dicho conocimiento es requerido por la actividad a la que está conectado; si existe un tema en la parte de obtenido, indica que dicho conocimiento es obtenido de la actividad con la que está conectado.

Es importante mencionar que debido a que una fuente puede contener una gran variedad de temas específicos, podría ser más adecuado agrupar dichos temas para simplificar los diagramas, como en el caso del ejemplo anterior. Sin embargo, el nivel de detalle de los diagramas dependerá de las necesidades de quien utilice la nomenclatura

propuesta. Así mismo, este tipo de diagramas, de ser necesario, podrían construirse como una secuencia de niveles de abstracción. Por ejemplo, si en una determinada actividad o flujo de trabajo se genera un conjunto de conocimientos o información que es agrupada en un paquete. Posteriormente dicho paquete se podría detallar para definir las actividades específicas donde cada tema concreto es usado o generado.

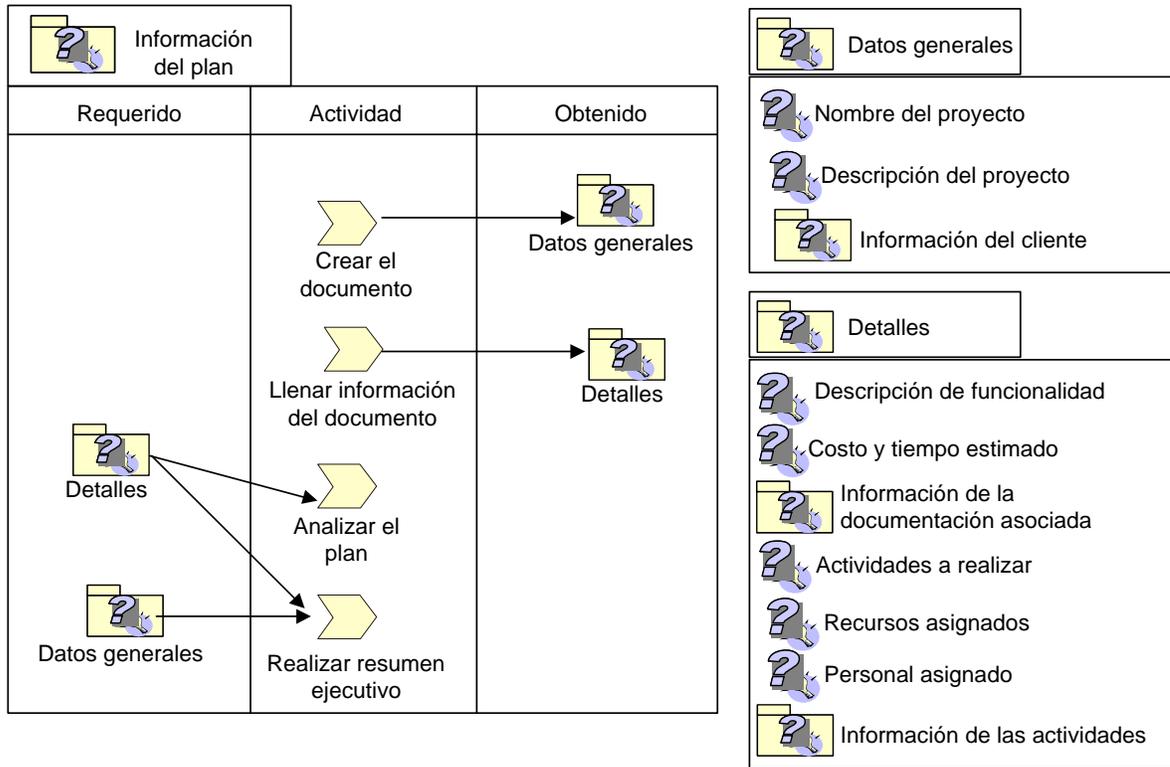


Figura 31. Ejemplo de diagrama del flujo del conocimiento de una fuente. Muestra el conocimiento e información que una fuente aporta u obtiene a lo largo de las actividades en las que interviene en el proceso.

En ocasiones puede ser importante modelar la forma en que una determinada fuente va cambiando conforme va siendo utilizada. Asumiendo que cada modificación a dicha fuente representa un estado, podrían usarse diagramas de estado para indicar dichos cambios, por ejemplo: fuente consultada, información almacenada, etc. De forma que se muestre claramente la secuencia que sigue la modificación de dicha fuente a lo largo del

proceso. En estos casos no se ha definido una adaptación específica, por lo que se sugiere el uso de los diagramas de estado tal cual y se define en la especificación de SPEM [OMG, 2002].

Transferencia de conocimiento entre distintas fuentes

El segundo tipo de diagrama de flujos de conocimiento se enfoca en ilustrar el conocimiento que es transferido entre distintas fuentes dentro de una actividad o flujo de trabajo determinado. Para lograr lo anterior, se hace uso de la relación *KnowledgeTransfer* definida en el metamodelo de conceptos de conocimiento y descrita anteriormente en este capítulo. Esta relación conecta a las fuentes que participan en la transferencia, con la actividad o flujo de trabajo donde ésta se da. Así mismo, cada fuente se conecta con el tema o paquete de conocimiento que aporta u obtiene durante la transferencia por medio de relaciones de tipo *KnowsAbout*. Para que el diagrama resulte más claro, las asociaciones pueden ser nombradas de la siguiente manera: *knows about* (o *conoce sobre* -la etiqueta predefinida para esta asociación-) para indicar la fuente que aporta el conocimiento, y *obtiene*, para indicar la fuente que obtiene el conocimiento. Adicionalmente, las asociaciones que conectan a cada fuente en la relación pueden también nombrarse de la siguiente manera: *inicia* para indicar la fuente que inicia la transferencia, y *participa* para el resto de las fuentes participantes en la transferencia.

La Figura 32 presenta un ejemplo del uso de los diagramas de transferencias de conocimientos. Este diagrama continúa con el ejemplo anterior, y muestra que durante las actividades realizadas para elaborar el plan del proyecto, se obtiene información que se encuentra en la especificación de requerimientos, esta información es almacenada en el plan, pero a su vez, debido a su participación en dichas actividades, tanto el Responsable del Proyecto con el Jefe del Departamento obtienen parte de esa información. El diagrama también muestra quien inicia la transferencia es el Responsable del Proyecto, debido a que es quien obtiene la información del repositorio de solicitudes y elabora el reporte, mismo que es posteriormente enviado al Jefe del Departamento. Del diagrama anterior podemos

observar que existen 4 fuentes de conocimiento participando en este flujo de trabajo, una de ellas aporta información, mientras que las otras tres la obtienen.

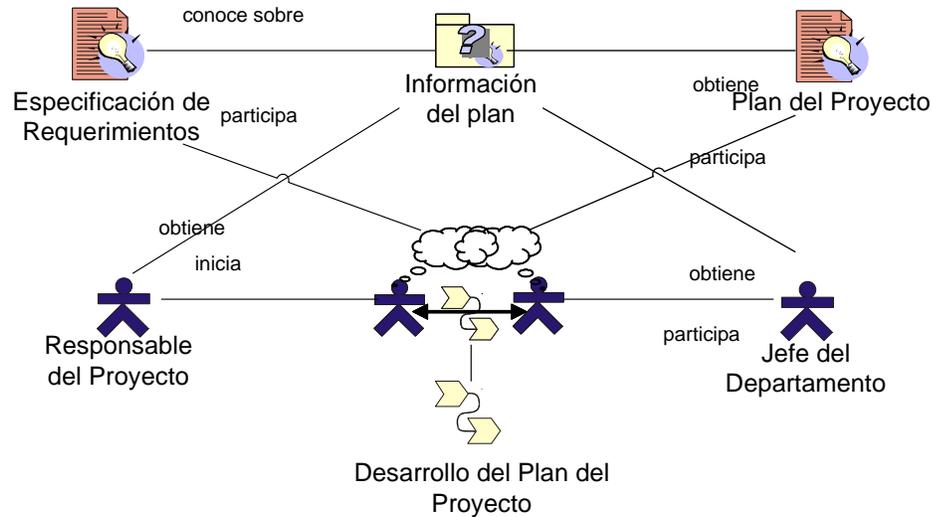


Figura 32. Ejemplo de diagrama de flujo de conocimiento entre fuentes. Muestra las fuentes que participan, el conocimiento que es transferido entre estas, y el flujo de actividades donde se lleva a cabo la transferencia.

Del ejemplo anterior podemos decir que el reporte sirve como mecanismo de transferencia de información entre los roles participantes. Sin embargo, esto no es tan claro en el diagrama. En caso de requerirse especificar este tipo de detalles, puede hacerse uso de diagramas de secuencia, donde se ilustre la secuencia de interacciones entre las distintas fuentes, de forma tal que se indique la secuencia en la cual se obtiene información de, o se almacena información en una fuente específica, indicando, por ejemplo, cuáles son los mecanismos que están apoyando en la transferencia de dicha información entre fuentes. La forma de elaboración de los diagramas de secuencia, al igual que con los de estado, no ha sido modificada en esta propuesta, por lo que se sugiere también seguir la especificación de SPEM para su elaboración [OMG, 2002].

Como punto final, es importante aclarar que el escenario presentado en esta y las secciones anteriores, es muy simple debido a que sólo ha sido tomado para ejemplificar el

uso de los diagramas propuestos. Pueden existir casos donde el proceso a analizar resulte tan complejo que se requiera de distintas vistas del proceso para lograr un buen entendimiento del mismo. Es este tipo de casos por los que se proponen los distintos tipos de diagramas descritos. La variedad de diagramas propuestos, como se ha ejemplificado, resulta útil para obtener tanto vistas estáticas como dinámicas del proceso; cada una de las cuales permite analizar aspectos distintos del mismo, tales como:

- la estructura de actividades,
- los principales temas, áreas o grupos de conocimientos involucrados, así como su estructura,
- las fuentes donde los conocimientos son obtenidos o almacenados, así como su estructura y relaciones,
- la forma en que fluye el conocimiento de una fuente específica a lo largo de las actividades en las que participa, y
- la forma en que fluye el conocimiento entre diversas fuentes a lo largo de las actividades del proceso.

Al igual que con la especificación básica de SPEM [OMG, 2002], los diagramas propuestos pueden ser usados tanto para definir modelos genéricos (por ejemplo familias de procesos) como de situaciones específicas. Esto último quedará más claro con el caso de estudio presentado en el Apéndice B, y resumido en el Capítulo VI; el cual ha sido realizado para mostrar el uso de la metodología y de las técnicas que la componen.

IV.4 Recomendaciones en el uso de los diagramas

Con el objetivo de facilitar el uso de los diagramas durante la utilización de la metodología, en esta sección se dan una serie de recomendaciones sobre cuándo y cómo usar cada diagrama, buscando que dichas recomendaciones ayuden durante el modelado del proceso y análisis de flujos de conocimiento. Se propone usar los diagramas en el orden de la Tabla V, y siguiendo las recomendaciones para cada tipo de diagrama.

Tabla V. Recomendaciones en el uso de los distintos tipos de diagramas.

Diagrama	Recomendaciones
Gráfica Rica	<ul style="list-style-type: none"> • Usarlos cuando sólo se quiera obtener una vista global del proceso, sus fuentes y el conocimiento involucrado; o antes de iniciar el modelado detallado, cuando no se tenga una visión clara del proceso, con el fin de primero identificar las principales actividades, las principales fuentes involucradas, y los principales temas o áreas de conocimiento. • Sólo incluir los temas o áreas de conocimiento más relevantes a cada elemento del modelo, con el fin de no generar modelos muy grandes y complejos. Los detalles deberán realizarse posteriormente. • Cuando el proceso sea muy grande, utilizar varias gráficas ricas para dividir el proceso en secciones más manejables.
Casos de uso	<ul style="list-style-type: none"> • Usarlos al inicio del modelo detallado, con el fin de identificar las principales actividades del proceso y los principales paquetes de conocimiento requeridos o generados en estas, buscando que dichos paquetes sean definidos de forma tal que posteriormente sean la base para la clasificación de los temas o áreas de conocimiento. • Los casos de uso permiten dividir el proceso en fases, etapas y flujos de actividades, de esta forma el proceso se puede ir viendo a distintos niveles de detalle. Así mismo, es importante identificar paquetes de conocimiento en los distintos niveles, de forma que estos posteriormente sean usados para definir el esquema de clasificación de conocimientos.
Paquetes de conocimiento	<ul style="list-style-type: none"> • Usarlos cada vez que se defina un nuevo paquete de conocimiento para especificar sub-paquetes, temas o áreas concretas que contiene. • Se deberán ir actualizando los diagramas de paquetes conforme se avance en el modelado, de forma que se vayan incluyendo los temas que sean identificados en los demás diagramas.
Clases (relaciones entre paquetes y fuentes)	<ul style="list-style-type: none"> • Usarlos para especificar fuentes relacionadas con cada paquete de conocimiento. • En una primera instancia, sólo será necesario definir la relación entre las fuentes y el paquete, pero conforme se avance en el modelado, será necesario especificar los temas o áreas concretas dentro del paquete con los cuales se relaciona cada fuente.
Actividades	<ul style="list-style-type: none"> • Usarlos para detallar la secuencia de actividades de los flujos de trabajo identificados en los casos de uso. Permiten también identificar las actividades que realiza cada uno de los roles que participa en un determinado flujo de trabajo. • Si el conocimiento generado o utilizado en las actividades reside en una fuente explícita, utilizar dicha fuente como el objeto que fluye a lo largo de las actividades, de forma que sea claro que dicha fuente es dónde se obtiene o almacena el conocimiento involucrado. • Si el conocimiento que se genera o utiliza no se encuentra en una fuente explícita, o éste se encuentra distribuido en varias fuentes, utilizar un paquete de conocimientos como el objeto que fluye entre las actividades. Esto ayudará posteriormente a fijar la atención en aquellos conocimientos que podrían estarse perdiendo debido a que no son almacenados en ningún lugar, o en su defecto, conocimiento que podría ser difícil de obtener debido a que se encuentra distribuido en diversas fuentes. • En el último caso, deberá especificarse en los diagramas de clases las distintas fuentes dónde residen cada uno de los temas o áreas que conforman el paquete. • De ser necesario, se podrán incluir en los diagramas temas de conocimiento concretos, dando más realce al conocimiento específico requerido o generado. Sin embargo, deberá incluirse, con el diagrama de actividad, el diagrama del paquete o paquetes donde dichos temas están agrupados. En la medida de lo posible, es preferible usar paquetes, dado que el uso de temas concretos podría generar diagramas con demasiados elementos, haciéndolo difícil de seguir y entender.

Tabla V. Recomendaciones en el uso de los distintos tipos de diagramas (continuación).

Diagrama	Recomendaciones
Clases (estructura de fuentes de conocimiento)	<ul style="list-style-type: none"> • Usar los diagramas de clases para definir la estructura de las fuentes de conocimiento, es decir, la forma en que están clasificadas, los roles responsables de las mismas, así como los paquetes que agrupan el conocimiento existente en dichas fuentes. • Se pueden usar elementos del tipo producto del trabajo (WorkProduct) para definir las categorías y tipos de fuentes de conocimiento.
Clases (estructura de paquetes de conocimiento)	<ul style="list-style-type: none"> • Usar los diagramas de clases para definir la estructura de los temas y áreas de conocimiento, identificando las relaciones entre estos, por ejemplo, paquetes de conocimiento que contienen sub-paquetes, o dependencias entre áreas o temas de conocimiento (un tema o área que se requiere para poder aplicar otro). • Estos diagramas deberán mostrar también las fuentes relacionadas con cada paquete, área o tema concreto, sobre todo en el caso de que los temas agrupados en un paquete se encuentren en distintas fuentes.
Transferencia de conocimiento (de una fuente)	<ul style="list-style-type: none"> • Usarlos una vez modelado el proceso completo, con el fin de definir de forma explícita la forma en que el conocimiento de una fuente específica fluye a lo largo de las actividades donde es usada. • Este tipo de diagrama deberán crearse sobre todo para aquellas que sean identificadas como mecanismos de transmisión de conocimiento en el proceso.
Transferencia de conocimiento (entre fuentes)	<ul style="list-style-type: none"> • Usarlos para resaltar los flujos de conocimiento más relevantes, buscando especificar las fuentes que participan en el flujo, así como el conocimiento que aporta y obtiene cada fuente. • Deberá intentarse que estos diagramas muestren los posibles problemas relacionados con el flujo, por ejemplo, conocimiento que no es almacenado en medios formales, o cuellos de botella, donde una persona se satura debido a que es constantemente consultada por diversos actores del proceso. Este tipo de observaciones pueden agregarse al diagrama como notas.

IV.5 Resumen del capítulo

En este capítulo se han presentado dos técnicas que han sido propuestas para apoyar la aplicación de la metodología KoFI, en la etapa del modelado de procesos. Estas técnicas están encaminadas a apoyar dos aspectos distintos del modelado del proceso. En primer lugar, el modelado se da a un nivel muy general y abstracto, con el fin de servir para capturar el proceso de forma que puedan ser identificados, desde una etapa temprana, los principales tipos y fuentes de conocimiento involucrados en las actividades generales del proceso. Para este primer aspecto se ha propuesto el uso de una técnica genérica y flexible. El segundo aspecto es el análisis detallado del proceso, para lo cual se ha propuesto una

adaptación a un lenguaje de modelado formal y orientado a procesos de ingeniería del software. Ambas técnicas han sido explicadas y ejemplificado su uso para modelar procesos incluyendo de forma explícita el conocimiento y sus fuentes involucrados en las actividades que los constituyen.

La elección de ambas técnicas se basó, en el primer caso, en que es una técnica ampliamente usada y probada, y en el segundo, debido a su relación con los procesos de ingeniería del software, que representan el enfoque de aplicación principal de la metodología KoFI. Esta relación puede hacer suponer que el lenguaje podría ser fácil de interpretar para personas con conocimientos en el área del desarrollo del software. Con el fin de darnos una idea de qué tanto es cierto esto último, se ha realizado un experimento que es descrito en el Capítulo VI de esta tesis, y presentado en el Apéndice A de este documento.

En este trabajo no se ha realizado una evaluación para determinar qué técnica de modelado de procesos es más adecuada. Esto debido a las diferencias existentes entre cada técnica, dado que han sido diseñadas para propósitos distintos, por lo que una comparación entre ellas resultaría muy probablemente insustancial. Además, los resultados de la aplicación de los lenguajes de modelado de procesos dependen en gran medida de las habilidades de las personas que los utilizan, ya sea quienes realizan los modelos o quienes los analizan. La literatura que presenta este tipo de propuestas se limita a ejemplificar el uso de las mismas por medio de casos de estudio; ver, por ejemplo: [Bera et al., 2005; Kim et al., 2003; Nissen y Levitt, 2004; Strohmaier y Tochtermann, 2005; Woitsch y Karagiannis, 2002; Zhuge, 2002]. Por lo anterior, para mostrar la utilidad de las propuestas planteadas en este capítulo, con respecto a su apoyo a las etapas de la metodología KoFI, en el Capítulo VI se presenta un resumen de un caso de estudio, mismo que se detalla en el Apéndice B. Sin embargo, antes de presentar este caso de estudio, en el siguiente capítulo se describe otro de los elementos de apoyo definidos para la metodología KoFI, el cual es un marco de trabajo para apoyar en la identificación de herramientas de soporte al flujo del conocimiento.

CAPÍTULO V

MAHAC: Un marco de trabajo para el análisis de herramientas de apoyo a la administración del conocimiento

"Los trabajadores del conocimiento en la actualidad pueden crear, compartir y usar información y conocimiento casi en cualquier lugar y en cualquier momento". [Davenport, 2007]

Como se ha descrito en capítulos anteriores en este documento, la fase final de la metodología consiste en analizar el papel que pueden jugar como habilitadoras del flujo del conocimiento, las herramientas que apoyan el trabajo que realizan los involucrados en el proceso en estudio. Con el fin de apoyar esta fase, se definió un marco de trabajo, denominado MAHAC, que define una serie de pasos a seguir para identificar las características de las herramientas a estudiar, desde el punto de vista de su aporte al flujo del conocimiento durante el proceso. El marco de trabajo es presentado en este capítulo. El resto del capítulo está organizado de la siguiente manera: primero se da una descripción de las principales características a tomar en cuenta al analizar herramientas de apoyo a la AC. Posteriormente se define el marco de trabajo en lo general, para finalmente detallar cada una de las fases a seguir durante la aplicación del mismo.

V.1 Características para el análisis de herramientas de administración del conocimiento

En la literatura existen diversas propuestas encaminadas a servir de apoyo para clasificar, comparar, evaluar o analizar herramientas o sistemas de AC. De estos trabajos, se desprenden un conjunto de aspectos que deben ser considerados al analizar sistemas de AC. Para la obtención de las características utilizadas para definir MAHAC, se han tomado principalmente enfoques prácticos, más que teóricos. Una revisión de marcos de trabajo

teóricos sobre las actividades de la AC puede ser encontrado en [Holsapple y Joshi, 1999]. Por otra parte, MAHAC no pretende ser una herramienta de evaluación, aunque podría servir de apoyo para este fin, como se mencionará más adelante. Para consultar enfoques de evaluación de sistemas de AC existen los siguientes trabajos [Folkens y Spiliopoulou, 2004; Kankanhalli y Tan, 2004; Smith y McKeen, 2005].

La definición de las características consideradas en MAHAC, así como sus fases han sido definidas tomando en cuenta, principalmente, propuestas generales sobre análisis de sistemas de AC. Sin embargo, también se han revisado propuestas desde otros enfoques, tales como la clasificación de herramientas para dominios de aplicación específicos, particularmente en el área de Ingeniería de Software que es nuestro enfoque principal [Lindvall y Rus, 2003; Rus *et al.*, 2001]; así como para enfoques para el análisis de tipos específicos de herramientas de AC, en particular sistemas colaborativos y de apoyo al trabajo en grupo [Qureshi *et al.*, 2004; Vizcaíno *et al.*, 2005], tecnologías Web [Tiwana y Ramesh, 2001] y sistemas de AC basados en agentes de software [van Elst *et al.*, 2003].

De la revisión de los trabajos mencionados se obtuvieron cuatro aspectos principales a considerar para el desarrollo de MAHAC: 1) el dominio de aplicación de las herramientas, 2) la estructura y formato del conocimiento manejado, 3) las actividades, servicios o procesos de AC soportados, y 4) aspectos técnicos que podrían ser considerados importantes en situaciones particulares.

En la Tabla VI se presentan los distintos trabajos que fueron revisados y considerados en el desarrollo de MAHAC. En ella se muestran el propósito para el cual cada propuesta fue desarrollada o usado, el enfoque que cada propuesta aplica para dicho propósito, y cuáles de los cuatro aspectos antes mencionados son cubiertos por dichas propuestas. Los aspectos cubiertos por cada enfoque están ordenados de acuerdo a la importancia dada en cada uno. Es importante mencionar que cada enfoque maneja estos aspectos desde distintos puntos de vista y con diferentes propósitos. Incluso, ciertas propuestas sólo los incluyen de forma implícita. El único aspecto considerado por la mayoría de las propuestas es el de las actividades de AC. Por su parte, el único trabajo que

parcialmente considera los cuatro aspectos es el de Woitsch y Karagiannis [2002]. Sin embargo, en ese trabajo toda la información es manejada en un vector, y no se distingue de forma explícita un aspecto de otro en la misma profundidad que se hace en MAHAC. Los cuatro aspectos identificados son la base para la definición de MAHAC, y son descritos a continuación.

Tabla VI. Propuestas para comparar, caracterizar o clasificar sistemas de AC.

Trabajo	Propósito	Enfoque	Aspectos cubiertos
[Borghoff y Pareschi, 1998]	Clasificar tecnologías de AC	Una arquitectura para memorias organizacionales de cuatro componentes. Se clasifican las tecnologías dependiendo de los componentes que implementan. Los componentes considerados son: el flujo del conocimiento, comunidades de trabajadores del conocimiento, repositorios y librerías de conocimiento, y cartografía del conocimiento.	- Actividades de AC - Estructura del conocimiento (tácito y explícito)
[Newman y Conrad, 2000]	Organizar y clasificar métodos, prácticas tecnologías de AC	Una tabla que puede ser usada como lista de chequeo para especificar algunos conceptos relacionados como los métodos, prácticas o tecnologías de AC. Los conceptos considerados son: actividades de AC, nivel de actividad de trabajo (proceso de alto nivel, de medio nivel, decisión o acción), tipo de agente (individual, organización, automatizado), tipo de artefacto de conocimiento (explícito, implícito, tácito), y el enfoque (agente, artefacto, proceso)	- Actividades de AC - Estructura del conocimiento (tácito, implícito y explícito) - Dominio de aplicación (uso, alcance, y dominio del conocimiento)
[Alavi y Leidner, 2001]	Analizar el rol de los sistemas de AC en la AC organizacional	Un conjunto de procesos de AC. Se clasifican los sistemas de AC dependiendo de los procesos de AC que apoyan.	- Actividades de AC
[Binney, 2001]	Clasificar aplicaciones de AC mencionadas en la literatura	Un “espectro de AC” que establece un conjunto de tipos de aplicaciones de la AC. Se clasifica cada aplicación dependiendo de la actividad de AC en la cual es más citada en la literatura. Este enfoque es usado y extendido en [Haggie y Kingston, 2003]	- Dominio de aplicación (uso del conocimiento) - Actividades de AC
[Marwick, 2001]	Describir las características de las tecnologías de AC	Los mecanismos de conversión de conocimiento propuestos por Nonaka [1991]. Se clasifican y describen tecnologías de AC dependiendo de si apoyan la conversión de conocimiento tácito a tácito, de tácito a explícito, de explícito a explícito, o de explícito a tácito.	- Actividades de AC - Estructura del conocimiento (tácito y explícito)
[Rus et al., 2001]	Clasificar herramientas de AC usadas en la ingeniería del software	Procesos y actividades organizacionales que las herramientas apoyan. También se consideran las estrategias de AC desde el punto de vista de su alcance, particularmente, si están enfocadas a los individuos, proyectos, organización, o industria.	- Dominio de aplicación (uso y alcance del conocimiento)

Tabla VI. Propuestas para comparar, caracterizar o clasificar sistemas de AC (continuación).

Trabajo	Propósito	Enfoque	Aspectos cubiertos
[Tiwana y Ramesh, 2001]	Clasificar tecnologías de AC y describir algunos sistemas de AC basados en el Web	Una “red de AC” compuesta por un conjunto de fuentes de conocimiento, herramientas y actividades. Se clasifican las tecnologías de AC de acuerdo a los procesos organizacionales que apoyan, o los objetivos de AC que permiten lograr. También se consideran algunos atributos técnicos que son importantes para sistemas de AC basados en el Web.	<ul style="list-style-type: none"> - Dominio de aplicación (uso del conocimiento) - Actividades de AC - Aspectos técnicos (específicos para sistemas Web)
[Valente y Housel, 2002]	Analizar y comparar herramientas de AC	Una matriz de servicios de AC y tipo de estructura del conocimiento. Se especifica en la matriz los servicios que apoya la herramienta, y el tipo de estructura del conocimiento que maneja. La estructura del conocimiento considera niveles dentro del rango de no estructurado a estructurado. Los servicios consideran aquellos requeridos para lograr los procesos organizacionales (de trabajo y administrativos), y de AC.	<ul style="list-style-type: none"> - Actividades de AC - Estructura del conocimiento (un conjunto de niveles desde conocimiento altamente no estructurado hasta formalmente estructurado) - Dominio de aplicación (uso del conocimiento)
[Woitsch y Karagiannis, 2002]	Definir la funcionalidad requerida por sistemas de AC empresariales	Un vector de “dimensiones de AC” que constituyen un conjunto de propiedades o atributos del conocimiento manejado. Algunas de estas son: representación, medio, usuario, tiempo, origen, sofisticación, ciclo de vida, relevancia, aplicabilidad, nivel, dinámico, expresión, límites del servicio, abstracción, acción y estructura.	<ul style="list-style-type: none"> - Actividades de AC - Dominio de aplicación (uso y alcance del conocimiento) - Estructura del conocimiento (tácito, explícito / no estructurado a estructurado) - Aspectos técnicos (un conjunto de atributos para el conocimiento específico que es manejado)
[Kankanhalli <i>et al.</i> , 2003]	Analizar el rol de las TI en la AC en un conjunto de organizaciones	Un marco de cuatro dimensiones que considera si un sistema es orientado a servicios o productos, y si permite manejar un contexto alta o bajamente volátil, es decir, un medio ambiente que cambia rápido o lento.	<ul style="list-style-type: none"> - Dominio de aplicación (uso del conocimiento)

Tabla VI. Propuestas para comparar, caracterizar o clasificar sistemas de AC (continuación).

Trabajo	Propósito	Enfoque	Aspectos cubiertos
[van Elst et al., 2003]	Analizar y clasificar enfoques de AC basados en agentes de software	Un marco tridimensional donde las primeras dos dimensiones consideran aspectos técnicos de los sistemas basados en agentes, y la tercera, llamada nivel de aplicación de AC, se centra en el área de aplicación de los sistemas. Aquí se consideran el alcance del sistema, las actividades de AC apoyadas, y si el sistema se orienta a servicios o productos.	- Actividades de AC - Dominio de aplicación (uso del conocimiento)
[Qureshi et al., 2004]	Analizar herramientas de soporte a trabajo en grupos como tecnologías de AC	Un conjunto de características de tecnologías groupware para clasificarlas e ilustrar los procesos de AC que apoyan.	- Actividades de AC - Dominio de aplicación (uso del conocimiento) - Aspectos técnicos (relacionados con herramientas de apoyo al trabajo en grupo)
[Banerjee, 2005]	Analizar la forma en que son usadas las herramientas de AC en las organizaciones	Procesos de AC y alcance del negocio que apoyan las herramientas. En el alcance del negocio se considera el contexto de la organización, la distribución de quienes aportan y usan el conocimiento, y el tipo de acceso temporal al conocimiento (síncrono/asíncrono)	- Actividades de AC - Dominio de aplicación (uso del conocimiento) - Aspectos técnicos (distribución de las fuentes y temporalidad en el acceso al conocimiento)
[Rao, 2005a]	Clasificar y describir las características de las tecnologías de AC	Distintos enfoques: espacio-tiempo, actividades de AC, los mecanismos de conversión de conocimiento de Nonaka [Nonaka, 1991], y el dominio de aplicación y alcance de las herramientas.	- Actividades de AC - Dominio de aplicación (uso del conocimiento) - Aspectos técnicos (características del conocimiento y tipos de herramientas específicas; las mismas que [Banerjee, 2005])
[Vizcaíno et al., 2005]	Evaluar la forma en que los sistemas colaborativos soportan actividades de AC	Un conjunto de preguntas cuyas respuestas pueden ayudar a identificar si una herramienta apoya actividades de AC específicas.	- Actividades de AC

V.1.1 Definir el Dominio de aplicación

Las herramientas y tecnologías de AC deben considerar lo que es importante para los trabajadores del conocimiento [Dalkir, 2005: p. 21], ya que las estrategias de AC deben estar orientadas a facilitar las actividades que tales conocimientos deben realizar [Wiig, 2004]. Las estrategias de AC deben identificar dónde está el conocimiento y dónde será aplicado, por ejemplo, los usuarios del conocimiento, o los procesos en los cuales es usado [Alavi y Tiwana, 2002; Haggie y Kingston, 2003]. Los sistemas de AC que requiere una organización dependen de aspectos relacionados con el tipo de organización de que se trate [Kankanhalli et al., 2003]. Por lo tanto, el dominio de aplicación de los sistemas de AC resulta una característica clave en el análisis de tales sistemas. Identificar el dominio de aplicación es también un paso importante para la evaluación de sistemas de AC, lo cual incluye identificar cómo el conocimiento es usado, dónde, cuándo, por quién, etc. [Smith y McKeen, 2005]. Junto con lo anterior, tomar en cuenta los procesos de trabajo y las personas que pueden resultar beneficiadas o afectadas por el uso de sistemas de AC sirve también para definir el contexto de la organización, lo cual es un aspecto importante para saber si una herramienta está alineada con las estrategias o procesos de una organización [Bleistein *et al.*, 2006].

V.1.2 Definir el tipo y estructura del conocimiento

Definir el tipo y estructura del conocimiento se refiere a identificar la clase de conocimiento manejado y la manera en que éste está estructurado. El conocimiento en una organización puede ser clasificado desde distintos puntos de vista [Alavi y Leidner, 2001]. La clasificación más común en la literatura sobre AC es la definida en [Nonaka y Takeuchi, 1995], la cual considera al conocimiento de dos formas: tácito y explícito. La mayoría de los sistemas de AC se centran en el manejo de conocimiento explícito [Aurum *et al.*, 2003; Borghoff y Pareschi, 1998; Rao, 2005b; Tiwana, 2000]. Este tipo de conocimiento puede estructurarse a varios niveles de formalidad, en los cuales entre más formal esté definido, más fácil es manejarlo por medios automatizados [Valente y Housel, 2002]. De todo lo

anterior, se deriva que el tipo y estructura del conocimiento es una característica importante a tomar en cuenta al analizar sistemas de AC.

V.1.3 Definir actividades de AC o etapas del flujo de conocimiento apoyadas

El principal objetivo de la AC es facilitar el flujo del conocimiento de donde es creado o almacenado, hacia donde es aplicado [Borghoff y Pareschi, 1998], en este trabajo se considera a las actividades de AC como las distintas etapas requeridas para facilitar este flujo, por ejemplo, capturando conocimiento para permitir su uso futuro, o proveer métodos a través de los cuales facilitar la aplicación del conocimiento de la organización. La principal característica que ha sido usada en trabajos previos para analizar sistemas de AC, es precisamente las actividades de AC específicas que apoyan dichos sistemas. De hecho, la mayoría de los trabajos estudiados para definir MAHAC consideran este aspecto como la principal característica para analizar o evaluar sistemas de AC. Sin embargo, la mayoría de los enfoques existentes consideran actividades de AC diferentes.

Existen aspectos que pueden ser importantes para actividades de AC específicas, por ejemplo, en el caso de la transferencia del conocimiento puede ser importante considerar una dimensión espacio-temporal [Banerjee, 2005], es decir, especificar si el sistema considera la localidad o distribución de quienes aportan o usan el conocimiento, y si estos sistemas permiten acceder al conocimiento de forma síncrona o asíncrona.

V.1.4 Definir los principales aspectos técnicos

Algunos SACs tienen limitaciones técnicas que dificultan su uso en algunos dominios [Bonifacio *et al.*, 2002b; van Elst *et al.*, 2003]. Por lo tanto, en estos casos es importante tener en consideración esos aspectos técnicos que pudieran resultar relevantes. Por ejemplo, en las organizaciones de software se ha observado que uno de los problemas que obstaculizan el uso de sistemas de AC tradicionales, es que requieren que los usuarios consuman mucho tiempo capturando o buscando conocimiento [Desouza, 2003; Lindvall y Rus, 2003]. Por lo tanto, una vez definido el dominio de aplicación de un sistema de AC, el

tipo de conocimiento que maneja, la manera en que apoya a las distintas actividades de AC, el siguiente paso es identificar los principales aspectos técnicos que deben ser considerados, así como la forma en que la herramienta los suporta. Siguiendo estos cuatro aspectos se ha definido la estructura general de MAHAC, presentada a continuación.

V.1.5 Estructura general de MAHAC

La estructura general de MAHAC se muestra en la Figura 33. Como se puede observar, MAHAC se compone de cuatro etapas obtenidas del estudio de literatura presentado previamente: 1) identificar el dominio de aplicación de las herramientas a analizar, 2) identificar la estructura del conocimiento manejado, 3) identificar las actividades de AC soportadas, y 4) identificar los principales aspectos técnicos que puedan ser relevantes. Cada una de estas etapas se describe a continuación en las siguientes secciones.

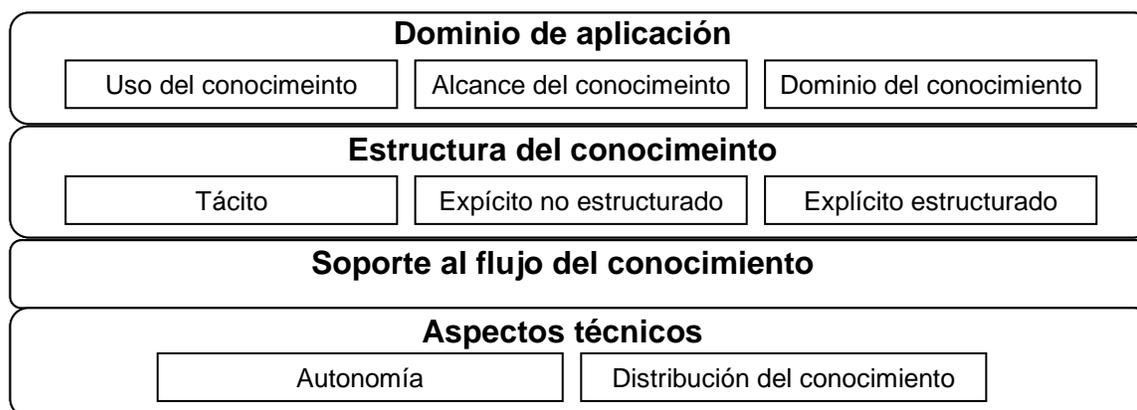


Figura 33. Etapas del marco de trabajo para el análisis de herramientas de apoyo a la administración del conocimiento.

V.2 Dominio de aplicación

La definición del dominio de aplicación se enfoca en tres aspectos: 1) el uso del conocimiento, lo cual se relaciona con las actividades o procesos de la organización donde la herramienta es usada; 2) el alcance del conocimiento, relacionado con las personas que

podrían resultar beneficiadas del uso de la herramienta; y 3) el dominio del conocimiento manejado, que define el área a la que pertenece el conocimiento que maneja la herramienta. A continuación se detalla cada uno de estos aspectos.

V.2.1 El uso del conocimiento

Un sistema de información puede apoyar distintas actividades, algunas comunes a cualquier organización, como relaciones con los clientes, habilidades y competencias de la administración, etc. Otras son específicas a cierto tipo de organizaciones, tales como las herramientas CASE o sistemas de administración de la configuración del software que han sido diseñadas para organizaciones de desarrollo de software. Con el fin de ayudar a definir el o los procesos para los cuales una herramienta de AC será de utilidad, se ha definido un conjunto de categorías mostrado en la Tabla VII. Sin embargo, es recomendable describir explícitamente qué procesos y actividades son soportadas por la herramienta que se analice.

Tabla VII. Esquema para clasificar los procesos y actividades apoyados por las tecnologías de AC.

Categoría	Descripción
Estrategias de negocio	Procesos y actividades que ayudan a cumplir las estrategias de negocio, tales como herramientas para el manejo de conocimiento sobre el mercado, competidores, etc.
Administración de la organización	Procesos y actividades para administrar la organización, tales como administración de recursos humanos, financieros, etc.
Vida de la organización	Procesos y actividades relacionados con el trabajo diario en la organización. Por ejemplo, manejo del inventario, infraestructura de la organización, etc.
Procesos del dominio de trabajo	Procesos y actividades relacionados con el dominio del trabajo de la organización, tales como las directamente relacionadas con los productos o servicios proveídos por la compañía.
Actividades técnicas	Procesos y actividades relacionados con aspectos específicos de los procesos del dominio. Por ejemplo, herramientas usadas para el desarrollo de los productos o para la prestación de los servicios.

V.2.2 Alcance del conocimiento

El alcance del conocimiento se refiere al rango de personas que usan o podrían resultar beneficiadas de usar el conocimiento manejado por una herramienta de AC. Algunas herramientas de AC pueden ser de uso personal [Teevan *et al.*, 2006], otras pueden dar soporte a pequeños o grandes grupos de trabajadores, por ejemplo, a un grupo dentro de la organización (un departamento o división), a grupos formados por personas dentro y fuera de la organización; incluso, el conocimiento podría ser útil para toda una organización, o más aun, para toda un área de la industria [Davenport, 2007; Rus *et al.*, 2001]. La Tabla VIII presenta un conjunto de niveles que pueden ser usados para definir el alcance de una herramienta; este conjunto es una extensión del utilizado por Rus *et al.* [2001].

Tabla VIII. Estructura de niveles de granularidad para definir el alcance de una herramienta de AC.

Nivel	Definición
Personal	El conocimiento es útil para personas específicas en una forma individual. Por ejemplo, un repositorio de documentos personales en el ambiente de trabajo de un empleado de la organización.
Grupo Intra-Organizacional	El conocimiento es útil para un grupo de personas bien definido dentro de la organización. Por ejemplo, ciertos roles o estratos de la organización, un departamento, una división, etc.
Grupo Extra-Organizacional	El conocimiento es útil para un grupo de personas dentro y fuera de la organización. Por ejemplo, una comunidad de expertos en la que algunos empleados participen.
Organización	El conocimiento es útil para toda la organización.
Múltiples organizaciones	El conocimiento es útil también para otras organizaciones. Por ejemplo, los clientes o proveedores.
Nivel industrial	El conocimiento puede ser útil para toda organización del mismo giro o en un área de influencia similar.

V.2.3 El dominio del conocimiento

El dominio del conocimiento manejado por una herramienta específica está directamente relacionado con los procesos y actividades para los cuales la herramienta es usada. Sin embargo, es importante describir explícitamente el dominio del conocimiento o información. Por ejemplo, si la herramienta maneja información sobre los productos o servicios provistos por la organización, sobre sus empleados, sus clientes, etc. La Tabla IX presenta un esquema de clasificación que puede ser usado en este caso. No obstante, una vez clasificado el dominio del conocimiento, es importante describirlo explícitamente.

Tabla IX. Esquema para clasificar el dominio del conocimiento.

Tipo de conocimiento	Definición
De negocios	Conocimiento relacionado con las estrategias de negocio, tales como conocimiento de los clientes, proveedores, del mercado, etc.
Organizacional	Conocimiento relacionado con la operación del negocio, como su estructura, recursos materiales o humanos, etc.
Administrativo	Conocimiento relacionado con aspectos de la administración de la organización, como planeación o administración de proyectos.
Producto-Servicio	Conocimiento relacionado con aspectos específicos de los productos o servicios de la organización. Por ejemplo, los procesos de producción, productos intermedios, etc.
Técnico	Conocimiento requerido para realizar las actividades productivas de la organización, tales como metodologías, uso de herramientas, etc.

V.3 Estructura del conocimiento

Para definir la estructura del conocimiento manejado, se ha definido un enfoque que extiende el propuesto por Valente y Housel [2002], presentado en la Tabla X. En esta tabla, el conocimiento es dividido en distintas formas, basándonos en las dimensiones tácita y explícita. Según Nonaka [1991], el conocimiento tácito tiene dos dimensiones: la técnica, que se refiere al conocimiento usado por las personas para realizar sus actividades; y la cognoscitiva, que corresponde al conocimiento usado por los individuos para percibir su

mundo. En este trabajo nos enfocamos en la parte técnica del conocimiento tácito, la cual puede ser considerada como el “saber cómo”. Las habilidades técnicas son todas aquellas que hacen experta a una persona en un dominio específico dentro de la organización. Aun cuando una parte de dicho conocimiento técnico puede ser articulada y convertida en conocimiento explícito con relativa facilidad (por ejemplo los procedimientos, técnicas, etc.), hay otra parte que depende del individuo, y por lo tanto es difícil, y en algunas ocasiones imposible, formalizarlo para hacerlo explícito. Por ejemplo, una persona podría saber qué procedimientos usar para realizar una actividad, pero no es lo mismo que tener la experiencia que le permite a un experto saber cuando adaptar tales conocimientos a situaciones excepcionales que no había tenido que encarar previamente. Basados en estas observaciones hemos dividido la parte tácita del modelo en dos tipos: “saber cómo”, y “habilidades” respectivamente.

Por otra parte, el conocimiento explícito es aquel que ya ha sido formalizado y almacenado en algún medio formal. Esta formalización puede tener diversos niveles de estructuración, desde altamente no estructurado, el texto libre por ejemplo, hasta altamente estructurado, como información representada en una forma rigurosamente matemática. En la Tabla X hemos hecho una distinción entre dos niveles de conocimiento explícito: no estructurado y estructurado. Para definir de mejor manera el grado al cual éste está estructurado, se han definido una serie de posibles formatos de representación, donde el nivel de estructuración crece desde la parte de arriba de la tabla hacia abajo. Es preciso añadir que los formatos presentados son sólo a manera de ejemplos, por lo que no pretendemos dar una lista exhaustiva, esta puede ser modificada según las necesidades de cada caso.

Tabla X. Esquema de clasificación de estructuras de conocimiento.

Estructura	Formato	Descripción
Tácito (T)	Habilidad	Conocimiento técnico dependiente de la experiencia de las personas que resulta difícil, si no imposible, de formalizar y hacer explícito.
	Saber cómo	Conocimiento técnico que puede ser relativamente fácil de hacer explícito, por ejemplo procedimientos, métodos, técnicas, etc. También este tipo de conocimiento puede estar contenido de forma implícita en algunas fuentes formales, como un flujo de trabajo, o un sistema basado en procesos que ayuda a las personas a saber qué tienen que hacer, por ejemplo qué actividades deben realizarse para alcanzar una meta.
Explicito no estructurado (ENE)	Audio o video (A V)	Medios de almacenamiento que pueden ser usados para guardar y almacenar conocimiento, pero debido a su naturaleza, es difícil procesarlo automáticamente, por lo que generalmente se requiere de las personas para interpretarlo.
	Imagen (I)	Pueden ser usadas para representar información visualmente. Aunque hay técnicas para extraer información de imágenes en forma automática (algoritmos de procesamiento de imágenes), comúnmente se requiere de alguna persona para poder interpretarlas completamente.
	Texto libre (TL)	Esta categoría se refiere a fuentes de conocimiento expresadas en forma de texto libre como documentos no estructurados, por ejemplo reportes, memos, etc. Existen herramientas que permiten manejar este tipo de fuentes usando las palabras que contienen, por ejemplo para indexarlas o clasificarlas.
Explicito estructurado (EE)	Grafica (GR)	Esta categoría es usada para especificar conocimiento representado gráficamente y que sigue un lenguaje gráfico bien definido, de forma tal que cualquiera que conozca el lenguaje pueda entender la información representada, e incluso, un sistema que interprete dicho lenguaje podría extraer información de forma automática.
	Texto Semi-estructurado (TSE)	Esta representación permite identificar cierta información de un documento en forma automatizada, tales como títulos, tipos de párrafos, etc., debido al uso de una representación semi-estructurada de la información. Un ejemplo es el uso de HTML [W3C-HTML], el cual facilita obtener información sobre la estructura de un documento.
	Texto estructurado (TE)	Esta representación almacena información más detallada sobre un documento, lo cual facilita su manejo con computadoras. Un ejemplo es XML [W3C-XML], que permite obtener información sobre el contenido y estructura de un documento.
	Datos (D)	Los datos por si solos no tienen significado, pero agrupados y puestos en contexto pueden proveer información y conocimiento. Más aun, existen técnicas que pueden ser usadas para descubrir conocimientos en grandes conjuntos de datos, por ejemplo la minería de datos.

Tabla X. Esquema de clasificación de estructuras de conocimiento (continuación).

Estructura	Formato	Descripción
Explicito estructurado (EE)	Metadatos (MD)	Los metadatos permiten obtener información para facilitar el manejo de datos o información por medios computarizados. Los metadatos pueden ser vistos como información a cerca de los datos u otra información. Un ejemplo son las ontologías usadas para describir el contexto de algunos datos o dominios, por ejemplo, los conceptos, sus atributos y relaciones [Ding y Foo, 2002].
	Conocimiento o información categorizado (CC)	Esta categoría puede ser usada para sistemas que hacen uso de estructuras bien definidas para clasificar documentos u otras fuentes de información. Ejemplos de esto son los directorios usados en algunos buscadores de páginas Web como Yahoo [Yahoo] o Google [Google] para buscar páginas por categorías.
	Formulación matemática (FM)	Esta categoría considera el conocimiento expresado de una forma matemática, con una definición de su semántica explícita y rigurosa, en forma tal que haga posible seguir reglas para manejar ese conocimiento sin ambigüedad. Un ejemplo puede ser un sistema experto con una base de conocimientos creada siguiendo una formulación matemática, haciendo fácil obtener respuestas concretas a preguntas específicas expresadas en un lenguaje formalmente definido.

Finalmente, es importante considerar el hecho de que algunas herramientas pueden usar más de uno de los formatos descritos en la Tabla X. Por ejemplo, existen sistemas para el manejo de contenidos basados en XML [W3C-XML], que manejan documentos por medio del almacenamiento de cierta información (metadatos) sobre los mismos, y que usan esta representación para clasificarlos dentro un repositorio. Así mismo, estos sistemas también pueden proveer motores de búsquedas que indexan los documentos usando tanto los metadatos, como el texto (texto libre) contenido en ellos [Grehan, 2005].

V.4 Soporte al flujo del conocimiento

En esta etapa, se analiza la herramienta para identificar en cuál, o en cuáles de las etapas que constituyen el flujo del conocimiento influye; ya sea de forma positiva o negativa. Para este fin se toman en consideración las etapas propuestas en el modelo presentado en el Capítulo II, Sección II.1.2.3 de este documento. Como se describió en

dicho capítulo, el modelo sugiere un conjunto de actividades de AC que son las que constituyen el flujo del conocimiento desde donde es creado, hacia donde es almacenado y/o utilizado, pasando por las distintas etapas requeridas para su transmisión de un lugar a otro. Las etapas principales que han sido consideradas en dicho modelo son: 1) creación o adquisición, 2) externalización, 3) almacenamiento y mantenimiento, 4) transmisión (que incluye la transmisión de conocimiento tácito por medio de la socialización, y la de explícito por medio de su formalización, almacenamiento y posterior recuperación), 5) internalización (compuesta de la recuperación y la interpretación y filtrado), y finalmente 6) aplicación y evolución. Una descripción de cada etapa puede ser consultada en la Sección II.1.2.3 del Capítulo II de este documento.

Para la definición del grado al cual un determinado sistema soporta las distintas actividades antes mencionadas, se ha definido un esquema de medición compuesto por un rango de 5 posibles valores, mostrado en la Tabla XI. Como se observa, el esquema es muy general, por lo que no define detalles sobre el tipo de soporte. Así una vez determinado el nivel, es preciso detallar explícitamente el tipo de apoyo que da el sistema a las etapas definidas. La ventaja en el esquema propuesto, es que es posible definir cuando un determinado sistema no sólo no soporta alguna etapa, sino que la obstaculiza. Esto facilitaría el centrarse en identificar las causas del problema con el fin de buscar evitar que dicho sistema interfiera de manera negativa en el flujo del conocimiento.

Tabla XI. Conjunto de valores para definir el grado en que una herramienta interfiere con las etapas del flujo del conocimiento.

Valor	Definición
++	Lo mejora
+	Lo permite
+-	Lo soporta parcialmente
-	No lo soporta
--	Lo obstaculiza

V.5 Aspectos Técnicos

Con frecuencia los SACs no son usados adecuadamente en las organizaciones debido a cuestiones técnicas [Stewart, 2002], por ejemplo si requieren demasiado tiempo para buscar o capturar conocimiento [Desouza, 2003; Rus et al., 2001]. Por lo tanto, es importante identificar estos aspectos con el objetivo de analizar si un SAC específico reduce dichos problemas. Los aspectos técnicos que podrían ser considerados son muchos y muy variados, y podrían depender del tipo de sistema y de las necesidades particulares de cada organización. Sin embargo, hemos identificado dos aspectos que han sido definidos entre las principales barreras para la adopción de SACs en diversos ambientes. Estos dos aspectos son: 1) reducir el trabajo requerido por los usuarios al proveer apoyo automatizado para algunas de las actividades del ciclo de vida de la AC [Marwick, 2001; van Elst et al., 2003], y 2) facilitar el manejo del conocimiento que se encuentra distribuido a lo largo de una organización [Banerjee, 2005; Bonifacio et al., 2002b; Cuel, 2003; van Elst et al., 2003].

En ocasiones, en las organizaciones, parte del conocimiento que es importante administrar se encuentra distribuido, por lo que se dificulta su administración, e incluso en algunos casos no es considerado siquiera como parte de la base de conocimientos de la organización debido a que no se encuentra en los repositorios de conocimiento centralizados, por ejemplo, documentos almacenados en los espacios de trabajo de los empleados, lo que hace difícil que sus colegas tengan acceso a dichos documentos. Los sistemas de AC tradicionales no están diseñados para tomar en cuenta la natural distribución del conocimiento dentro de las organizaciones, como ha sido observado por Bonifacio et al. [2002b], y por van Elst et al. [2003]. Por otro lado, los sistemas de AC tradicionales requieren demasiado trabajo por parte de los usuarios, quizá porque es necesario que los usuarios capturen su conocimiento en el sistema, o debido a que es difícil para ellos buscar conocimientos específicos. Por ejemplo, en el área del desarrollo de software este último parece ser un problema común [Desouza, 2003; Lethbridge *et al.*, 2003; Lindvall y Rus, 2003; Seaman, 2002]. Con base en lo anterior, hemos considerado que tomar en cuenta la manera en que una determinada herramienta disminuye dichos

problemas podría ser importante, sobre todo en el área del desarrollo de software donde dichos aspectos tienden a dificultar el uso de sistemas de AC. Para indicar el grado al cual una determinada herramienta considera estos dos aspectos técnicos, se han definido dos características: el nivel de autonomía provisto por una herramienta, y el tipo de distribución de conocimiento que la herramienta maneja. Los posibles valores para estas características son resumidos en la Tabla XII, y detallados a continuación.

Tabla XII. Parámetros usados para medir el grado en que una herramienta de AC soporta la autonomía y la distribución del conocimiento.

Parámetro	Descripción	Valores
Autonomía	Indica qué tan autónoma es una herramienta al soportar actividades específicas del ciclo de vida de la AC. Si requiere que las personas hagan todo, o si puede actuar de forma automática bajo ciertas circunstancias.	Un valor del conjunto: [no-autónomo, semi-autónomo, autonomía-compartida, autonomía-delegada, autonomía-total]
Distribución	Indica la distribución del conocimiento que es manejado por la herramienta.	Un valor del conjunto: [local, centralizado internamente, centralizado externamente, distribuido internamente, distribuido externamente]

V.5.1 Nivel de autonomía

Un sistema autónomo es aquel capaz de sensar el medio ambiente en el que se encuentra embebido, y actuar en él en busca de sus propias metas, tales sistemas son considerados como agentes autónomos [Franklin y Graesser, 1996]. Esta definición de autonomía es muy general, por lo que posibilita que estos tipos de sistemas tengan diversos tipos y grados de autonomía [Beavers y Hexmoor, 2003; Brainov y Hexmoor, 2001; Carabelea *et al.*, 2003; Hexmoor, 2000]. En el caso de MAHAC, el interés se centra en la autonomía de usuario, lo cual significa “que un sistema es autónomo si puede decidir que acciones realizar sin la intervención del usuario” [Carabelea *et al.*, 2003]. Sin embargo, aun

cuando algunas actividades pueden ser realizadas sin intervención de los usuarios, otras podrían requerir las acciones del mismo, por lo que este tipo de autonomía puede ir de una dependencia completa hasta una total independencia del usuario [Carabelea et al., 2003]. Hexmoor [2000] ha propuesto cinco niveles de autonomía que pueden ser usados para determinar el grado de autonomía de usuario:

- 1) **No-autónomo.** Implica que el sistema no tiene ningún tipo de autonomía.
- 2) **Semi-autónomo.** Significa que la autonomía del sistema depende de restricciones externas, tales como la intervención del usuario para que el sistema actúe. Por ejemplo, cuando un usuario debe indicar al sistema exactamente qué hacer y cuándo hacerlo, o proveerle de toda la información necesaria para realizar una actividad específica.
- 3) **Autonomía-compartida.** Indica que existe una coparticipación del sistema y el usuario para la obtención de un objetivo. Por ejemplo, el sistema puede apoyar al usuario al sugerir información a considerar o acciones a ser realizadas, pero el usuario decide si usar o no la información, o realizar o no dichas acciones.
- 4) **Autonomía-delegada.** Significa que el sistema dirige el cumplimiento de un objetivo, pero que puede delegar ciertas decisiones al usuario.
- 5) **Autonomía-total.** Este tipo de autonomía fue denominada como autonomía-propia por Hexmoor [2000]. Sin embargo, le hemos denominado autonomía-total para diferenciar el concepto de la autonomía-propia usado por Carabelea et al. [2003]. Este tipo de autonomía indica que el sistema tiene control de si mismo, y que realiza todas o la mayoría de las acciones requeridas para completar un objetivo.

Con el fin de ilustrar la forma de utilizar los niveles definidos, se presenta a continuación un caso ejemplo, el cual está dividido en varios escenarios para un sistema de búsquedas enfocado en ayudar a obtener información o conocimiento útil para completar alguna tarea específica:

- 1) Un sistema de búsquedas *no-autónomo* sería aquel que requiere que el usuario decida el momento, el lugar, y el conocimiento a ser buscado, por ejemplo al introducir palabras claves o consultas de búsqueda relacionadas con el información que desea encontrar.

- 2) Un sistema *semi-autónomo* podría ser uno capaz de sugerir palabras clave o consultas de búsqueda que puedan estar relacionadas con las introducidas por el usuario, de manera que el sistema ayude a este último a redefinir los parámetros de búsqueda con el fin de encontrar información más adecuada.
- 3) Supongamos ahora que el usuario solicita al sistema empezar la búsqueda de información relacionada con las actividades que está realizando, y que el sistema es capaz de conocer el contexto del usuario (por ejemplo la tarea que está realizando) para formular las palabras claves o consultas de búsqueda con el fin de iniciar la búsqueda sin que el usuario proporcione dicha información; en este caso, el sistema estaría trabajando en un esquema de *autonomía-compartida*.
- 4) Ahora supongamos que una vez que la búsqueda ha iniciado, el sistema identifica que existen distintos tipos de fuentes (tipos de documentos, personas, reportes de lecciones aprendidas, etc.) o que éstas proceden de distintos lugares, por lo que decide preguntar al usuario qué tipo de fuentes son más importantes; en este caso podría considerar que el sistema presenta una *autonomía-delegada*, ya que delega ciertas decisiones al usuario para mejorar los resultados de la búsqueda.
- 5) Finalmente, si el sistema es capaz de identificar la necesidad de conocimiento del usuario y empezar la búsqueda antes de que se le sea solicitado, además es capaz de manejar el historial y perfil del usuario con el fin de inferir qué fuentes puedan ser más relevantes para él y ordena las fuentes de forma automática en base a su rango de relevancia para presentarle al usuario sólo aquellas que puedan ser más relevantes, realizando de esta manera una búsqueda sin la intervención directa del usuario, entonces dicho sistema podría estar presentando una *autonomía-total*.

V.5.2 Distribución del conocimiento

La administración del conocimiento distribuida es un área de investigación que ha emergido recientemente para abordar el problema del manejo del proceso de creación e intercambio de conocimiento que es local a grupos autónomos [Bonifacio et al., 2002b]. El conocimiento en las organizaciones se encuentra distribuido por naturaleza; reside en diversas fuentes que pueden estar distribuidas a lo largo y ancho de toda la organización.

Algunas de estas fuentes pueden estar incluso localizadas fuera de la organización, o dispersas geográficamente a lo largo del mundo; un ejemplo son las compañías transnacionales [Gilbert, 2007]. Los sistemas de AC tradicionales tienden a usar repositorios centralizados de todo tipo de fuentes de conocimiento [Kühn y Abecker, 1998] para manejar y controlar el conocimiento de forma centralizada [Bonifacio et al., 2002b]. Sin embargo, no todo el conocimiento debe ser manejado en esta forma, y no considerar la naturaleza distribuida del conocimiento puede ser causa de que los usuarios consideren que un sistema no es del todo útil [Bonifacio et al., 2002b; van Elst et al., 2003]. La utilidad del conocimiento, por otro lado, en ocasiones puede depender del lugar de donde fue obtenido o creado o donde será aplicado [Bonifacio et al., 2002b; Davenport y Prusak, 2000: p. 41]. Por lo tanto, en algunos casos resulta importante conocer si un sistema de AC permite el manejo de conocimiento de forma centralizada o distribuida.

El significado de conocimiento distribuido depende de la definición que de conocimiento local se tenga. Puede ser conocimiento que pertenece a una sola persona, a un grupo o unidad de una empresa, a una organización completa, y así sucesivamente. De aquí que para definir si un SAC permite o no el manejo de conocimiento distribuido, primero debemos definir los límites de los *nodos de conocimiento* inmersos en el ambiente en el cual el sistema será usado [Cuel, 2003]. Un nodo de conocimiento puede definirse como “la representación de un dueño de conocimiento dentro de la organización, una entidad nombrada (individual o colectiva) que tiene la capacidad de manejar su propio conocimiento tanto desde el punto de vista conceptual como técnico” [Bonifacio *et al.*, 2002a]. Una vez definidos los límites de los nodos de conocimiento de la organización, será posible, entonces, identificar si el conocimiento es manejado de forma centralizada o distribuida desde el punto de vista de los nodos de conocimiento. Para hacer esto, el siguiente conjunto de alternativas de distribución pueden ser usadas:

- 1) **Local.** El conocimiento de cada nodo es creado, almacenado, administrado o usado de forma local a dicho nodo.
- 2) **Centralizado-internamente.** El conocimiento es creado, almacenado, administrado o usado en una localidad centralizada dentro de la organización de nodos de

conocimiento.

- 3) **Centralizado-externamente.** El conocimiento es creado, almacenado, administrado o usado en una localidad centralizada fuera de la organización de nodos de conocimiento.
- 4) **Distribuido-internamente.** El conocimiento es creado, almacenado, administrado o usado de forma distribuida a través de la organización de nodos de conocimiento.
- 5) **Distribuido-externamente.** El conocimiento es creado, almacenado, administrado o usado de una forma distribuida a través de nodos de conocimiento dentro y fuera de la organización.

Para ejemplificar el uso de las distintas posibilidades de distribución del conocimiento presentaremos un caso ejemplo. El ejemplo trata de distintas posibilidades para la distribución del conocimiento requerido por un grupo de ingenieros de software de una organización de desarrollo de software. La empresa tiene diferentes unidades a lo largo del mundo, y cada unidad tiene su propio departamento de desarrollo de software, cada departamento de la organización es considerado como un nodo de conocimiento, y queremos proveer apoyo a la AC a un departamento de desarrollo de software de una unidad en particular, donde el conjunto de nodos de conocimiento de una unidad es considerado como una organización de nodos de conocimiento. De lo anterior se derivan los siguientes escenarios:

- 1) Si el conocimiento usado por el departamento de desarrollo de software es almacenado en un repositorio de conocimiento controlado por los miembros del mismo departamento, entonces el conocimiento es *local*.
- 2) Si el conocimiento es manejado por un servidor fuera del departamento de desarrollo, por ejemplo por un servidor central de la unidad a la que pertenece el departamento, entonces el conocimiento es manejado de forma *centralizada-internamente*.
- 3) Si el conocimiento es manejado por un servidor central de la compañía que se encuentra en una unidad distinta a la cual pertenece el departamento de desarrollo en cuestión, entonces es manejado de forma *centralizada-externamente*.
- 4) Si el conocimiento es manejado por distintos departamentos de la misma unidad, entonces es manejado de forma *distribuida-internamente*.

5) Finalmente, si el conocimiento es manejado por distintos departamentos de diferentes unidades (por ejemplo, si cada departamento de desarrollo tiene su propio repositorio de lecciones aprendidas, pero las comparten con el resto de los departamentos de desarrollo de software), entonces es manejado de forma *distribuida-externamente*.

Es importante remarcar que los dos factores técnicos definidos no son los únicos que podrían ser usados. Podrían definirse otros aspectos técnicos acordes a cada caso en particular dependiendo de las necesidades que se tengan, como del tipo de herramientas a analizar. Por ejemplo, Tiwana y Ramesh [2001] han considerado aspectos como robustez, seguridad, o escalabilidad, entre otros, para analizar sistemas de AC basados en el Web. Otro ejemplo es el uso de una dimensión espacio-temporal como la usada por Banerjee [2005] y Rao [2005a], en la cual el lugar (colocalizado/remoto) en el que se encuentran quienes aportan o toman el conocimiento, y el tiempo de acceso al mismo (síncrono/asíncrono) son considerados. Finalmente, también es importante decir que el marco de trabajo en su conjunto puede ser usado según la conveniencia de quien lo aplique. Sin embargo, para facilitar un poco este trabajo, se propone una matriz de definición de características que puede ser usada como guía para la aplicación de MAHAC. Esto se ejemplifica en la siguiente sección.

V.6 Una matriz para la aplicación de MAHAC

Con el fin de facilitar la aplicación de MAHAC, se ha definido una matriz para la definición de las características que cubre una determinada herramienta. Esta matriz es presentada en la Tabla XIII. Como se puede observar, la matriz se divide, de izquierda a derecha, en tres partes, primeramente se listan las distintas características definidas en MAHAC, las cuales se encuentran agrupadas según la etapa a la que pertenecen. En seguida de cada característica viene un espacio para definir el nivel o valor correspondiente, el cual indica el grado al cual la herramienta cubre dicha característica. Este nivel o valor corresponde al que resulte más adecuado de entre las distintas posibilidades definidas para cada característica en el marco de trabajo, y que ya han sido presentadas en este capítulo. Finalmente, se incluye un campo de observaciones, para indicar mayores detalles sobre la

forma en que la herramienta soporta alguna característica determinada. El uso de la matriz aquí presentada será ejemplificado con un caso concreto en el siguiente capítulo, donde se muestra el uso de la metodología en lo general, y del marco de trabajo MAHAC en lo particular.

Tabla XIII. Matriz de definición de características consideradas en MAHAC.

Característica	Nivel / Valor	Observaciones
Dominio del conocimiento		
Uso		
Alcance		
Dominio		
Estructura del conocimiento		
Tácito		
Explícito		
Soporte al flujo de conocimiento		
Creación		
Externalización		
Almacenamiento		
Transferencia		
Tácito		
Explícito		
Internalización		
Recuperación		
Interpretación/Filtrado		
Aplicación		
Aspectos técnicos		
Autonomía		
Distribución		

V.7 Resumen del capítulo

En este capítulo se presentó un marco de trabajo para el análisis de herramientas de soporte al flujo del conocimiento. La aplicación de dicho marco es el paso final en la metodología KoFI propuesta en esta tesis. Si bien la aplicación de MAHAC podría ser independiente a la metodología KoFI. Su uso como parte de la metodología facilita la

identificación de la información requerida para definir el nivel al cual una determinada herramienta soporta las distintas características definidas. Esto es debido a que las etapas previas en la metodología proporcionan parte de esta información, además de que los modelos desarrollados en la primera etapa de la metodología y durante la realización de la fase de análisis pueden servir de base para el análisis de las herramientas que intervienen en el proceso analizado, identificando, por ejemplo, en qué tipo de actividades son útiles, qué personas las requieren, qué tipo de conocimiento permiten obtener o almacenar, etc.

El uso de MAHAC puede ayudar tanto en el análisis de herramientas diseñadas desde el punto de vista de la AC, como aquellas que habiendo sido desarrolladas con otros propósitos, de una u otra manera interfieren en el flujo del conocimiento dentro de un proceso. Esto último es importante debido a que permite lograr uno de los objetivos para los cuales KoFI fue definida, y que es integrar la infraestructura tecnológica de una organización dentro de sus estrategias de AC. Lo anterior deberá facilitarse al identificar el papel que dicha infraestructura juega dentro del flujo del conocimiento, por ejemplo, que actividades de la AC apoya y cuales obstaculiza. La obtención de dicha información podrá ser posteriormente utilizada para definir alternativas encaminadas a mejorar el uso de dichas herramientas como habilitadoras del flujo del conocimiento, modificarlas, o en su defecto intercambiarlas por otras que proporcionen un servicio similar, pero que apoyen de mejor manera el flujo del conocimiento durante el proceso.

Con el objetivo de ejemplificar y de mostrar la manera en que la aplicación de la metodología efectivamente puede ayudar a lograr lo anteriormente expuesto, en el siguiente capítulo se presentan los resultados de un caso de estudio realizado en un grupo de mantenimiento de software. Este caso de estudio es también utilizado para ejemplificar la forma en que pueden aplicarse tanto el marco de trabajo propuesto en este capítulo, como las otras técnicas que componen KoFI y que han sido presentadas en los dos capítulos anteriores.

CAPÍTULO VI

Resultados: utilización de la metodología

“Para que la investigación académica sea relevante, los investigadores deberían probar sus teorías con practicantes en situaciones y organizaciones reales” [Avison et al., 1999]

Como se observó a lo largo de este documento, esta tesis propone un conjunto de técnicas y métodos para apoyar en el análisis de procesos organizacionales, y en especial de software, desde un enfoque centrado en el flujo del conocimiento que se da en dichos procesos, de ahí que se haya definido como una metodología. Es importante tener en cuenta que esta metodología, para ser usada requiere una intervención humana directa muy fuerte. Este hecho dificulta una evaluación formal (por ejemplo un experimento) [Checkland y Scholes, 1999: p. A12]. Una alternativa podría ser compararla con varios enfoques similares. Sin embargo, hasta donde conocemos no hay un enfoque similar al que proponemos, además de que el resultado de aplicar uno u otro enfoque podría depender en gran medida de factores externos difíciles de medir y controlar.

Los enfoques que hemos encontrado en la literatura persiguen distintos objetivos y son lo bastante diferentes al propuesto en esta tesis como para hacer una comparación que realmente resulte útil como mecanismo de evaluación de la metodología en su conjunto. Por lo anterior, sólo se realizaron comparaciones con las técnicas individuales propuestas como apoyo a la metodología, como lo es el lenguaje de modelado, o el enfoque para el análisis de herramientas de AC. Estas comparaciones se presentan en el Capítulo IV para el caso de la propuesta de modelado, y en el Capítulo V para el marco de trabajo para el análisis de sistemas de apoyo a la AC.

Por otro lado, hay que tener en cuenta que validar cualquier enfoque de AC en general es algo sumamente complicado, como apuntan Anquetil et al. [2007: p. 523],

debido a que los resultados sólo se observan en el largo plazo, y lo que es más importante, están influenciados por un conjunto de variables externas que hacen difícil identificar los factores concretos que puedan mostrar claramente los beneficios [Folkens y Spiliopoulou, 2004; Kankanhalli y Tan, 2004].

Kitchenham y Pickard [1995] sostienen que dada la existencia de factores difíciles de medir o controlar, lo más adecuado es el uso de casos de estudio para la evaluación de métodos en ingeniería de software. Siguiendo esta observación, y debido a todo lo expuesto anteriormente, para la validación de la metodología como tal, se decidió evaluar los resultados de su uso por medio de un caso de estudio. Específicamente la validación se ha centrado en determinar si la metodología efectivamente tiene el potencial de cumplir con el objetivo para el que fue diseñada, es decir, ayudar a obtener información para proponer el diseño de sistemas o estrategias de AC que consideren el trabajo real desempeñado por las personas que participan en un proceso, tomando en cuenta las herramientas que dichas personas utilizan para realizar sus actividades.

En el caso de estudio se aplicó la metodología para analizar un proceso de mantenimiento de software con dos objetivos: 1) ejemplificar el uso de la metodología en un ambiente real, y 2) observar los posibles beneficios de su uso. Para complementar los resultados de este caso de estudio, se obtuvieron datos de un segundo caso de estudio en el que la metodología fue aplicada en un contexto distinto y por una persona ajena a este trabajo, lo que da un interesante punto de vista de una persona no influenciada por el hecho de haber participado en la definición de la metodología.

Adicionalmente, antes de la realización del caso de estudio se decidió validar si el lenguaje de modelado de procesos propuesto como apoyo a la fase de análisis de la metodología era adecuado para el contexto en el que se usaría; es decir los procesos de software. La decisión que nos llevó a la elección del lenguaje de modelado fue el supuesto de que por haber sido diseñado para procesos de software, podría ser adecuado en el contexto en el que se usaría la metodología. Por lo tanto, para saber si la elección podría ser

correcta, se diseñó un experimento que nos proporcionó datos para ayudarnos a determinar que el lenguaje elegido es adecuado para dicho contexto.

En este capítulo se describen los principales resultados de ambos casos de estudio y del experimento. Detalles de cada uno de ellos se encuentran en los apéndices B (primer caso de estudio), C (segundo caso de estudio) y A (experimento).

VI.1 Caso de estudio

Esta sub-sección tiene dos objetivos principales: 1) mostrar la manera en que las distintas técnicas propuestas en este trabajo apoyan durante la utilización de la metodología, y 2) ejemplificar los beneficios o resultados potenciales que podrían ser obtenidos de su uso. Esta sección hace una descripción general de la forma en que se usaron las distintas técnicas para el análisis de un proceso de mantenimiento de software, así como de los principales resultados de dicho análisis. En particular, se hace énfasis en la forma en que el lenguaje de modelado de procesos propuesto apoya para la realización de la fase de análisis de la metodología, y la forma en que el marco de trabajo apoya en el análisis de herramientas que intervienen en el flujo del conocimiento. Finalmente se describen los resultados del caso de estudio, que incluyen las propuestas de mejora al proceso estudiado, y las principales lecciones aprendidas durante el mismo. Los detalles de la descripción del proceso estudiado se presentan en el Apéndice B.

VI.1.1 Contexto del caso de estudio

El caso de estudio se realizó en el Departamento de Informática (DI) del CICESE. En ese momento, el departamento se encontraba constituido por 14 personas: el jefe del departamento, una secretaria, seis ingenieros de software, y seis programadores asistentes (becarios). El estudio del proceso se hizo con apoyo de los Ingenieros de Software, o técnicos, como los llaman en el DI. Uno de los técnicos tenía sólo 2 meses de haber entrado al grupo, mientras que el resto tenía entre 5 y 11 años de trabajar dentro del DI. Todos los programadores asistentes tenían menos de un año trabajando en el departamento.

El DI no sigue un proceso de mantenimiento estándar, no obstante, se observó que muchas de sus actividades son similares a las definidas en procesos estándar, como por ejemplo [Chapin *et al.*, 2001; IEEE, 1995; ISO/IEC, 1999; Polo *et al.*, 2003a]. El trabajo en el DI está organizado en proyectos que nacen de: 1) solicitudes de cambio hechas por los usuarios de los distintos sistemas a los que el DI da soporte, ya sea para agregar nuevas funciones, o adaptarlos a cambios en el ambiente (por ejemplo en el proceso apoyado por el sistema); o 2) reportes de problemas, que describen problemas que los usuarios han experimentado con el sistema². Cada técnico tiene asignado uno o varios sistemas, y es responsable de las solicitudes de mantenimiento que se hagan sobre dichos sistemas, aún cuando eventualmente los cambios sean realizados por otro técnico o por algún programador asistente. Las solicitudes de mantenimiento son almacenadas en un sistema conocido como Sistema de Control de Proyectos (SCP). El SCP puede ser usado para rastrear el estado de cada solicitud. Los técnicos lo usan para rastrear y consultar información de las solicitudes que tienen asignadas. Debido a que el SCP resultó ser una parte central del proceso, decidimos enfocar el estudio hacia el análisis del proceso que se sigue en torno al uso del SCP como apoyo al proceso de mantenimiento del DI. A continuación se describe la forma en que se realizó el estudio por medio del seguimiento de las distintas fases que describe la metodología KoFI.

VI.1.2 Utilización de la metodología KoFI

La primera fase de KoFI consiste en realizar un modelo del proceso que posteriormente será empleado para analizar el conocimiento, fuentes y herramientas involucradas en los flujos de conocimiento dentro del proceso, para finalmente identificar el papel que juegan las herramientas que apoyan al proceso como facilitadoras del flujo del conocimiento. En el caso del DI, se detectó que el SCP es la principal herramienta de este

² En esta sección, se usará el término solicitud de mantenimiento para referirse tanto a las solicitudes de cambio como a los reportes de problema.

tipo, por lo que se decidió hacer el análisis del proceso centrándose en dicha herramienta, como se describe en las siguientes secciones.

VI.1.2.1 Modelado del proceso

La información para modelar el proceso en el caso de estudio se obtuvo en dos fases: primero se obtuvo una idea general del proceso por medio del análisis de un estudio previo [Rodríguez Elias, 2003]. Posteriormente, se hicieron dos entrevistas a uno de los técnicos del DI para que nos ayudara a definir el modelo general del proceso actual (en el momento del estudio), ya que algunas cosas habían cambiado desde el estudio previo y el nuevo (en particular, el SCP no existía en el estudio previo). Una vez que el técnico estuvo de acuerdo con el modelo general del proceso, se hicieron entrevistas a otros cuatro técnicos para que corroboraran los modelos y dieran sus observaciones y sugerencias. Cada entrevista duró alrededor de una hora. Durante este proceso, fue necesario hacer ajustes a los modelos conforme los técnicos los analizaban, debido a que cada técnico realizaba algunas actividades que eran distintas a las realizadas por el resto. Estas modificaciones consistieron en agregar elementos a los modelos, más que en rehacer los que ya estaban. Una vez que se tuvo un modelo completo del proceso, se procedió con la fase de análisis.

Un aspecto a destacar durante esta fase, es que los técnicos no tuvieron problemas para entender los modelos, aun cuando no se les dio una explicación previa de la nomenclatura empleada. Sólo dos de los entrevistados hicieron preguntas sobre el significado de algunos elementos del modelo. En particular, las preguntas fueron para corroborar si lo que entendían era lo que se pretendía representar, principalmente el conocimiento, al usar paquetes de conocimiento en vez de temas específicos. Esto refuerza lo encontrado durante el experimento (descrito en detalle en el Apéndice A) sobre la facilidad de los desarrolladores de software de entender los modelos generados con la nomenclatura elegida.

El proceso estudiado es el proceso de mantenimiento de software que realiza el departamento de informática del CICESE. Este proceso se basa en la atención de

solicitudes de cambios y resolución de problemas que el departamento recibe de usuarios en diversos puestos y áreas del CICESE, incluido el mismo personal del departamento de informática. Las solicitudes son almacenadas en un sistema de control de proyectos, en el cual los miembros del departamento llevan control de las solicitudes que tienen a su cargo. A continuación se describe la manera en que los distintos modelos generados ayudaron durante los distintos pasos de la fase de análisis de la metodología.

VI.1.2.2 Identificación y clasificación de fuentes y áreas de conocimiento

Los dos primeros pasos de la fase de análisis de la metodología consisten en la identificación y clasificación de las principales fuentes y tipos de conocimiento que intervienen en el proceso. Una de las principales contribuciones del lenguaje de modelado de procesos propuesto es que apoya en la realización de estos pasos. La Figura 34 muestra un diagrama de casos de uso en el cual se indican las principales áreas de conocimiento involucradas en el proceso. Este tipo de diagrama permite indicar los principales conocimientos que son generados o requeridos en el proceso. Estos diagramas pueden también mostrar flujos o dependencias de conocimiento entre las actividades o fases del proceso, por ejemplo conocimientos que son generados en una fase y requeridos en otras.

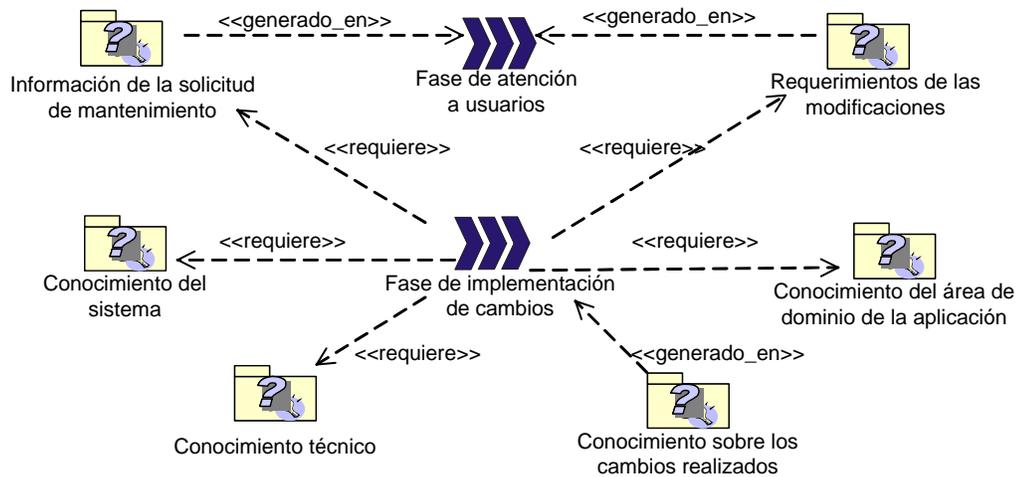


Figura 34. Diagrama de casos de uso que presenta las principales fases del proceso y principales paquetes de conocimiento requerido y generado en las mismas.

En el diagrama de la Figura 34 se puede observar, por ejemplo, que existe conocimiento generado en la fase de atención a usuarios que posteriormente es requerido en la fase de implementación de cambios. Estos conocimientos han sido agrupados en dos paquetes de conocimiento: “Información sobre la solicitud de mantenimiento”, y “Requerimientos de las modificaciones”. Una vez identificados los principales paquetes, áreas o temas de conocimiento, se pueden emplear diagramas de paquetes de conocimiento para comenzar a organizar dichas áreas o temas. En la Figura 35 se presenta un ejemplo.

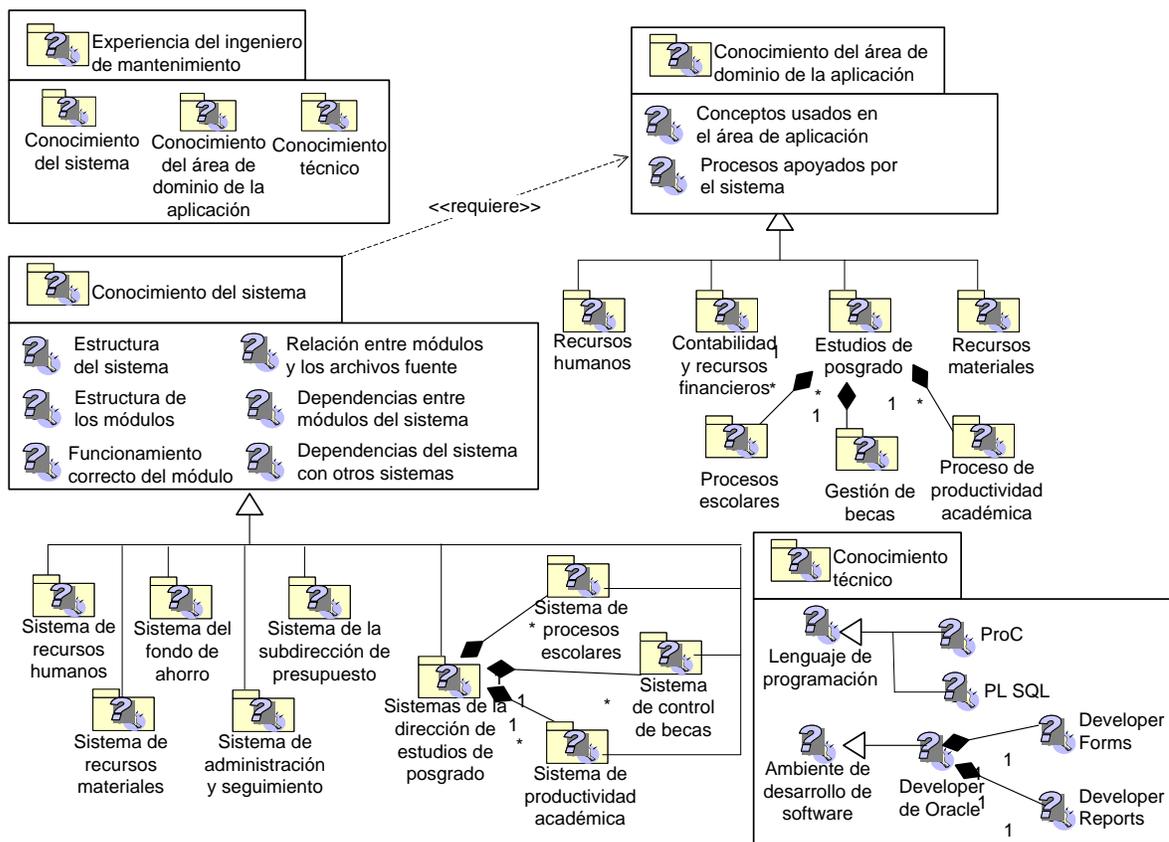


Figura 35. Diagrama de paquetes que clasifica áreas de conocimiento. Se muestran las relaciones de los paquetes de conocimiento (generales y específicos) requeridos en la fase de implementación de cambios.

El diagrama de la Figura 35 presenta una vista de las áreas y temas de conocimiento genéricos identificados en el caso de estudio. Durante el caso de estudio observamos que

era posible definir conjuntos de temas de conocimiento genéricos que posteriormente podrían ser especializados. Por ejemplo, se detectó que independientemente del sistema al que se diera mantenimiento, había un conjunto de temas comunes, cuya diferencia dependía del sistema al que se diera mantenimiento. Debido a esto, se decidió generar paquetes de temas generales, y posteriormente indicar los temas específicos referentes a cada tipo de sistema al especializar los paquetes generales. De esta manera pudimos definir de manera general los temas, e indicar que el tema concreto depende del sistema o aplicación de que se trate.

El diagrama de la Figura 35 muestra las principales áreas de conocimiento requeridas durante la fase de implementación de cambios. Si se comparan los diagramas de la Figura 34 y Figura 35, se puede observar que en el primero se identifican los paquetes generales que intervienen en cada fase, por ejemplo aquellas áreas de conocimiento que son requeridas. Por su parte, en el segundo se detallan estos paquetes y se identifican sus relaciones principales. Así mismo, los diagramas de paquetes de conocimiento pueden también ser usados para comenzar a definir dependencias entre temas o áreas de conocimiento, como se muestra en la Figura 35 entre los paquetes de “Conocimiento del sistema” y “Conocimiento del área de dominio de la aplicación”. De esta forma, podemos inferir, por ejemplo, que si estamos trabajando con el Sistema de la Dirección de Estudios de Posgrado (SIDEPE), entonces requeriremos, además del conocimiento sobre dicho sistema, conocimientos sobre los procesos del área de Estudios de Posgrado, que incluye los procesos de servicios escolares, gestión de becas y productividad académica; cada uno de los cuales tiene su propio sistema de apoyo, como también se muestra en la Figura 35³.

Otro uso de los diagramas de paquetes de conocimiento es que pueden ser usados dentro de los diagramas de clases para permitir la identificación de las fuentes de conocimiento donde pueden ser obtenidos los temas incluidos en dichos paquetes. Por

³ En esta sección usaremos el área de Estudios de Posgrado como caso particular para ejemplificar los distintos usos de los diagramas.

ejemplo, la Figura 36 muestra un diagrama en donde se han usado los paquetes de conocimiento que definen áreas de conocimiento generales, para definir el tipo de fuentes que pueden utilizarse para obtener información o conocimiento de dichas áreas. En el diagrama se indican los distintos tipos de actores o roles dentro del proceso que podrían tener conocimiento sobre determinadas áreas, los tipos de documentos o cualquier otro tipo de fuente.

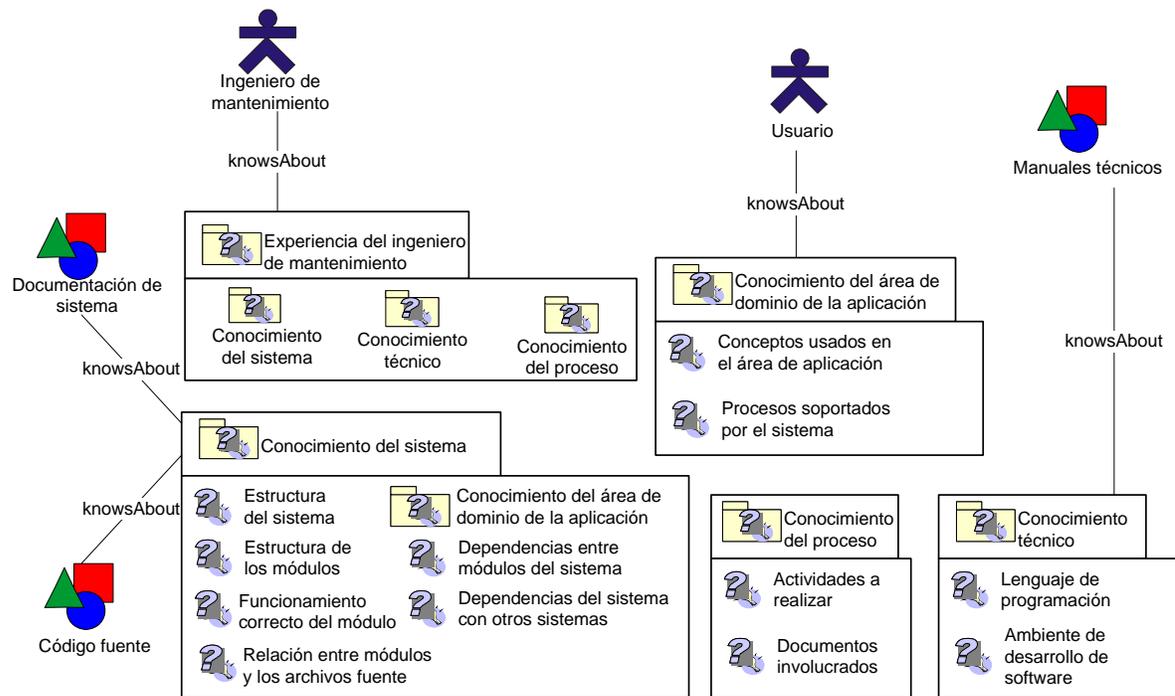


Figura 36. Diagrama de clases que identifica las relaciones entre fuentes y áreas de conocimiento.

De esta manera, una vez identificados los paquetes de conocimiento requeridos en las actividades, así como sus detalles y relaciones, podemos comenzar a identificar y hacer explícitas las fuentes donde pueden ser obtenidos los conocimientos agrupados en los paquetes. Como es el caso del diagrama de la Figura 36 que indica, por ejemplo, que los usuarios son fuente importante de conocimiento sobre el área de dominio de la aplicación, o que la documentación del sistema y el código fuente son las principales fuentes de

conocimiento explícito sobre los sistema que son mantenidos por el DI. No obstante, se encontró que no existe documentación para todos los sistemas, y la que existe, en su mayor parte no está actualizada.

De ser necesario, este tipo de diagrama puede detallarse más para indicar las fuentes específicas que pueden ser usadas para obtener conocimiento sobre temas concretos, como se muestra en la Figura 37. Tomando en cuenta la nomenclatura clásica de UML [Jacobson *et al.*, 2000], este tipo de detalles podrían considerarse como una especie de diagrama de objetos, donde se están identificando instancias específicas de los distintos tipos de fuentes de conocimiento.

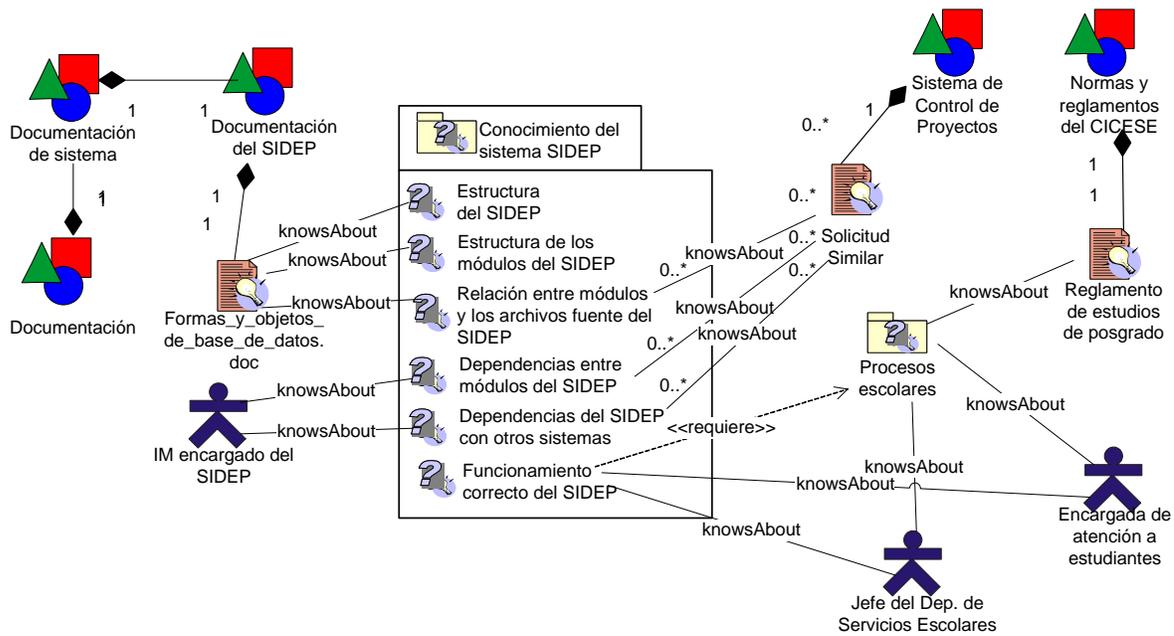


Figura 37. Diagrama de clases que especifica fuentes de conocimiento concreto. Se identifican las fuentes particulares donde pueden ser obtenidos los distintos temas agrupados en el paquete de conocimientos sobre el sistema SIDEP.

La Figura 37 presenta un diagrama que muestra las diferentes fuentes de conocimiento de donde pueden ser obtenidos los distintos temas que componen el paquete de conocimientos relacionados con el sistema SIDEP. Este diagrama tiene como objetivo

hacer énfasis en el conocimiento requerido al realizar cambios en el sistema SIDEPE, y las fuentes donde éste se encuentra distribuido. Al analizar este caso, se detectó que en ocasiones existen solicitudes de cambios anteriores que podrían ser similares. En algunos casos, los técnicos incluyen información en dichas solicitudes que podría ayudar a conocer los archivos fuentes que corresponden con un módulo específico, las dependencias con otros módulos del sistema, e incluso con otros sistemas. Sin embargo, esto no es un caso común. En ocasiones, la única fuente de dichos temas podría ser sólo el encargado de dar mantenimiento al sistema, por lo que si éste falta, una parte muy importante del conocimiento se podría perder. Este nivel de detalle puede ser útil para que quien requiera hacer modificaciones a un sistema en particular conozca el tipo de fuentes, e incluso las fuentes concretas que podría consultar para cada tema en particular, así como para identificar el conocimiento que podría no estar almacenado en ningún lado.

Tabla XIV. Ejemplo de descripción de una fuente de conocimiento. Presenta información sobre el documento “Formas_y_objetos_de_base_de_datos.doc”, el cual describe la estructura interna del sistema SIDEPE.

Nombre (id):	Formas_y_objetos_de_base_de_datos.doc		
Categoría:	Documentación		
Tipo:	Documentación de sistema/ Documentación del SIDEPE		
Descripción:	Este documento presenta tablas que definen las aplicaciones que son llamadas por cada uno de los elementos de los menús del SIDEPE, así como los objetos de bases de datos que están relacionados con dichas aplicaciones, tales como formas, tablas, reportes, etc.		
Localización:			
Tipo	Descripción	Soporte físico	Formato
Electrónico	Servidor de archivos/CIDESI/ documentos/		Word
Conoce a cerca de:			
Concepto		Nivel (Experto, avanzado, medio, etc)	
Estructura del SIDEPE		Avanzado	
Estrucutra de los módulos del SIDEPE		Avanzado	
Relación entre módulos y archivos fuente del SIDEPE		Avanzado	

Durante la fase de captura del proceso se encontraron documentos que el personal genera para ayudarse en sus tareas. Por ejemplo, en la Figura 37 se muestra el denominado “Formas_y_objetos_de_base_de_datos.doc”. Este documento fue generado por uno de los técnicos para ayudarse en sus tareas, sin embargo, se detectó que sólo el técnico que lo

generó conoce de su existencia. Con este tipo de observaciones, nos percatamos que los técnicos del DI han visto la necesidad de contar con ciertos conocimientos, y han tomado iniciativas propias encaminadas a lograr que dichos conocimientos no se pierdan. Desgraciadamente, estos documentos por lo general son de uso personal, y el resto del grupo no sabe de su existencia. Con el fin de resaltar la importancia de dichos documentos, se utilizó el formato de descripción de documentos (descrito en el Capítulo III, sección III.4.1.2) para especificar la información relacionada con los mismos, un ejemplo se presenta en la Tabla XIV. Gran parte de la información contenida en dichos formatos puede observarse en los modelos, lo que hace evidente la utilidad del enfoque de modelado propuesto para obtener la información que se requiere durante la fase de análisis de la metodología.

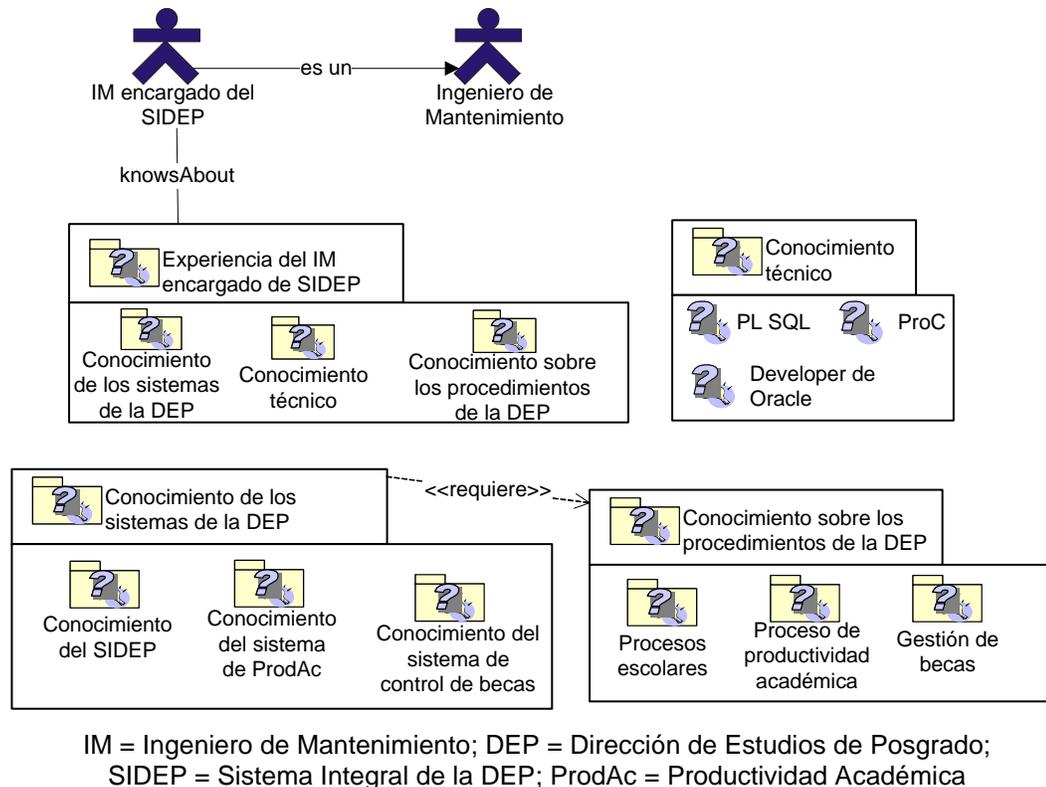


Figura 38. Diagrama de paquetes que identifica conocimiento requerido por los roles o actores en el proceso. Muestra las áreas de conocimiento que debe conocer el encargado de dar mantenimiento al sistema SIDEp.

Otra utilidad más, observada con respecto a la inclusión de los diagramas de paquetes como parte de los de clases, es que pueden ayudar a definir el conocimiento que requieren los roles del proceso, e incluso actores particulares, para realizar sus tareas. Un ejemplo de esto se presenta en la Figura 38 (anterior), donde se incluye un diagrama que indica el conocimiento que debe tener la persona encargada de dar mantenimiento al sistema SIDEPE.

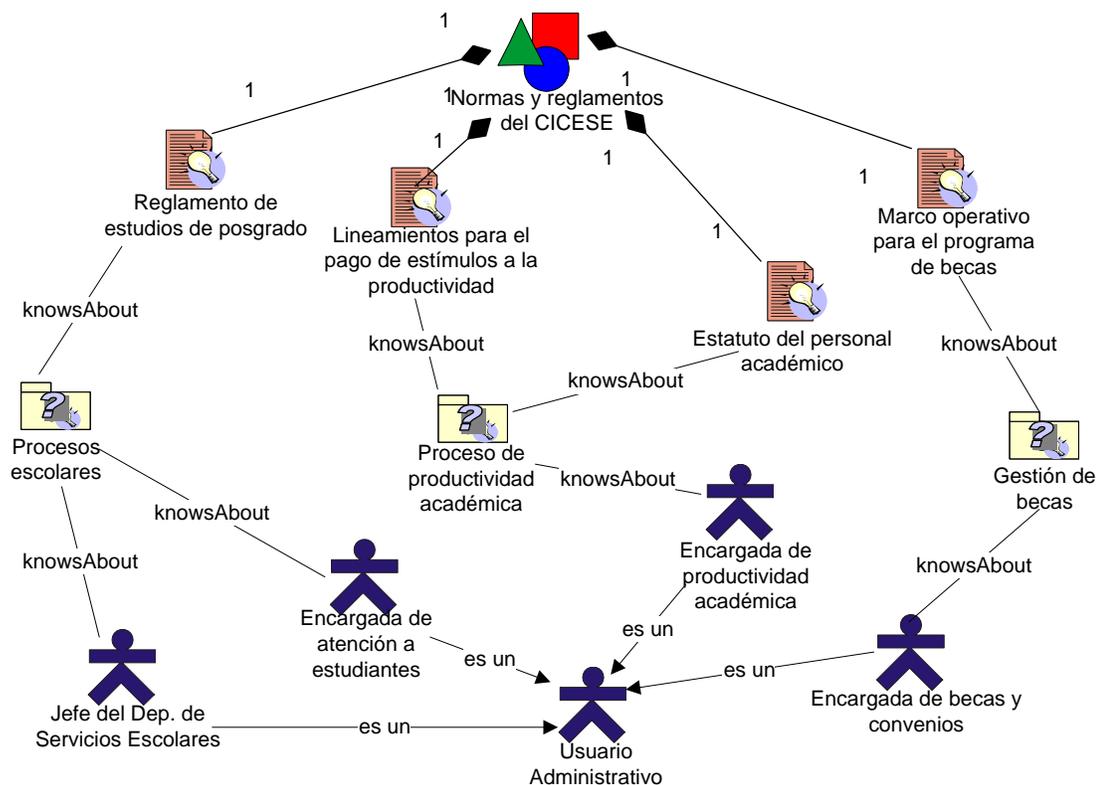


Figura 39. Diagrama de clases que identifica relaciones entre fuentes y áreas de conocimiento. Se muestran las fuentes de conocimiento relacionado con los procesos apoyados por el sistema SIDEPE.

Finalmente, el diagrama de la Figura 39 (arriba) presenta un ejemplo de otra de las utilidades principales del enfoque de modelado propuesto, la cual corresponde con la clasificación de las fuentes de conocimiento. Como se puede observar, el diagrama muestra dos clases de fuentes: personas y documentos, particularmente, en la primer clase se

encuentra un tipo específico que es “Usuario Administrativo”, al que pertenece el personal encargado de la DEP, que incluye al Jefe del Departamento de Servicios Escolares y encargados de atención a estudiantes, de productividad académica, y de becas y convenios. Estas últimas personas son las que conocen sobre los procedimientos de dicho departamento. Por otro lado, existen un conjunto de documentos que también tiene información sobre dichos procedimientos, los cuales han sido agrupados dentro del tipo definido como “Normas y reglamentos del CICESE”. Es decir, este diagrama clasifica un conjunto de tipos de fuentes de conocimiento relacionadas por el conocimiento que puede ser obtenido de ellas. Todas estas fuentes son útiles para quien requiera conocer sobre la forma en que debe funcionar el sistema SIDEPE, por ejemplo, los procesos que debe apoyar, información que debe proporcionar, etc.

Como se ha podido observar, la propuesta de modelado ayuda a representar gráficamente y de forma explícita gran parte de la información que se requiere obtener durante los dos primeros pasos de la fase de análisis de la metodología KoFI, los cuales son la identificación y clasificación de los tipos y fuentes de conocimiento. Esto permite que durante el modelado del proceso se esté trabajando ya en la obtención de dicha información, debido a que los modelos indican el conocimiento principal involucrado en el proceso, sus fuentes, e incluso la clasificación preliminar de los mismos. Por lo tanto, eventualmente la fase de análisis debería ser más rápida y sencilla en la medida en que los modelos del proceso presenten mayores detalles del conocimiento, sus fuentes, su clasificación, etc. Esto último es una de las razones por las que se observó la necesidad de un lenguaje de modelado como el propuesto.

Sin embargo, hay que tener en cuenta que para lograr el nivel de detalle mencionado, durante el mismo modelado del proceso se requiere ir realizando los pasos de identificación de tipos y fuentes, lo que muestra la naturaleza iterativa de la metodología, haciendo evidente que las distintas fases y pasos de la misma no constituyen actividades aisladas, sino que están interrelacionadas unas con otras. Esto muestra también la utilidad de un enfoque de modelado que permita ir incluyendo en los modelos del proceso los temas, áreas y fuentes de conocimiento que se vayan identificando durante la captura y

análisis del proceso. La principal utilidad del enfoque presentado, es que permite identificar lo que deben conocer las personas que deberán realizar ciertas actividades, así como las fuentes que podrían serles de ayuda para obtener dicho conocimiento. Una vez hecho esto, el siguiente paso sería entonces identificar la manera en que el conocimiento requerido y generado en las actividades está fluyendo a lo largo del proceso.

VI.1.2.3 Identificación de flujos de conocimiento

Los flujos del conocimiento que se dan dentro del proceso pueden ser analizados con base en diversos diagramas. Para ejemplificar esto, tomaremos los flujos de conocimiento que se dan alrededor de las solicitudes de mantenimiento, proceso en el cual interviene activamente el Sistema de Control de Proyectos (SCP). De esta manera, aprovecharemos este ejemplo para posteriormente mostrar el uso del marco de trabajo para análisis de sistemas de AC (descrito en el Capítulo V), para analizar el rol del SCP como facilitador del flujo del conocimiento.

La Figura 40 muestra un diagrama de actividad en donde se puede observar que el SCP es donde se capturan las solicitudes de cambio. Una vez capturada una solicitud, el SCP se lo informa al responsable del sistema a modificar por medio de un correo electrónico, para que posteriormente éste inicie la atención de dicha solicitud. Lo anterior infiere que la información relacionada con cada solicitud debe poder ser accesible a través del SCP. Esta información en ocasiones incluye parte de los requerimientos para las modificaciones. Sin embargo, frecuentemente los detalles de los requerimientos sólo se encuentran en las libretas de trabajo de los técnicos responsables de su captura, por lo que se ha utilizado un paquete de conocimientos y no una fuente para representar los requerimientos. De esta manera en el diagrama se hace evidente una posible pérdida de conocimiento debido a que no se está guardando, o por lo menos no de forma adecuada.

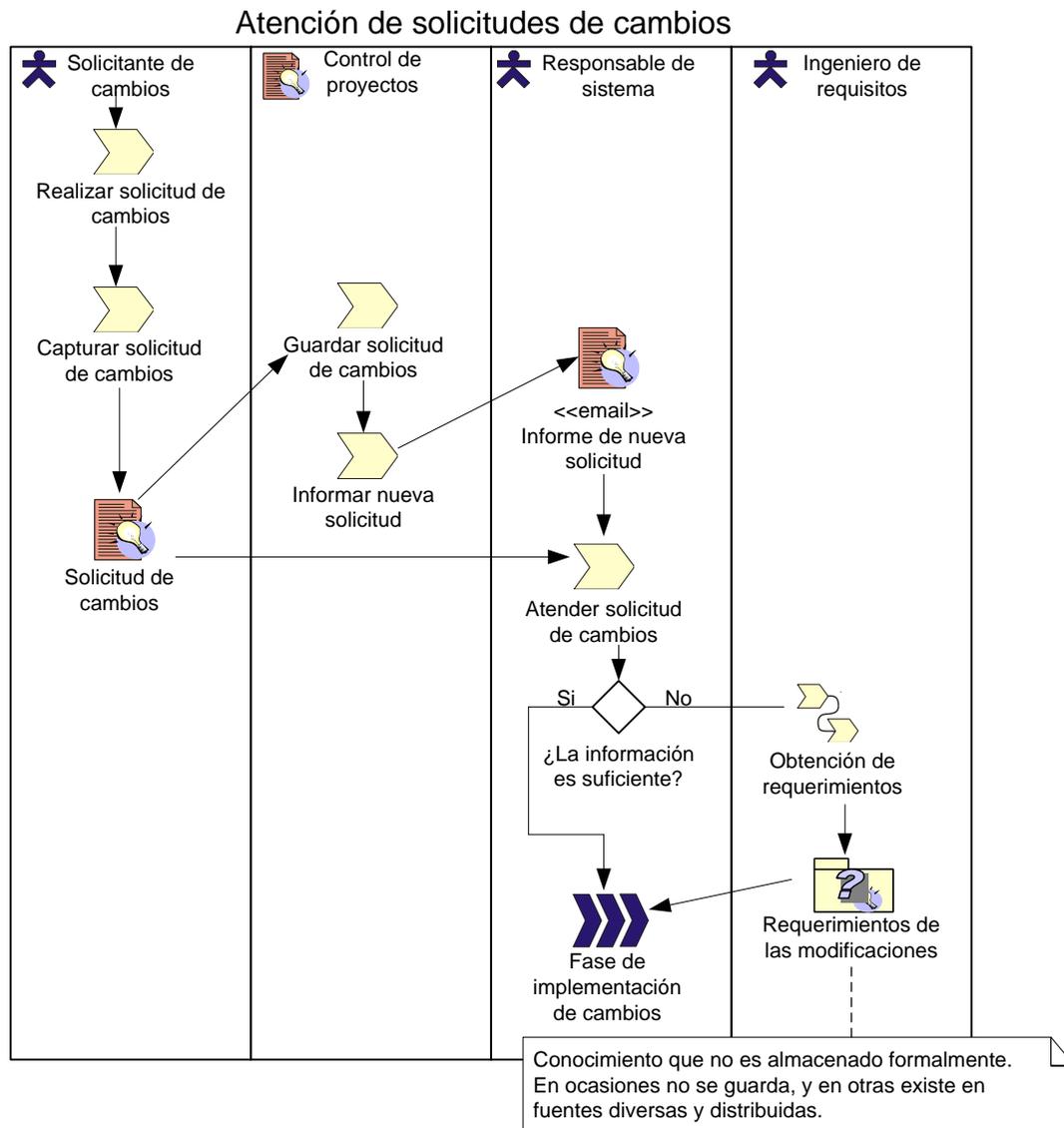


Figura 40. Diagrama de actividades que muestra flujos de conocimiento, por medio de la transferencia de fuentes entre actividades y/o roles o actores del proceso.

La información relacionada, tanto a la solicitud como a los requerimientos se muestra en la Figura 41, lo que da una idea de la información que puede ser transmitida a través del SCP. Sin embargo, como se mencionó, la información de requerimientos no tiene una fuente específica, en ocasiones se encuentra en la misma solicitud o en las libretas de trabajo de los encargados de la captura de los requerimientos, en otras ocasiones se utilizan

memorandos con dicha información, entre otros medios. Por lo tanto, al analizar los casos como los presentados en los diagramas de la Figura 40 y Figura 41, donde se observan paquetes de conocimiento que no se encuentran en una fuente específica, y el conocimiento contenido en dicho paquete (respectivamente), podemos darnos una idea del conocimiento que podría estarse perdiendo, y comenzar a tomar medidas para evitar esto.

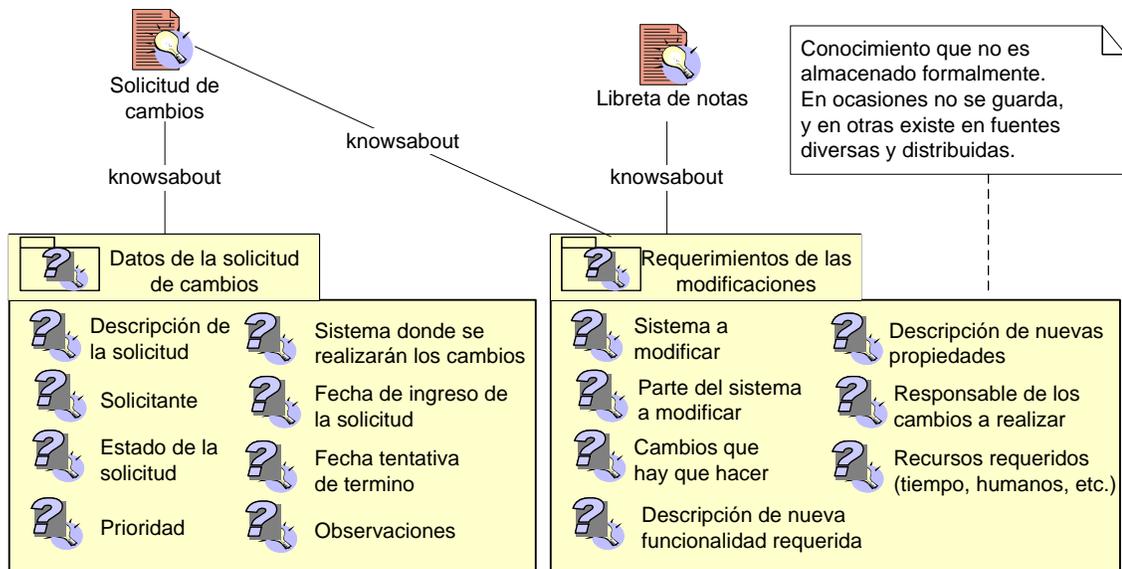


Figura 41. Diagrama de paquetes que muestra la información de la solicitud de cambios. Se muestran temas concretos que pueden obtenerse de la solicitud de cambios, junto con los relacionados con los requerimientos.

Una vez observada la importancia del SCP en el flujo del conocimiento a lo largo del proceso, se decidió analizar con mayor detalle la manera en que la información de una solicitud de cambios fluye a lo largo del proceso. Para este fin fueron útiles los diagramas de flujos de conocimiento de fuentes específicas, como el mostrado en la Figura 42.

El diagrama de la Figura 42 muestra las actividades donde el conocimiento relacionado con la solicitud de cambios es generado o utilizado, mostrando la secuencia de estas actividades. De esta manera, se facilita seguir el flujo de la información y conocimiento que se encuentra almacenada en una determinada fuente, a la vez de observar

las actividades donde dicha fuente interviene. Como se puede observar en el diagrama, ciertos datos han sido agrupados en paquetes de conocimiento para simplificar el modelo. Sin embargo, de ser necesario, podría después modelarse cada paquete, o incluso, la secuencia que se da dentro de cada flujo de actividades. En nuestro caso, este nivel de detalle fue suficiente para el análisis. La información contenida en este diagrama fue útil posteriormente al analizar el SCP como apoyo al flujo del conocimiento, dado que permitió identificar el conocimiento que puede fluir a través de este, así como las actividades dónde el mismo es útil.

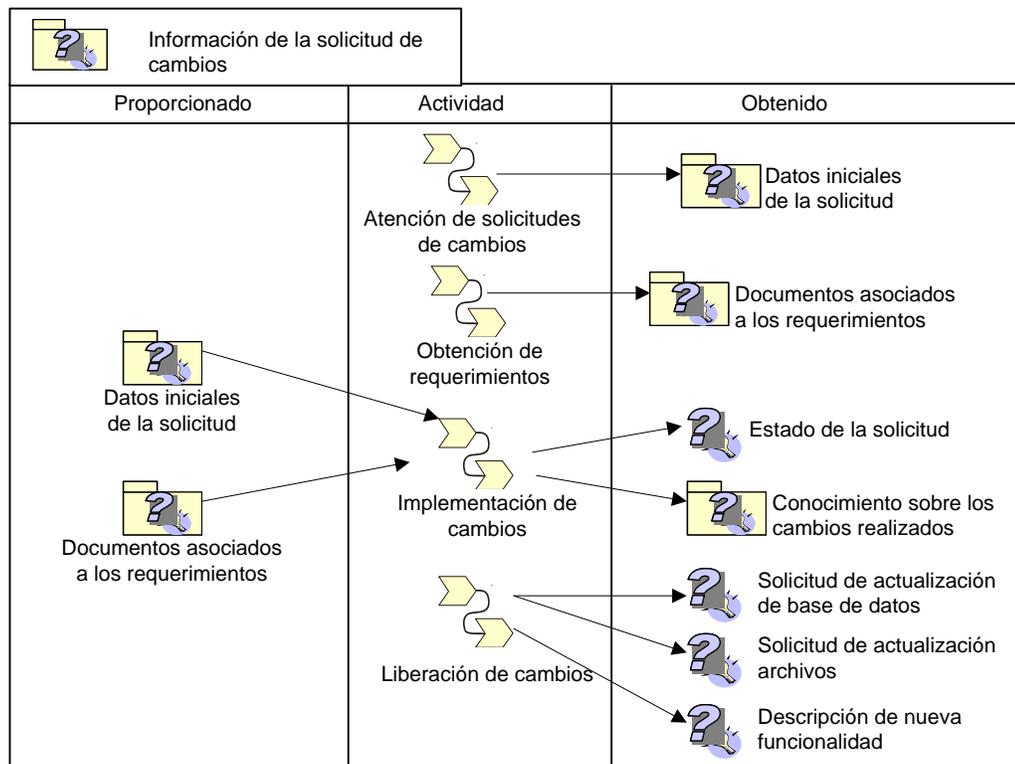


Figura 42. Diagrama de transferencia de conocimiento de una fuente que muestra el flujo del conocimiento relacionado con la solicitud de cambios.

Durante el análisis se observó que una parte importante de la información relacionada con una solicitud de cambios, principalmente aquella que describe los cambios

realizados, como los archivos que fueron modificados, se encuentra distribuida en varias fuentes como lo muestra la Figura 43.

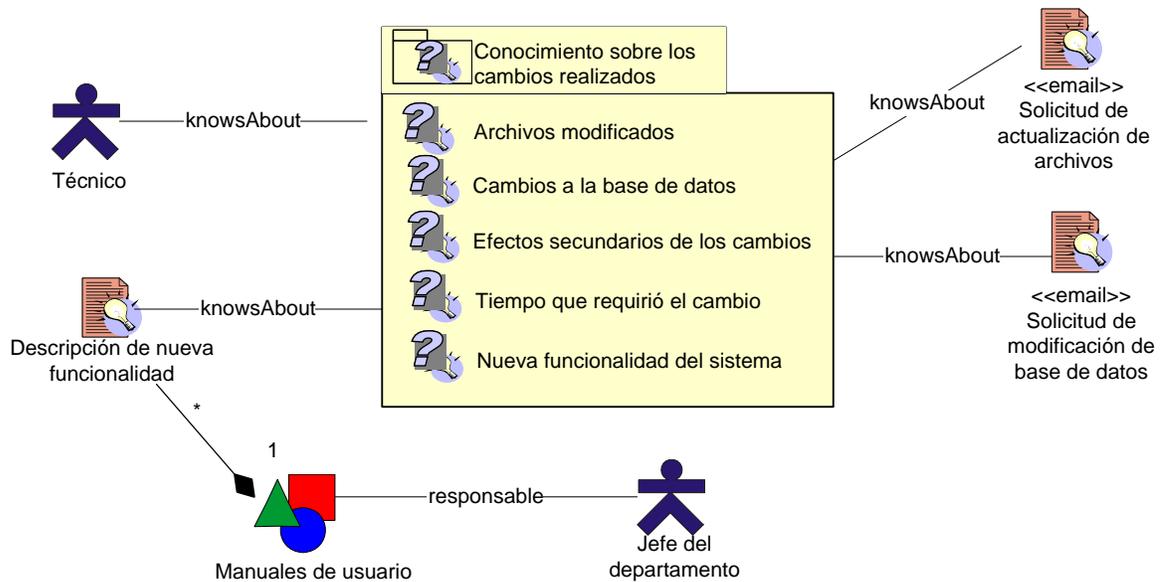


Figura 43. Diagrama de clases que muestra las fuentes que contienen información sobre los cambios realizados.

Aun cuando en ocasiones parte de la información de estas otras fuentes es almacenada como observaciones dentro de la misma, esto no es el caso común. Además, algunas de estas fuentes son utilizadas como canales de transmisión de conocimiento, como es el caso de dos tipos de correos electrónicos: “Solicitud de actualización de archivos” y “Solicitud de modificación de base de datos”. En la Figura 44 se presenta un diagrama de transferencia de conocimientos que muestra el rol que dichas fuentes juegan como canales de transmisión de conocimientos entre actores del proceso. Este tipo de diagrama forma parte de las extensiones a SPEM propuestas en esta tesis (Capítulo IV), y permiten identificar los principales actores que participan en una transferencia de conocimientos. De esta manera, se facilita identificar los mecanismos que sirven para transferir conocimientos (en el caso del ejemplo: los correos electrónicos), así como a las personas, roles o actores que intervienen, ya sea aportando y recibiendo el conocimiento que es transferido.

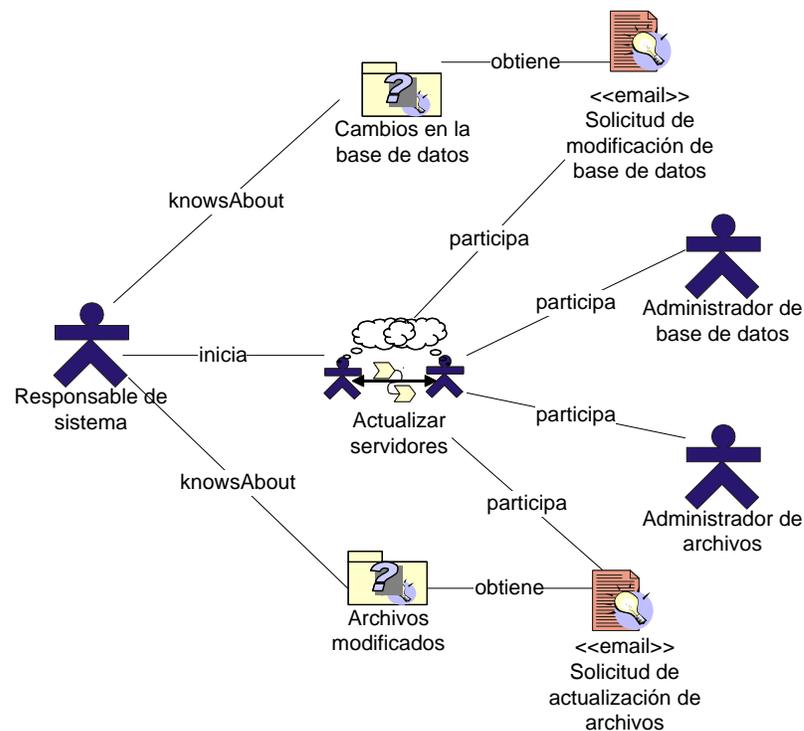


Figura 44. Diagrama de transferencias de conocimiento entre fuentes que muestra las fuentes que participan en la transferencia de la información relacionada con las solicitudes de actualización de servidores.

Como podemos observar del ejemplo presentado en esta sección, el SCP juega un papel importante dentro del proceso, sobre todo debido a que todo el proceso se realiza alrededor de las solicitudes de cambio. Sin embargo, mucha de la información relacionada con dichas solicitudes, en vez de encontrarse almacenada junto con la solicitud a la que pertenece, se encuentra distribuida en diversas fuentes, algunas de las cuales, debido a la información que contienen sirven como canales de transmisión de conocimiento. Lo anterior nos llevó a preguntarnos si no es posible agrupar todas estas fuentes con el fin de facilitar el acceso a toda la información relacionada con una solicitud particular. Esto fue uno de los problemas observados dentro del análisis del proceso, y que dieron paso a las propuestas de mejora al flujo del conocimiento que se describen más adelante.

VI.1.2.4 Problemas con el flujo del conocimiento

De los diagramas y el ejemplo descrito en la sección anterior, podemos extraer los siguientes problemas:

- El principal problema observado fue que la información asociada a una solicitud de cambios se encuentra distribuida en diversas fuentes.
- No es fácil encontrar dicha información. Si un técnico requiere saber qué cambios se hicieron debido a una solicitud particular, no le resultaría fácil saberlo, ya que, por ejemplo, tendría que encontrar los correos electrónicos de solicitudes de actualización de servidores y base de datos, lo cual no es fácil si no se tienen datos precisos sobre la fecha en que se enviaron.
- No existe un estándar para la captura y redacción de los requerimientos de una solicitud de cambios. Incluso, en ocasiones no hay fuentes formales dónde ir a buscar dichos requerimientos. Por lo tanto, si pasado un determinado tiempo se quiere regresar a analizar los requerimientos que dieron paso a una solicitud, podría no ser posible. Incluso, aún cuando se puedan obtener los documentos con dicha descripción, saber a qué solicitud específica pertenecen podría ser difícil, debido a que no hay una forma de asociarlos.

Queda claro que los problemas anteriores, pueden ser identificados al analizar los modelos del proceso, como se fue haciendo a lo largo de las secciones previas. De este análisis, se observó que una posible solución podría ser la asociación de todas las fuentes relacionadas a una solicitud de cambios con la misma solicitud, y de esta manera, utilizar el SCP como la principal herramienta para el acceso a dichas fuentes. A continuación se presentan las principales propuestas para mejorar el proceso. Con base en estas, se decidió posteriormente analizar el SCP desde el punto de vista del apoyo que actualmente proporciona al flujo del conocimiento dentro del proceso, y así definir la manera en que las modificaciones al SCP afectarían este apoyo. Lo anterior se hizo analizando los modelos donde interviene el SCP y utilizando el marco de trabajo descrito en el Capítulo V para

describir la forma en que el SCP participa como apoyo al flujo del conocimiento. En las siguientes sub-secciones se resume esto último.

VI.1.2.5 Propuestas para mejorar el flujo del conocimiento

Del análisis del proceso y del SCP, se obtuvieron una serie de propuestas que podrían mejorar significativamente el flujo del conocimiento dentro del DI, sin necesidad de grandes cambios en su infraestructura ni en sus procesos actuales, al mismo tiempo que se aprovecha uno de sus sistemas clave como apoyo al proceso. A continuación se listan estas propuestas.

Integrar documentos relacionados con las solicitudes de modificaciones.

Durante el análisis del proceso, se observó que la información relacionada con las solicitudes de cambios se encuentra distribuida en diversas fuentes. Por ejemplo, en ocasiones existen correos electrónicos que los usuarios envían a los técnicos explicándoles algunos de los requerimientos para los cambios. En otras ocasiones existen memorandos, o documentos que describen parte de dichos requerimientos, como ejemplo podemos poner las modificaciones que se realizaron al sistema de la dirección de estudios de posgrado debido a los cambios en los reglamentos de dicha dirección en una fecha cercana a la realización del estudio. En este caso, parte de los requerimientos están definidos en el documento que describe el nuevo reglamento. Gran parte de los documentos que pueden servir para obtener información relacionada con los cambios solicitados se encuentran disponibles en formato electrónico. Viendo esto, se ha propuesto que el SCP sea modificado de forma tal que en cada solicitud de cambios los técnicos puedan incluir ligas a documentos asociados con dichas solicitudes, indicando qué tipo de información podría ser encontrada en los mismos. Por ejemplo, requerimientos o información del diseño del módulo a modificar, documentos que describen la nueva funcionalidad agregada al sistema, etc.

Lograr la integración de las solicitudes con documentos que puedan estar relacionados, debería facilitar su acceso a los técnicos encargados de hacer las

modificaciones, facilitando así la aplicación de conocimiento actualmente existente dentro del departamento. Así mismo, sería más fácil saber cuáles fueron los requerimientos que dieron pie a una solicitud de cambio determinada.

Una forma de lograr lo anterior es por medio de un **mapa de conocimientos**. Los mapas de conocimiento son herramientas que permiten asociar las distintas fuentes existentes en una organización, con el conocimiento que contienen y las actividades donde podrían ser útiles. De esta forma, el mapa puede ayudar a encontrar de forma rápida fuentes útiles para obtener información para realizar tareas específicas. Los ejemplos presentados en esta sección muestran que los modelos pueden ayudar en la creación de dicho mapa, ya que en los mismos se identifican las fuentes donde pueden obtenerse los distintos temas de conocimiento requeridos en las actividades del proceso. Así mismo, al crear las tablas de descripción de fuentes, se indica el lugar donde cada fuente puede ser encontrada. Esta información es indispensable en la creación de un mapa de conocimientos.

Integrar las solicitudes de actualización de servidores a las solicitudes de modificaciones. Los correos electrónicos que los técnicos envían para solicitar la actualización de los servidores de archivos y base de datos (ver Figura 43 y Figura 44), contienen información útil para saber lo que se cambió debido a una solicitud de cambios específica. Sin embargo, es difícil saber asociar dichos correos a la solicitud que los generó si el técnico que desea hacer dicha relación no tiene conocimiento sobre el día específico y sobre la solicitud que estaba atendiendo. Para buscar un correo en particular, se tiene que hacer una búsqueda dentro del buzón de correo del técnico o del administrador de archivos o base de datos, con el tiempo que esto podría implicar, y sin la seguridad de que se encontrara el correo que se quiere. Viendo esta situación, se propone que dichos correos sean enviados desde el SCP, de forma tal que una copia del correo enviado se almacene junto con la solicitud a la que pertenece.

Al agregar una opción para el envío de los correos dentro del SCP, se podría definir un formato que estandarice el tipo de información que dichos correos deben contener y la organización de la misma, facilitando a los técnicos la externalización de su conocimiento

dentro de estos correos. Lo anterior ayudaría también a facilitar el acceso a información que podría describir con cierto detalle las partes del sistema fueron modificadas debido a una solicitud particular. Por ejemplo, si en un momento determinado un sistema empieza a fallar al tratar de acceder a la base de datos, podría buscarse dentro del SCP si existe alguna solicitud de cambios que requirió que se modificara la parte de la base de datos que ocasiona el fallo. De esta manera, se podría analizar dicho cambio para ver si éste fue el causante del problema, y de ser así, tomar las medidas pertinentes.

Agregar campos para capturar información sobre las soluciones de cada solicitud de modificación. Durante el estudio se observó que los técnicos en ocasiones capturan descripciones de lo que requirieron hacer para resolver una solicitud de cambios. Sin embargo, el SCP no cuenta con campos específicos para esto, por lo que los técnicos deben hacer esto dentro de un campo destinado a observaciones. Con el fin de facilitar la captura de información relacionada con las soluciones dadas a las solicitudes de cambio, se propone extender el SCP para agregar formas para capturar información relacionada con las soluciones a cada solicitud, por ejemplo, el tiempo requerido para solucionarlas, las partes del sistema que requirieron ser modificadas, posibles efectos secundarios de los cambios, los documentos que se hayan generado como consecuencia de la atención a dicha solicitud, etc. Esto ayudaría a los técnicos en el proceso de externalización de su conocimiento, al definirles el tipo de información que deben capturar, así mismo, les ayudaría a recordar con posterioridad el trabajo que requirieron invertir en cada solicitud. Lo anterior podría ser útil, por ejemplo, a la hora de que los técnicos realizan su reporte anual de actividades. Ya que el sistema estaría concentrando gran parte de la información que requieren para realizar dicho reporte, como lo es la cantidad de solicitudes recibidas, la cantidad de solicitudes atendidas, el tiempo invertido en la obtención de los requerimientos, en la realización de los cambios, etc.

Incrementar las opciones para la búsqueda de solicitudes de modificaciones. El sistema permite a los técnicos organizar las solicitudes que tienen asignadas por prioridad, fecha de ingreso, responsable, solicitante, estado, entre otros. Al aumentar la cantidad de información asociada a cada solicitud, será posible incluir más opciones de búsqueda, por

ejemplo, todas las solicitudes que han requerido cambios en un archivo específico, o en una tabla específica de la base de datos.

Con el fin de conocer de mejor manera las posibles repercusiones que podrían ocasionar los cambios en el SCP, se realizó un análisis de la forma en que actualmente el SCP apoya el flujo del conocimiento, y la forma en que dicho apoyo sería modificado de realizarse los cambios en el sistema.

VI.1.2.6 El Sistema de Control de Proyectos como herramienta de apoyo al flujo del conocimiento

Durante el estudio se observó que el flujo del conocimiento dentro del proceso de mantenimiento del DI se da entorno a las solicitudes de cambio. Debido a que estas solicitudes son por lo general manejadas dentro del SCP, dicho sistema actualmente funciona como el principal mecanismo que interviene en el flujo del conocimiento dentro del proceso. Fue con base en esta observación que varias de las propuestas están enfocadas en modificar dicho sistema con el fin de mejorar el flujo de conocimiento. Para identificar la forma en que dicho flujo sería afectado de realizarse los cambios al sistema, se requiere primero conocer la forma en que el SCP influye en dicho flujo, para lo cual se realizó un análisis del sistema siguiendo el marco de trabajo descrito en el Capítulo V. La Tabla XV presenta el resultado de este análisis, el cual corresponde con la descripción del SCP como facilitador del flujo del conocimiento por medio de la matriz de definición de las características del marco de trabajo.

Tabla XV. Descripción del SCP como facilitador del flujo del conocimiento usando la matriz de definición de características consideradas en MAHAC.

Característica	Nivel / Valor	Observaciones
Dominio del conocimiento		
Uso	Procesos del dominio de trabajo	Proceso de mantenimiento del software. La herramienta ayuda a los técnicos a obtener información útil para realizar las modificaciones solicitadas o corregir los problemas reportados.
Alcance	Grupo Intra-organizacional	Dentro de este dominio de aplicación, los beneficiarios del sistema son los Técnicos del DI.
Dominio	Producto-Servicio	<ul style="list-style-type: none"> - Información de las solicitudes de modificaciones, como los requerimientos, o el problema reportado por el cliente. - Información histórica que puede ser útil a los Técnicos para atender solicitudes de cambios. Por ejemplo, solicitudes similares. - Información histórica de los sistemas mantenidos por el DI, que puede ser usada para obtener datos estadísticos, como los problemas más comunes, los sistemas o módulos más problemáticos, etc.
Estructura del conocimiento		
Tácito	-----	
Explícito	EE/D ENE/TL	El SCP almacena información en forma de datos estructurados como registros en una base de datos relacional. El SCP permite guardar observaciones y detalles sobre las solicitudes en forma de texto libre.
Soporte al flujo de conocimiento		
Creación	-	
Externalización	+ -	Provee medios para capturar cierto tipo de información de forma estructurada y no estructurada
Almacenamiento	+	Es básicamente un repositorio de información
Transferencia	+ -	Debido a que el SCP es accesible a todo el DI, tiene la capacidad de servir para compartir información entre los miembros del DI.
Tácito	+ -	En ocasiones, los técnicos describen lo que tuvieron que hacer para resolver una solicitud.
Explícito	+	Prácticamente toda la información contenida en la solicitud cae dentro de este tipo
Internalización	+ -	Permitir ordenar información de diversas formas.
Recuperación	+ -	El ordenamiento ayuda a buscar información.
Interpretación / Filtrado	-	
Aplicación	-	
Aspectos técnicos		
Autonomía	No-autónomo	
Distribución	Centralizado internamente	La información se encuentra en un servidor dentro del DI.

Como se puede observar, el SCP apoya en la obtención y captura de conocimiento dentro del proceso principal del DI, el cual es el mantenimiento del software que apoya las actividades del CICESE. Si bien dentro del dominio de aplicación definido el SCP tiene alcance intra-organizacional, es importante decir que también puede prestar ciertos servicios a los usuarios de los sistemas mantenidos por el DI. Con respecto al papel que el SCP juega como habilitador del flujo de conocimiento, se observó que prácticamente es un repositorio de información, por lo que caería dentro de la categoría de sistema de apoyo a la etapa de almacenamiento. Sin embargo, presenta características que apoyan de forma parcial actividades de externalización e interanilización. Además, debido a que el sistema es compartido por todos los miembros del DI, e incluso es accesible a los usuarios, tiene el potencial de servir como un mecanismo de transmisión de conocimiento, tanto en el espacio como en el tiempo. Por ejemplo, puede servir para informar a los usuarios sobre el estado que guarda cada una de sus solicitudes, o sobre el trabajo que implicó la solución a ellas. Así mismo, les puede servir a los técnicos para recuperar información sobre solicitudes pasadas, por ejemplo a la hora de hacer sus reportes anuales de actividades.

Después de analizar al SCP y los problemas identificados durante el análisis del proceso. Se observó que el SCP tiene el potencial de servir de mejor manera como un facilitador del flujo del conocimiento relacionado con las solicitudes de mantenimiento. Con base en las propuestas para modificar tanto el diseño como el uso del SCP, a continuación se describe la forma en que la realización de las mismas podría influir en las características del SCP como facilitador del flujo del conocimiento.

Integrar documentos relacionados con las solicitudes de modificaciones. Este cambio permitirá mejorar la etapa de recuperación, debido a que facilitaría el acceso a las fuentes de información existentes, sobre todo, aquellas relacionadas con solicitudes determinadas, lo que apoyaría también en la etapa de filtrado, al permitir sólo recuperar aquellos documentos realmente relacionados con una solicitud determinada. Adicionalmente, este cambio podría repercutir positivamente en la etapa de transferencia de conocimiento explícito.

Integrar las solicitudes de actualización de servidores a las solicitudes de modificaciones. Como en el caso anterior, este cambio también repercutiría positivamente en las etapas de recuperación y filtrado, así como en la transmisión de conocimiento explícito al facilitar el acceso a dichas solicitudes. Adicionalmente, podría apoyar en la etapa de externalización; ya que al estandarizar el formato de las solicitudes se facilitaría la captura de la información que deben contener, reduciendo el esfuerzo de los miembros del DI al no tener que pensar demasiado en qué datos incluir en una solicitud. Este cambio incrementaría también el apoyo que el SCP da como mecanismo de almacenamiento de información, ya que la información de dichas solicitudes de actualización también se guardaría en el sistema, evitando que estas se pierdan entre las carpetas de correo de los técnicos del DI. Para finalizar, este cambio también repercutiría en el dominio del conocimiento del sistema, debido a que estaría ayudando a conocer los pasos que se deben seguir al realizar una solicitud, particularmente, indicaría qué acciones deben realizarse después de realizado un cambio, así como la forma de llevarla a cabo.

Agregar campos para capturar información sobre las soluciones de cada solicitud de modificación. Este cambio ayudaría a incrementar el dominio del conocimiento manejado por el SCP. Así mismo, apoyaría en la etapa de externalización, al facilitar la captura de la solución dada a una solicitud de una forma estandarizada, indicando la información que debe ser capturada. Adicionalmente, ayudaría en la conversión y transferencia de conocimiento que comúnmente es tácito dentro del DI, particularmente del tipo “saber cómo”, debido a que apoyaría a los miembros del DI a externalizar la forma en que una determinada solicitud fue resuelta.

Incrementar las opciones para la búsqueda de solicitudes de modificaciones. Este cambio está enfocado a mejorar la etapa de internalización, tanto en la recuperación como en el filtrado de información. Ayudando a los miembros del DI a acceder de forma más fácil y rápida a la información almacenada en el SCP.

VI.1.3 Resultados del caso de estudio

El principal resultado de la realización del caso de estudio fue que nos ayudó a mostrar la utilidad de la metodología y el apoyo que las técnicas desarrolladas dan a las etapas de obtención y análisis de la información. De los resultados podemos decir que tanto la metodología como las técnicas propuestas para apoyar su aplicación, cumplen sus objetivos, dado que:

- Se mostró como el lenguaje de modelado propuesto efectivamente apoya en la obtención de información requerida durante las fases de análisis de la metodología.
- Se mostró que la aplicación de la metodología puede ayudar a identificar los sistemas que intervienen en el flujo del conocimiento, así como la forma en que estos interfieren dentro del flujo.
- Lo más importante. Se mostró que del análisis de la información obtenida de la aplicación de la metodología se pueden extraer propuestas para mejorar el flujo del conocimiento que no requieran cambios grandes en los procesos de trabajo, y que, incluso, consideren como parte central las herramientas que apoyan dichos procesos de trabajo, lo cual forma parte del objetivo principal de esta tesis.
- Así mismo, el uso de la metodología, y particularmente las técnicas propuestas para apoyarla, permiten obtener información que posteriormente podría ser utilizada para el diseño y desarrollo de propuestas para mejorar el flujo del conocimiento. Por ejemplo en el caso de la creación de mapas de conocimiento; uno de los principales mecanismos para facilitar el acceso a las fuentes de conocimiento existentes en una organización.

Otro de los resultados importantes de la realización del caso de estudio, fue que durante su ejecución se obtuvieron algunas observaciones que nos ayudaron a ir mejorando las distintas propuestas que componen la metodología. Por ejemplo, se hicieron ajustes en el lenguaje de modelado para mostrar de mejor manera la información requerida durante la fase de análisis.

Sin embargo, existen detalles observados durante el caso de estudio que han quedado pendientes. Principalmente, fue difícil decidir como catalogar los sistemas que intervienen en el flujo del conocimiento dentro de los modelos del proceso, dado que sólo se tenían dos opciones: definirlos como una fuente de conocimiento, o como un producto de trabajo básico de SPEM. Sin embargo, se llegó a la conclusión de que dicho tipo de sistemas deberían caer en otra categoría, dado que no son propiamente fuentes, ni productos del trabajo, sino que son actores que de una u otra forma intervienen en el flujo del conocimiento. Debido a esto, se hace la siguiente propuesta para mejorar el lenguaje de modelado: *hacer una distinción entre actores normales, y actores que intervienen en el flujo del conocimiento*. Dentro de estos últimos, hacer una distinción entre personas y sistemas o herramientas.

Finalmente, el caso de estudio también fue útil para obtener algunas lecciones importantes a considerar, las cuales se presentan a continuación, seguidas de las principales limitaciones del caso de estudio.

VI.1.3.1 Lecciones aprendidas del caso de estudio

Adicionalmente a los resultados presentados anteriormente, durante el caso de estudio se observaron algunos aspectos que son importante tener en cuenta a la hora de proponer alguna estrategia o sistema de AC para apoyar al DI.

Los miembros del DI tienden a realizar acciones de administración del conocimiento por iniciativa propia. Durante el estudio quedó claro que los miembros del DI realizan actividades de AC aun cuando no lo hacen de forma consiente. Esto concuerda con lo que otros estudios han encontrado, en el sentido de que las organizaciones de software tienden a realizar actividades de AC aun cuando no tengan sistemas o estrategias explícitas para administrar su conocimiento [Lutters y Seaman, 2007; Meehan y Richardson, 2002; Ward y Aurum, 2004]. Esto hace evidente la necesidad del manejo del conocimiento por parte de este tipo de organizaciones.

Lo anterior es relevante debido a que puede ser un factor que facilite la adopción de estrategias o sistemas de AC, sobre todo si se consigue que dichos sistemas o estrategias se enfoquen en mejorar las prácticas actuales de la organización, y no sólo en introducir nuevas prácticas o herramientas. Esto hace notar la importancia de una metodología como la propuesta en esta tesis, enfocada a apoyar en la obtención de información que permita lograr este objetivo.

Los miembros del DI no estaban concientes del conocimiento que tienen, ni del que se requiere para realizar sus actividades. En el estudio se observó que los miembros del DI realizan sus actividades sin pensar en el conocimiento que requieren para hacerlas, ni en el que ellos realmente poseen. Con base en el análisis de los diagramas, y al momento de estarles haciendo las entrevistas, algunos de ellos hicieron comentarios haciendo ver que no habían pensado en todo el conocimiento que se necesitaba para realizar sus actividades, ni en la cantidad de conocimiento que ellos tenían, y por tanto, lo importante que son ellos para el buen desempeño del departamento. El estudio ayudó a que vieran la relevancia que su conocimiento tiene dentro del proceso. Así mismo, comenzaron a entender las dificultades que podría tener el departamento si alguno de ellos faltaba y no dejaba a alguien capacitado para tomar su lugar. Esto puede hacer que las personas involucradas tomen una actitud enfocada en buscar la forma de evitar ese problema. Por ejemplo, informando a quien pudiera ocupar su lugar sobre los documentos que ellos habían generado y que podrían ser útiles, o pensando qué tipo de capacitación deberían ellos dar a un posible sustituto.

Las necesidades de conocimiento varían con el tiempo y la experiencia. Un punto importante a tomar en cuenta, es que las necesidades de conocimiento resultaron distintas dependiendo del tiempo que los técnicos tenían trabajando en el DI. Cuando son novatos, tienden a preocuparse más por cuestiones técnicas, por ejemplo, los lenguajes de programación usados, las herramientas de desarrollo, etc. Por otra parte, conforme obtienen experiencia en aspectos técnicos, su preocupación se enfoca más hacia los procesos que el sistema debe de apoyar, así como la forma en que los usuarios interactúan con el sistema. Este tipo de observaciones no la hemos visto reportada en los trabajos relacionados que

hemos estudiado. Sin embargo, es importante tenerla en cuenta al diseñar sistemas de AC para este tipo de procesos, ya que el sistema deberá apoyar a personas con necesidades de conocimiento que dependen del perfil del usuario y que éste varía con el tiempo.

Con base en las dos observaciones anteriores podemos decir que la metodología fue útil para lograr otro de los objetivos que dieron pie a su desarrollo. Al inicio de esta tesis se comentó que uno de los factores para lograr el éxito de una estrategia o sistema de AC en una organización pequeña, es lograr primero que los miembros de dicha organización vean sus procesos en términos del conocimiento involucrado, y las implicaciones que la AC tienen en ellos [Sparrow, 2001; Wong, 2005]. El caso de estudio fue útil precisamente para lograr que los miembros del DI comenzaran a percatarse de la importancia de administrar el conocimiento involucrado en su trabajo. Más aun, pudimos darnos cuenta de que actualmente realizan actividades de AC por iniciativa propia. Esto facilita que perciban la importancia de la AC para sus actividades, lo que podría también facilitar la aceptación de una estrategia o sistema de AC, sobre todo si este se ajusta a su forma de trabajo y considera los sistemas o herramientas que ellos utilizan para sus actividades diarias. El caso de estudio mostró que el uso de la metodología propuesta en esta tesis puede ayudar al logro de éste último objetivo.

VI.1.3.2 Limitaciones del caso de estudio

La principal limitación del presente caso de estudio es quizá el hecho de que el mismo fue realizado por quien definió la metodología, lo que podría ser un factor para restar fuerza a los resultados y lecciones obtenidas del mismo. Debido a esto, se ha decidido incluir en este capítulo algunas observaciones sobre la metodología que fueron obtenidas de una persona ajena al desarrollo de la misma. Esta persona aplicó la metodología propuesta en esta tesis para realizar un caso de estudio en una empresa con un giro distinto al del desarrollo de software. En la siguiente sección se resumen las principales observaciones obtenidas de quien realizó este segundo caso de estudio.

VI.2 Un segundo caso de estudio

Este segundo caso de estudio fue realizado por una empleada de una empresa industrial de origen Mexicana con sucursales en varias partes de la república. El estudio se hizo en una de estas sucursales, específicamente una ubicada en la ciudad de Ensenada, B.C. Es importante tener en cuenta que el objetivo de la empresa es que el sistema resultante pueda ser implementado en toda la empresa en un futuro. A continuación se resumen los principales datos obtenidos de un cuestionario que se le hizo a la realizadora del estudio. El cuestionario completo se encuentra en el Apéndice C.

VI.2.1 Resumen de los resultados del cuestionario

La realizadora de este caso labora en el área de sistemas de la empresa. Esta empresa se dedicada a la elaboración de envases metálicos. El proceso estudiado fue el que consiste en la transformación del metal, y el cual es el proceso inicial, definido como clave en el total de las actividades de la empresa. El estudio completo duró ocho meses, aunque no con dedicación de tiempo completo.

Según la persona que aplicó la metodología, ésta resultó fácil de aprender, llevándole un mes entenderla con base en dos artículos publicados y tres entrevistas cortas con quien esto escribe. Cabe mencionar que el modelado de los procesos se hizo sólo a un alto nivel, por lo que sólo fue necesario utilizar la adaptación de la técnica de gráfica rica que se propuso en este documento (Capítulo IV, sección IV.2). El nivel de detalle que proporciona esta técnica fue suficiente para las necesidades del estudio.

La principal dificultad con el entendimiento de la metodología fue en relación a la definición de un metamodelo que describiera la estructura de las fuentes y tipos de conocimiento, es decir, su clasificación y relaciones. Lo anterior se debió principalmente a la carencia de conocimientos en el área de la orientación a objetos y el lenguaje UML por parte de la realizadora del caso de estudio. De lo que se deduce que se requiere conocimiento de UML, aunque sea básico, para entender todas las propuestas que incluye la metodología.

De las respuestas del cuestionario se observa que la parte más importante en este segundo caso de estudio fueron la identificación de fuentes y tipos de conocimiento, ya que, en palabras de la encuestada: *“en esta etapa es buscar (en documentos, sistemas, estructuras) lo que debe saberse dentro del proceso a estudiar, además que te permite interactuar con las personas que al final son las que traen el conocimiento que queremos preservar y las necesidades reales como usuarios finales”*. De lo anterior, podemos decir que la metodología, en este caso cumplió con los objetivos para los que fue diseñada, debido a que ayudó a *“la identificación del conocimiento requerido,[...] lo que necesitan para trabajar en el proceso estudiado, además de la identificación de flujos de conocimiento”*.

Con respecto a la utilidad de la metodología en el logro de los resultados del estudio, la encuestada considera que fue *“muy útil, fue la punta de lanza, la guía para poder entender lo que se necesita dentro de una herramienta de gestión de conocimiento”*. Cabe destacar el resultado reportado por la encuestada, el cual fue un prototipo de un portal de conocimiento para la empresa, mismo que fue mostrado y evaluado con el personal de ésta, quienes en su totalidad consideraron el producto muy útil para sus actividades: *“En el análisis del resultado de la evaluación del grado de aceptación, la mayoría de las personas (85%) considera extremadamente útil el portal y el resto lo considera muy útil, en cuanto a la facilidad de uso la mayoría opina que es una aplicación muy fácil de usar, por lo que se puede concluir que el grado de aceptación del portal es muy alto”*.

Para finalizar, se presentan las siguientes frases, extraídas de un correo electrónico enviado por la persona encargada de realizar el estudio, y resumen el papel que jugó la metodología como apoyo a la realización de los objetivos del mismo⁴.

⁴ Detalles de este estudio pueden encontrarse en el reporte de proyecto de maestría de la estudiante Jacqueline Ivone Lavandera Macedo, de la Maestría en Tecnologías de la Información y la Comunicación de la Universidad Autónoma de Baja California, campus Ensenada. Al momento de la realización de la presente tesis, dicho reporte no se encontraba culminado.

“Lograr definir el metamodelo⁵ fue muy entretenido, revise documentación relacionada a notación UML y las relaciones entre clases, sin embargo resultó algo muy curioso porque este modelo representa lo que ya se había analizado, finalmente describe el proceso que se siguió para la definición de temas de conocimiento, fuentes, tipos y su categorización, estuve tratado de encontrar el hilo negro siendo que el metamodelo que Oscar Mario presenta en un artículo representa la esencia del análisis previo, hice solo algunas omisiones debido a que a mi juicio no aplicaban a este proyecto, yo no quería tomar el modelo sin comprenderlo realmente quería entender el porque y al final todo resultó muy lógico... y entendido.

El análisis de toda la información y su modelado, resultó de gran ayuda para crear el prototipo del portal, fue como ir materializado en forma gráfica y amigable la información que un usuario de esta área necesita, y finalmente poder proyectar esto mismo a todas las áreas de la empresa”

VI.2.2 Discusión del segundo caso de estudio

Del cuestionario aplicado a la realizadora del segundo caso de estudio, podemos decir que el entendimiento de la metodología, así como su aplicación no resultaron de particular dificultad. Sin embargo, hemos observado que se requiere facilitar el paso del análisis hacia la definición de una solución concreta a los problemas obtenidos del estudio. Pasar del análisis de las fuentes y el conocimiento involucrado en el proceso, a la definición de la estructura de clasificación de dichas fuentes y conocimientos, así como a la definición de sus relaciones de forma que puedan concretarse en una solución técnica (por ejemplo un mapa de conocimientos para estructurar el portal, en este caso), no es del todo clara ni directa. Sin embargo, en un principio este no es el objetivo de la metodología. No obstante,

⁵ La estructura interna de la información que maneja el portal diseñado, está basada en un metamodelo de fuentes y tipos de conocimiento similar al propuesto en esta tesis (Capítulo III, sección III.4)

proponer métodos, técnicas, o herramientas que permitan facilitar este paso podría influir en el mejor uso de la metodología.

Finalmente, podemos destacar un punto que si bien no se observa en el cuestionario aplicado a la realizadora del caso de estudio, si se observó al momento de ver el diseño del producto resultante. Primero, queremos recordar que parte del objetivo que llevó a la definición de la metodología KoFI, fue el permitir integrar los sistemas existentes en un proceso, dentro de las estrategias de AC. En este segundo caso de estudio este objetivo no fue contemplado. Sin embargo, al analizar el portal de conocimientos propuesto, observamos que dentro de dicho portal se provee acceso a información que es generada o proporcionada por otros sistemas que actualmente utiliza la empresa. Es decir, que aun sin ser un objetivo perseguido en el estudio, el resultado final considera los sistemas existentes dentro de la empresa como parte de la solución de AC que se diseñó. Esto muestra que en este caso la metodología efectivamente cumplió con el objetivo para el que fue desarrollada, el cual es el perseguido en esta tesis. Es decir, apoyar en el diseño de sistemas o estrategias de AC que consideren el trabajo real de las personas, así como las herramientas que utilizan para realizar sus actividades diarias.

VI.3 Experimento

La elección de utilizar SPEM como apoyo en la fase de modelado, se debió al hecho de que ha sido diseñado para procesos de software y que está basado en UML, el lenguaje de modelado más ampliamente usado por las organizaciones de software [Bézivin y Breton, 2004]. No obstante lo anterior, varias dudas surgieron de esta elección, entre ellas: ¿es realmente SPEM fácil de entender por las personas dedicadas al desarrollo de software, principalmente aquellas con conocimientos en UML? Saber esto es importante debido a que los modelos desarrollados con el lenguaje propuesto deberán ser utilizados para analizar el proceso por parte de quienes lo realizan, por lo que emplear un lenguaje que les sea difícil de entender puede dificultar en gran medida esta tarea.

Otra duda fue si la inclusión de los nuevos elementos para representar el conocimiento y sus fuentes realmente podría apoyar en la identificación de los aspectos relacionados con el conocimiento que se querían identificar en el proceso. Particularmente, nos interesaba identificar si los modelos desarrollados con la nomenclatura propuesta pueden realmente apoyar en la identificación de lo siguiente:

- el conocimiento requerido y generado por las actividades;
- el conocimiento requerido por los participantes en el proceso modelado;
- las fuentes de información involucradas en las actividades;
- el conocimiento que puede ser obtenido en las fuentes participantes en el proceso;
- transferencias o flujos de conocimiento entre actividades y fuentes

Con el objeto de obtener información que nos ayudara a responder las dudas planteadas fue que se realizó un experimento que es detallado en el Apéndice A. En esta sección nos limitaremos a mencionar algunos de los resultados más relevantes del experimento.

VI.3.1 Principales resultados del experimento

El análisis detallado de los resultados del experimento se describe en el apéndice A. En esta sección sólo se presentan los resultados más relevantes.

VI.3.1.1 Las extensiones de SPEM permiten identificar el conocimiento involucrado en las actividades

El promedio general de calificación en las preguntas que median este punto indica que los participantes sí lograron identificar el conocimiento involucrado en el proceso. Sin embargo, existieron casos que mostraron algunas confusiones. Una confusión de los participantes se relacionó con identificar las fuentes de información y la información contenida en esa fuente. Sin embargo, las respuestas dadas por los participantes fueron en su mayoría correctas, lo que sugiere que realmente lograron identificar el conocimiento involucrado por medio de los diagramas presentados.

VI.3.1.2 Las extensiones de SPEM permiten identificar fuentes de conocimiento

En el caso de la identificación de fuentes de conocimiento, existió una confusión por parte de los participantes en el experimento con respecto a lo que es una fuente de información, y lo que es un paquete que agrupa el conocimiento que puede ser obtenido de una fuente determinada. Debido a esta confusión el promedio de calificación de algunas de las preguntas relacionadas con este tema fue bajo. Sin embargo, creemos que dejando bien claro lo que es una fuente de información, y lo que es un paquete que agrupa temas de conocimiento e información, el problema debe solucionarse.

Por otro lado, los participantes estuvieron de acuerdo en que los modelos presentados en el ejercicio ayudan a identificar las fuentes de información o conocimiento involucradas en el proceso (2 de acuerdo y 7 totalmente de acuerdo). Lo que sugiere que también este objetivo se está cumpliendo.

VI.3.1.3 Las extensiones de SPEM permiten identificar transferencias de conocimiento

Cuatro preguntas del experimento fueron utilizadas para evaluar si los participantes podían identificar transferencias de conocimiento en los diagramas presentados en el ejercicio. En dos de ellas se obtuvo un promedio de calificación mayor a 5 (la escala fue de 0 a 6), lo que sugiere que los participantes sí lograron identificar transferencias de conocimiento. Las otras dos preguntas arrojaron datos interesantes. En una de ellas se pidió identificar los flujos de conocimiento que se dan entre los roles en el proceso. En su mayoría, los sujetos proporcionaron sólo las dos representadas en los diagramas de transferencias de conocimiento que se incluyeron en los ejercicios. Por otro lado, en la segunda pregunta se les pidió que describieran una transferencia distinta a estas dos; dado que todos los participantes lograron identificar una, podemos decir que no sólo los diagramas de transferencia de conocimiento pueden ser útiles para identificar flujos de conocimiento entre fuentes. Incluso, varios participantes también consideraron los diagramas de paquetes, clases y actividad útiles para este propósito. Aun así, el

proporcionar diagramas específicos para representar transferencias de conocimiento explícitamente, puede reducir el esfuerzo de quienes analizan los modelos, de modo que no tenga que recurrir a analizar otros diagramas. Además, el no definir explícitamente estas transferencias puede provocar que algunas de ellas no sean tomadas en cuenta, como se observó en los resultados del experimento.

Finalmente, todos los participantes estuvieron de acuerdo en que los diagramas generados con la nomenclatura pueden ayudar a entender cómo fluye el conocimiento en un proceso (3 totalmente de acuerdo, 6 de acuerdo, y 1 ligeramente de acuerdo. Nadie en desacuerdo). Con base en todo lo anteriormente expuesto, consideramos que los mecanismos proporcionados para representar transferencias o flujos de conocimiento entre actividades y fuentes también cumplen su propósito, y permiten identificar información que de otra manera podría no ser tomada en cuenta.

VI.3.1.4 La nomenclatura es adecuada

Nueve de los participantes consideraron que los íconos usados en la nomenclatura son fáciles de asociar con los elementos que representan. También, hubo una mayor tendencia de los participantes a considerar que la nomenclatura y los tipos de diagramas empleados, son adecuados, además de claros, fáciles de entender y aprender. Sin embargo, como ya se ha mencionado, se presentaron algunas confusiones que es preciso tener en cuenta.

Durante el experimento se obtuvieron algunos comentarios de los participantes con respecto a la nomenclatura que hay que considerar. Primero, en los casos de uso se usaron dependencias para asociar el conocimiento generado y requerido por las actividades con los flujos de trabajo. En la nomenclatura de UML, las dependencias son flechas con líneas punteadas, cuya dirección va del elemento dependiente al elemento del que éste depende. Esto causó confusión entre los participantes, sobre todo en los diagramas de actividad, ya que SPEM define que las conexiones entre los objetos generados y usados en las actividades, son líneas semejantes a las usadas en las dependencias, sólo que en este caso la

dirección va de la actividad al objeto generado, o del objeto usado a la actividad donde se usa. Es decir, en el sentido inverso en el que este tipo de flechas son utilizadas en los casos de uso. Si bien la experiencia podría evitar este tipo de confusiones, con el fin de disminuirlas, se propone que las dependencias usadas en los casos de uso también indiquen el tipo de conexión, esto es, si se refieren a conocimiento usado, generado, o modificado. Así también, se podría usar otro tipo de conexión en los diagramas de actividad para hacer clara la diferencia de los dos tipos de relaciones: dependencias y asociación de productos del trabajo y actividades.

Finalmente, aun cuando el experimento fue realizado con un número de participantes muy pequeño, los datos muestran una tendencia que sugiere que las extensiones hechas a SPEM cumplen con los objetivos para los que fueron diseñadas. Sin embargo, existen algunos resultados que requieren de un estudio más amplio.

VI.3.1.5 La experiencia en UML puede facilitar el entendimiento de los modelos

Al analizar el resultado de los cuestionarios aplicados, no se observó una influencia clara entre la experiencia en UML que los participantes reportaron, y el promedio de calificación que obtuvieron. En contraste, la mayoría de los participantes (4 totalmente de acuerdo, 3 de acuerdo y 1 ligeramente de acuerdo) consideraron que el tener conocimientos de UML facilita el entendimiento de los modelos presentados en el experimento. Esto puede sugerir que existe una influencia entre el tener conocimientos de UML y la facilidad para entender los diagramas desarrollados con el lenguaje de modelado propuesto. Sin embargo, es necesario realizar más experimentos para ver si efectivamente esto es cierto.

VI.3.2 Conclusión del experimento

Del análisis de los resultados del experimento, podemos decir que las extensiones realizadas a SPEM cumplen su propósito. Sin embargo, es importante tener en consideración las confusiones y fallas que tuvieron los participantes con el fin de proponer mejoras a dichas extensiones. Por otro lado, el entendimiento de la nomenclatura empleada

no parece ser de particular dificultad para quienes participaron en el experimento. No obstante, creemos necesario hacer más estudios al respecto.

VI.4 Resumen del capítulo

Como se mencionó previamente en este capítulo, realizar una evaluación formal de la metodología como tal podría resultar sumamente difícil, debido a que su utilidad depende de factores externos difíciles de medir o controlar. Por lo tanto, la mejor forma de evaluar su utilidad depende de su uso en diversos casos con el fin de validar que apoya y cumple el objetivo para el que fue desarrollada. Debido al tiempo que requieren este tipo de estudios, para un trabajo con límite de tiempo como lo es una tesis doctoral, realizar varios casos de estudio en diversos escenarios resultaría inviable, es por esto que nos hemos limitado a realizar un solo caso de estudio para mostrar la manera en que la metodología puede apoyar en la definición de estrategias o sistemas de AC. El caso de estudio ayudó a mostrar el cumplimiento del objetivo principal para el cual la metodología fue definida. Por medio del caso de estudio presentado, se mostró la forma en que la metodología puede ayudar en la definición de estrategias de AC que consideren el trabajo realizado por las personas que participan dentro de un proceso, así como las herramientas que apoyan el proceso. De esta manera, dichas estrategias podrán diseñarse para mejorar el flujo del conocimiento dentro del proceso, sin la necesidad de involucrar grandes cambios en el mismo ni en los sistemas informáticos que lo apoyan.

Como un apoyo adicional a la validación de la utilidad de la metodología, se presentó un segundo caso de estudio donde la misma fue aplicada con el fin de diseñar un sistema de AC, particularmente un portal de conocimiento. Este caso de estudio inició con un objetivo distinto al que se realizó como parte de esta tesis. En el primer caso nos enfocamos a identificar la manera de proponer propuestas de mejora al flujo del conocimiento que consideraran las herramientas existentes dentro del grupo estudiado, mientras que en el segundo caso, se aplicó la metodología para proponer un sistema totalmente nuevo. No obstante, es importante mencionar que en este último caso, el sistema resultante también considera algunas de las herramientas que ya existían dentro del

proceso, aun cuando el considerarlas no fue un objetivo del estudio. El diseño del portal de conocimiento considera el acceso a diversos sistemas de información existentes dentro de la empresa, de tal forma que cierto tipo de información contenida en dichos sistemas pueda ser accedida desde el portal. Es decir, que aun sin ser un objetivo inicial, el uso de la metodología dio como resultado que en la propuesta de AC se consideraran las herramientas existentes dentro del proceso estudiado. Esto nos mostró que efectivamente la metodología ayuda a identificar el papel que juegan los sistemas de apoyo como facilitadores del flujo del conocimiento, con el fin de incluirlos en las propuestas de AC que se deriven de la aplicación de la misma en casos particulares.

Finalmente, como resultado del primer caso de estudio se identificaron algunos puntos que es importante tener en cuenta para mejorar y facilitar la aplicación tanto de la metodología como de las herramientas que la apoyan. Este tipo de observaciones dieron pie a la definición de posibles futuros trabajos derivados de la presente tesis, presentados en el siguiente capítulo junto con las principales aportaciones y conclusiones de este trabajo de investigación.

CAPÍTULO VII

Conclusiones, aportaciones y trabajo futuro

“El futuro tiene muchos nombres. Para los débiles es lo inalcanzable. Para los temerosos, lo desconocido. Para los valientes es la oportunidad.” Victor Hugo^{vi}

Este trabajo de tesis ha presentado el desarrollo de una metodología para apoyar a las organizaciones de software, e incluso organizaciones con otros giros, en el estudio de sus procesos con un enfoque en el flujo del conocimiento que se da en los mismos, con el objetivo de ayudar a dichas empresas en la definición de estrategias de AC que no impliquen grandes cambios en los procesos de trabajo, y que consideren la infraestructura tecnológica que apoya a dichos procesos. Lo anterior es de particular valía para las pequeñas y medianas empresas que quieran adoptar estrategias o sistemas de AC como apoyo para sus actividades, pero que difícilmente tendrán los recursos para desarrollar estrategias que requieran cambios grandes en sus procesos, o la adquisición o desarrollo de nuevos sistemas que podrían resultar sumamente costosos, y posiblemente no serían usados porque al ser genéricos no se ajustan al trabajo que realmente realizan los miembros de la organización.

En las siguientes secciones se describen las conclusiones de este trabajo, principales aportaciones obtenidas de la realización del mismo, y los principales puntos que han quedado pendientes como trabajo futuro.

VII.1 Conclusiones

Desarrollar sistemas de administración del conocimiento que resulten realmente útiles para las organizaciones no es una tarea fácil. Este trabajo de tesis inició como un esfuerzo por diseñar sistemas de AC para apoyar procesos de mantenimiento de software. Sin embargo, a lo largo del desarrollo del mismo nos dimos cuenta de que antes de diseñar

sistemas específicos era necesario entender el proceso desde un enfoque de AC, e identificar las necesidades de conocimiento reales del mismo. Así, nos dimos cuenta de que proponer soluciones de AC generales para el mantenimiento de software sería un reto sumamente complicado, si no imposible. Más bien, vimos que se requerían enfoques que ayudaran a identificar necesidades de conocimiento concretas de grupos específicos. Debido a la falta de un enfoque de este tipo, especialmente orientado a los procesos de software, fue que nos decidimos por desarrollar la metodología que se propuso en esta tesis.

Conforme se desarrollaban los distintos elementos de la metodología, observamos que un aspecto importante a tomar en cuenta eran las herramientas usadas por los actores de un proceso para realizar sus actividades. Los expertos en el área se han dado cuenta que los enfoques usados en el diseño de sistemas de AC con frecuencia fallan debido a que no consideran las necesidades reales de los trabajadores del conocimiento. Así mismo han observado que entre los principales factores para lograr el éxito de las estrategias de AC, sobre todo en empresas pequeñas y medianas, está el incluir las herramientas con que los empleados realizan sus actividades diarias. Debido a lo anterior, y basados en nuestra propia experiencia a lo largo del desarrollo de la metodología, decidimos incluir como un aspecto clave las herramientas usadas para apoyar las actividades de los procesos estudiados.

De lo anterior se derivó la metodología descrita en esta tesis, la cual presenta dos diferencias principales con las propuestas que hemos podido encontrar en la literatura. En primer lugar, nuestra metodología está enfocada más a ser un mecanismo para apoyar en el entendimiento del proceso desde un enfoque centrado en el flujo del conocimiento, que como un medio para desarrollar sistemas de AC específicos. Es decir, nuestro enfoque no se caza con un tipo particular de sistemas de AC, como es el caso de la mayoría de los que se encuentran en la literatura, sino que se centra en identificar las necesidades de AC de un proceso y con base en estas proponer las soluciones que puedan resultar más adecuadas. En segundo lugar, y quizá lo más relevante, es que nuestra propuesta toma como un factor muy importante las herramientas o sistemas usados como apoyo a las actividades del proceso

estudiado. De forma tal que las estrategias de AC puedan considerar dichos sistemas como parte de las mismas, e incluso como la base de ellas.

Con el fin de mostrar el uso y posibles beneficios de la metodología, se llevó a cabo un caso de estudio en un ambiente real, con el cual se pudo observar que la metodología tiene el potencial de cumplir con el objetivo para el que fue diseñada. Aun cuando para dar mayor fuerza a este argumento se requieren más casos en ambientes distintos, fue posible obtener datos de un segundo caso de estudio que refuerza lo observado en el primero. No obstante, consideramos conveniente seguir realizando más estudios de este tipo, como se describe en la sección de trabajo futuro más adelante.

Durante el proceso de desarrollo de la metodología, también se definieron técnicas para apoyar las distintas fases de la misma que representan aportes significativos. Por ejemplo aportaciones teóricas, como lo fue el caso del modelo general del flujo del conocimiento presentado en el Capítulo II; y prácticas, como lo es el metamodelo de fuentes y tipos de conocimientos que fue la base para las extensiones realizadas a SPEM, así como para la definición de mapas de conocimiento en ambos casos de estudio. Particularmente en el segundo caso de estudio, este metamodelo fue la base de la estructura interna del portal de conocimientos que fue diseñado. En la siguiente sección se detallan estas y otras aportaciones derivadas de esta tesis.

VII.2 Aportaciones

El principal objetivo de una tesis doctoral es la preparación del doctorando para que pueda desempeñarse como investigador independiente [Phillips y Pugh, 2001]. Esto implica obtener las habilidades y la experiencia que le permitan proponer y llevar a su culminación un proyecto de investigación, así como reportar y defender sus resultados ante la comunidad científica dentro de su área. El trabajo reportado en este documento de tesis, e incluso el desarrollo del documento mismo, contribuyen a dicha formación. Por otra parte, durante el proceso para la realización de este trabajo de tesis, se logró la participación en diversos congresos nacionales e internacionales, así como la publicación de artículos en

memorias de congresos, revistas e incluso un capítulo de un libro. Con base en estos resultados, podemos decir que la principal aportación de este trabajo de tesis fue la preparación del sustentante, dándole herramientas que le permitirán a futuro desempeñarse como investigador independiente.

Expuesto lo anterior, continuemos definiendo cuáles fueron las aportaciones concretas del tema de investigación abordado en esta tesis. Primero, la principal contribución del proyecto de investigación se centra en la identificación de los elementos que se requieren para apoyar en el estudio de procesos organizacionales con un enfoque hacia el flujo del conocimiento, así como la definición de una metodología desarrollada siguiendo como base dichos elementos. Otras aportaciones derivadas del desarrollo de ésta metodología fueron las dos propuestas de lenguaje de modelado de procesos, y un marco de trabajo para analizar sistemas de apoyo al flujo del conocimiento. Finalmente, las publicaciones derivadas de este trabajo constituyen una parte muy importante de las aportaciones que el mismo ha permitido obtener. A continuación se listan las principales aportaciones de este trabajo, seguidas de la lista de publicaciones desarrolladas con base en el mismo.

- 1) *Definición de los elementos a considerar durante el análisis de procesos de software con enfoque en el flujo del conocimiento que se da en dichos procesos.* Estos elementos se obtuvieron de un análisis extenso de literatura, tanto en el área de la administración del conocimiento, como de la ingeniería del software. Además, se estudiaron los problemas relacionados con el conocimiento que se dan en esta actividad. La identificación de estos elementos, presentada en el Capítulo II, fue la base para el desarrollo del resto de las propuestas de esta tesis.
- 2) *Un modelo que integra las distintas etapas de la administración del conocimiento como un ciclo continuo de tareas que permiten que el conocimiento fluya dentro de una organización.* Uno de los elementos principales dentro de cualquier estrategia y sistema de AC son las actividades que estos apoyan. Con el fin de determinar dichas actividades se hizo un análisis de propuestas existentes y se identificó la necesidad de ver estas tareas no como actividades aisladas, sino como un continuo que en su conjunto

permiten el flujo del conocimiento dentro de una organización. Con base en esto, en el Capítulo II se propuso un modelo integrado que muestra la forma en que las distintas actividades de la AC se relacionan para permitir el flujo del conocimiento en una organización.

- 3) *Una metodología para guiar el análisis de procesos organizacionales, especialmente procesos de software, desde el punto de vista del flujo del conocimiento en dichos procesos.* Con base en los elementos observados, se definió una metodología para ser aplicada como ayuda al estudio de procesos de software desde un enfoque del flujo del conocimiento que se da en dichos procesos, con el fin de identificar tanto ventajas como limitaciones que permitan definir estrategias para mejorar el flujo del conocimiento en dichos procesos. Esta metodología fue descrita en el Capítulo III y fue puesta en práctica en un caso de estudio presentado en el Capítulo VI.
- 4) *Un metamodelo de tipos y fuentes de conocimiento que puede ayudar a definir la estructura interna de un mapa de conocimiento para un proceso u organización.* Como parte de la metodología, se definió un metamodelo de fuentes y tipos de conocimiento que ha sido utilizado como base para la definición de la estructura general de mapas de conocimiento, que permitan desarrollar sistemas que hagan uso de dichos mapas para facilitar el acceso a fuentes de conocimiento. El metamodelo, presentado en el Capítulo III, fue la base para las extensiones realizadas a SPEM; uno de los lenguajes de modelado propuestos en el Capítulo IV. También, el metamodelo fue utilizado al principio de este trabajo de investigación para probar ideas y conceptos que dieron pie a varias de las publicaciones obtenidas en esta tesis. Así mismo, el metamodelo ayudó en la definición de la estructura interna del portal de conocimientos que se diseñó para el segundo caso de estudio descrito en el Capítulo VI.
- 5) *Dos adaptaciones a lenguajes de modelado de procesos para incluir explícitamente el conocimiento, sus fuentes y su flujo en los modelos.* Con el fin de apoyar la etapa del modelado de procesos de la metodología, se propusieron dos enfoques de modelado que consisten en la adaptación de dos lenguajes de modelado existentes, a los que se les agregaron elementos para representar de forma explícita el conocimiento, sus fuentes, y su flujo. La descripción de estas propuestas constituye la mayor parte del Capítulo IV.

- 6) *Un marco de trabajo que sirve de guía para el análisis de sistemas de AC, así como sistemas de información que podrían contribuir al flujo del conocimiento en procesos organizacionales.* Uno de los puntos importantes del objetivo de esta tesis, es considerar la infraestructura tecnológica que apoya a los procesos estudiados como parte de los sistemas o estrategias de AC que se propongan para mejora dichos procesos. Para esto, se desarrolló un marco de trabajo que sirve de guía para definir el rol que un sistema en particular puede jugar como habilitador del flujo del conocimiento dentro de un proceso u organización. Esta propuesta es descrita a detalle en el Capítulo V.
- 7) *Un conjunto de propuestas que pueden contribuir a mejorar el flujo del conocimiento dentro del Departamento de Informática de CICESE.* Como parte del caso de estudio realizado para mostrar la aplicación y utilidad de la metodología, se obtuvieron una serie de propuestas que pueden contribuir a mejorar el flujo del conocimiento dentro del proceso estudiado. El desarrollo de este caso de estudio es parte del Capítulo VI.

Finalmente, la principal aportación es la definición de todo un marco metodológico que apoya en la identificación, modelado y representación del conocimiento, sus fuentes y su flujo dentro de procesos de desarrollo de software, así como la identificación y análisis de las herramientas que intervienen en el flujo del conocimiento en dichos procesos. Todo esto se demostró que es útil para la definición de propuestas para el diseño de sistemas y estrategias para mejorar el flujo del conocimiento, por medio de dos casos de estudio, lo cual muestra que el objetivo general de esta tesis se ha cumplido.

VII.2.1 Publicaciones

Un resultado importante del presente trabajo de investigación, lo constituye un total de diecinueve publicaciones realizadas: tres de revista, dos capítulos de libro, y catorce artículos en memorias de congresos internacionales (tres de los cuales corresponden a la serie Lecture Notes in Computer Science). Así mismo se presentaron ponencias en ocho congresos internacionales, y dos en consorcios doctorales. A continuación se listan las referencias a los trabajos publicados.

Publicaciones en revistas

- R1.- Rodríguez-Elias, Oscar M., Ana I. Martínez-García, Aurora Vizcaíno, Jesús Favela, Mario Piattini, “Modeling and analysis of knowledge flows in software processes through the extension of the Software Process Engineering Metamodel”, *International Journal of Software Engineering and Knowledge Engineering*, World Scientific. (ISSN: 0218-1940) (**Aceptado para su publicación**) (Factor de impacto al 2006 de: 0.413).
- R2.- Rodríguez-Elias, Oscar M., Ana I. Martínez-García, Aurora Vizcaíno, Jesús Favela, Mario Piattini, (2007), “A Framework to Analyze Information Systems as Knowledge Flow Facilitators”, *Journal of Information and Software Technology*, Elsevier. (ISSN: 0950-5849). **Aceptado para su publicación**. (Disponible en línea desde Agosto de 2007, DOI: 10.1016/j.infsof.2007.07.002). (Factor de impacto al 2006 de: 0.726)
- R3.- Rodríguez-Elias, Oscar M., Aurora Vizcaíno Barceló, Ana I. Martínez-García, Jesús Favela Vara, Mario Piattini Velthuis, (2005), “Facilitando el acceso a las fuentes de información de un grupo de mantenimiento de software por medio de un mapa de conocimiento”, *Revista Gerencia Tecnológica Informática*, ITI Colombia, 4(10): 31-41, Diciembre 2005 (ISSN: 657-82364).

Capítulos de libro

- C1.- Cuestas, Pedro, Alma Gómez, González, Juan Carlos, et. al, (2008), "Agent Oriented Software Engineering", En: Moreno, Antonio y Juan Pavon (Eds.), *Issues in Multi-Agent Systems: The AgentCities.ES Experience*, Whitestein Series in Software Agent Technologies and Autonomic Computing, Birkhäuser, ISBN: 978-3-7643-8542-2. (Co-autor de la sección 7: "A Multi-Agent System to Manage Knowledge Sources in Software Maintenance Environments").
- C2.- Rodríguez-Elias, Oscar M., Ana I. Martínez-García, Aurora Vizcaíno, Jesús Favela, Mario Piattini, (2005), "Identifying Knowledge Flows in Communities of Practice", En: Coakes, Elayne, y Steve Cleark (Eds.), *Encyclopedia of Communities of Practice in Information and Knowledge Management*. IDEA Group Press, p. 210-217 (ISBN:

1-59140-556-4). También publicado en: Jennex, Murria E. (Ed.), Knowledge Management: Concepts, Methodologies, Tools, and Applications. IDEA Group Press, Vol. 2: p. 841-849, (2007). ISBN: 978-1-933-5.

Publicaciones en congresos

- M1.- Rodríguez-Elias, Oscar M., Ana I. Martínez-García, Aurora Vizcaíno, Jesús Favela, Mario Piattini (2007), “Knowledge flow analysis to identify knowledge needs for the design of knowledge management systems and strategies: a Methodological Approach”, Proceedings of the 9th International Conference on Enterprise Information Systems (ICEIS 2007), Funchal, Madeira – Portugal, del 12 al 16 de junio de 2007. pp. 492-497 (ISBN: 978-972-8865-89-4).
- M2.- Soto, Juan Pablo, Aurora Vizcaíno, Javier Portillo, Oscar M. Rodríguez-Elias, Mario Piattini, (2007), “Imitating the knowledge management of communities of practice”, Proc. of the fourth Intl. Conf. on Informatics in Control, Automation and Robotics (ICINCO 2007), Angers, Francia, 9-12 de Mayo de 2007. pp. 173-178.
- M3.- Rodríguez-Elias, Oscar M., Ana I. Martínez-García, Aurora Vizcaíno, Jesús Favela, Mario Piattini (2007), “Organización de conocimientos en procesos de Ingeniería de Software por medio del modelado de procesos: una adaptación de SPEM”, Proc. of the VI Jornada Iberoamericana de Ingeniería del Software e Ingeniería del Conocimiento (JIISIC’07), Lima, Perú, 31 de Enero al 2 de Febrero de 2007, p. 257-265 (ISBN: 978=9972-2885-1-7).
- M4.- Rodríguez-Elias, Oscar M., Ana I. Martínez-García, Jesús Favela, Aurora Vizcaíno, Mario Piattini (2006), “Knowledge Maps and Software Agents to Facilitate Knowledge Sources Identification in a Software Maintenance Group”, Memorias del 1er Congreso Internacional de Ciencias Computacionales, Ensenada, B.C., México, 6 al 8 de noviembre de 2006, p. 49-54 (ISBN: 970-735-051-2).
- M5.- Soto, Juan Pablo, Aurora Vizcaíno, Mario Piattini, Oscar M. Rodríguez, (2006) “A Multi-Agent Architecture to Develop Knowledge Management Systems”, Memorias del 1er Congreso Internacional de Ciencias Computacionales, Ensenada, B.C., México, 6 al 8 de noviembre de 2006, p. 37-42 (ISBN: 970-735-051-2).

- M6.- Rodríguez-Elias, Oscar M., Ana I. Martínez-García, (2006), “Hacia la definición de un esquema de clasificación de áreas de conocimiento para Ingeniería de Software en México”, En: Avances en la Ciencia de la Computación: Simposio de Ingeniería de Software, en el V Encuentro Mexicano de Ciencias de la Computación (ENC’ 2006), Arturo Hernández Aguirre y José Luis Zechineli Martini (Eds.), San Luís Potosí, México, 18-19 de Septiembre de 2006, p. 47-52 (ISBN: 968-5733-06-6).
- M7.- Soto, Juan Pablo, Oscar M. Rodríguez, Aurora Vizcaíno, Mario Piattini, Ana I. Martínez-García, (2006), “Localización de Fuentes del Conocimiento en el Proceso del Mantenimiento de Software”, En: Una Perspectiva de la Inteligencia Artificial en su 50 Aniversario: Memorias del Campus Multidisciplinar en Percepción e Inteligencia (CMPI-2006), vol. I, A. Fernández-Caballero, M. G. Manzano Arjona, E. A. González, y S. M. Tomé, Eds. Universidad de Castilla-La Mancha, Albacete, España, 2006, p. 118-123 (ISBN: 84-689-9561-4).
- M8.- Rodríguez-Elias, Oscar M., Ana I. Martínez-García, Aurora Vizcaíno, Jesús Favela, Mario Piattini, (2005), "Developing a Multi-Agent Knowledge Management System with INGENIAS", Memorias del II Taller de Desarrollo de Sistemas Multiagentes (DESMA 2005) at CEDI 2005, Granada, España, Sep. 13, González Moreno, Juan C. et al. (Eds.), Thomson, p. 37-43 (ISBN: 84-9732-454-4).
- M9.- Rodríguez-Elias, Oscar M., Ana I. Martínez-García, Aurora Vizcaíno, Jesús Favela, Mario Piattini, (2005), "Constructing a Knowledge Map for a Software Maintenance Organization", Poster Proceedings of the 21st IEEE International Conference on Software Maintenance (ICSM 2005), Budapest, Hungría, Sep. 26-29, p. 51-54 (ISBN: 963-460-981-3).
- M10.- Rodríguez, Oscar M., Aurora Vizcaíno, Ana I. Martínez, Mario Piattini, Jesús Favela, (2004), "Using a Multi-Agent Architecture to Manage Knowledge in the Software Maintenance Process", Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES 2004), Wellington, New Zealand. Sep. 22-24. **Lecture Notes in Artificial Intelligence**, Springer, Vol. 3213: 1181-1187 (ISSN: 0302-9743).

- M11.- Rodríguez, Oscar M., Ana I. Martínez, Jesús Favela, Aurora Vizcaíno, Mario Piattini, (2004), "Understanding and Supporting Knowledge Flows in a Community of Software Developers", Proceedings of the 10th International Workshop on Groupware (CRIWG'2004), San Carlos, Costa Rica, Oct.5-9. **Lecture Notes in Computer Science**, Springer, Vol. 3198: 52-66 (ISSN: 0302-9743).
- M12.- Rodríguez, Oscar M., Aurora Vizcaíno, Ana I. Martínez, Mario Piattini, Jesús Favela. (2004). "How to Manage Knowledge in the Software Maintenance Process", En: Advances in Learning Software Organizations: 6th International Workshop (LSO'2004), Banff, Alberta, Canada. June 20-21. **Lecture Notes in Computer Science**, Springer, Vol. 3096: 78-87 (ISSN: 0302-9743).
- M13.- Rodríguez, Oscar M., Ana I. Martínez, Aurora Vizcaíno, Jesús Favela, Mario Piattini, (2004), "Identifying Knowledge Management Needs in Software Maintenance Groups: A qualitative approach", Proceedings of the Fifth Mexican International Conference on Computer Science (ENC 2004), Colima, México. Sep. 20-24, IEEE Press, p. 72-79 (ISBN: 0-7695-2160-6).
- M14.- Rodríguez, Oscar M., Aurora Vizcaíno, Ana I. Martínez, Mario Piattini, Jesús Favela, (2004), "Applying Agents to Knowledge Management in Software Maintenance Organizations", Proceedings of the Workshop on Agent-Mediated Knowledge Management (AMKM 2004) at ECAI 2004, Valencia, España, Agosto 22-27, p. 39-45.

VII.3 Trabajo futuro

Durante el desarrollo de esta tesis quedaron algunos puntos sin concluir y surgieron distintas observaciones que no pudieron ser abordadas debido a cuestiones relacionadas con la delimitación temporal de la investigación. En particular, ha quedado pendiente la implementación y evaluación de las propuestas de mejoras al sistema de control de proyectos del departamento de informática de CICESE. Por otro lado, entre las observaciones surgidas se encuentran mejoras a las técnicas propuestas para apoyar la aplicación de la metodología, el desarrollo de herramientas que faciliten su aplicación, y más casos de estudio que permitan corroborar las ventajas y limitaciones. Estas

posibilidades se proponen como trabajo futuro, y algunas de ellas podrían por si solas constituir futuros trabajos de investigación. A continuación se detalla en qué consiste cada una de estas propuestas.

VII.3.1 Implementación y evaluación de los cambios al sistema de control de proyectos

Debido a la importancia del sistema de control de proyectos para el trabajo del departamento de informática, su modificación debe hacerse de manera cuidadosa, siguiendo un nuevo proceso de análisis del sistema actual. Lo anterior con el fin definir la forma de adaptarlo para tomar en cuenta las propuestas de mejora obtenidas del caso de estudio. Así mismo, una evaluación de los posibles beneficios del sistema una vez implementado podría ayudar a evaluar los resultados de la aplicación de la metodología KoFI. Por lo tanto, se propone como trabajo futuro la implementación de dicho sistema y su evaluación, no sólo a nivel conceptual, sino en la práctica.

VII.3.2 Seguimiento del segundo caso de estudio

Junto con la implementación y evaluación de los cambios propuestos al sistema de control de proyectos, otro factor que podría ayudar a evaluar la utilidad de la metodología en la práctica, es dar seguimiento al segundo caso de estudio. La empresa donde se desarrolló este segundo estudio tiene interés en desarrollar e implantar el portal de conocimiento propuesto como resultado del estudio. Por lo tanto, esto podría ser una buena oportunidad para medir la utilidad que las propuestas surgidas de la aplicación de la metodología pueden tener en la práctica.

VII.3.3 Mejoras en las adaptaciones a SPEM

Durante la aplicación de la metodología, y en especial las adaptaciones que se realizaron a SPEM, se observó la necesidad de extender el enfoque de modelado propuesto para representar de forma explícita aquellos sistemas que intervienen activamente como facilitadores del flujo del conocimiento. Es importante analizar adecuadamente la forma de incluir estos elementos dentro del metamodelo, así como definir íconos apropiados para los

mismos. Adicionalmente, se propone estudiar la pertinencia de hacer distinciones entre actividades comunes del proceso y actividades intensivas en conocimiento dentro de los modelos, dado que no todas las actividades requieren que se analicen a fondo en términos del conocimiento que se requiere para realizarlas, o del que se genera durante su realización.

Relacionado con este punto está el hecho de que existe una nueva versión del metamodelo SPEM, que se encuentra en desarrollo (versión beta con fecha del 8 de Julio de 2007). Es importante que dicha versión sea analizada con el fin de adaptar las extensiones propuestas en esta tesis a la nueva versión.

VII.3.4 Desarrollo de una herramienta de apoyo para la aplicación de la metodología

La aplicación de la metodología KoFI puede resultar un proceso laborioso y complicado si no se cuenta con una capacitación suficiente, sobre todo en lo relacionado a las restricciones para el modelado del proceso siguiendo la especificación de SPEM y las adaptaciones al mismo. Además, hacer descripciones de cada uno de los documentos que pueden existir dentro de una organización podría requerir mucho tiempo. Con el fin de facilitar esto, se propone el desarrollo de un sistema que apoye en el modelado, y que contenga elementos para ayudar en la automatización de ciertos elementos útiles durante el análisis, e incluso para el desarrollo de sistemas de AC para los procesos estudiados. A continuación se describen las propuestas en este sentido.

VII.3.4.1 Sistema de apoyo para el modelado del proceso

Primeramente, se propone el desarrollo, o adaptación en su defecto, de un sistema de apoyo para el modelado de procesos con base en SPEM y las adaptaciones realizadas. SPEM tiene la ventaja de haber sido desarrollado como un perfil de UML, por lo que para crear modelos con SPEM sólo hace falta contar con una aplicación que permita el modelado de UML y que apoye la inclusión de perfiles de UML. Esto hace que no sea estrictamente necesario desarrollar una herramienta completa para facilitar el desarrollo de

los modelos siguiendo el enfoque de modelado propuesto en esta tesis. Sin embargo, se recomienda realizar un estudio de las distintas herramientas que pudieran existir para este fin, y determinar la pertinencia de adaptar una de ellas, o desarrollar una nueva con un enfoque específicamente hacia el modelado de flujos de conocimiento con el lenguaje propuesto en esta tesis.

El contar con una herramienta de este tipo podría facilitar varias de las actividades que se requieren hacer durante la metodología. Por lo mismo, se recomienda que dicha herramienta se diseñe de forma tal que pueda ser extendida con módulos que faciliten otras actividades requeridas por la metodología KoFI. De esta manera, algunas de las actividades podrían ser automatizadas, o por lo menos semi-automatizadas. A continuación se describen las principales características que se proponen para ser incluidas dentro de estos módulos.

VII.3.4.2 Generación de perfiles de conocimiento

Un aspecto importante en toda organización es conocer los perfiles de conocimiento de sus empleados, y de los roles que éstos deben desempeñar. De esta manera podría facilitarse la asignación de recursos humanos para la realización de actividades específicas, así como la contratación de personal para cubrir dichos roles [Acuña *et al.*, 2006]. Debido a que el enfoque de modelado puede apoyar en la identificación del conocimiento que los roles del proceso requieren o aportan al proceso, se propone diseñar un sistema que apoye en la definición de perfiles de conocimiento. Tener un sistema para el manejo de estos perfiles podría permitir eventualmente utilizarlos para mejorar la identificación de fuentes de conocimiento útiles para quienes desempeñan las actividades de un proceso. Por ejemplo, un sistema de AC podría usar dichos perfiles para organizar las fuentes de conocimiento existentes y asignar mayor relevancia a aquellas fuentes que contengan conocimientos que no estén incluidos en el perfil del usuario. Además, teniendo una herramienta para apoyar el modelado del proceso siguiendo el enfoque propuesto en esta tesis, esta herramienta podría integrarse con el sistema de manejo de perfiles para que éstos últimos se generen automáticamente con base en los modelos del proceso.

VII.3.4.3 Automatización de las descripciones de tipos y fuentes de conocimiento

En el caso de estudio se observó que gran parte de la información contenida en los modelos del proceso se corresponde con la requerida por las plantillas para la descripción y clasificación de tipos y fuentes de conocimiento. Por lo tanto, teniendo un sistema que apoye el modelado con base en el lenguaje propuesto, se podría incluir un módulo para generar automáticamente las descripciones de tipos y fuentes de conocimiento, facilitando estas tareas durante la fase de análisis de la metodología KoFI.

VII.3.4.4 Automatización de la construcción de mapas de conocimiento

Otra de las observaciones obtenidas en los dos caso de estudio descritos, es que el metamodelo con el cual se definieron las extensiones a SPEM, puede servir de base para la estructura interna de un mapa de conocimientos que permita relacionar las distintas fuentes de conocimiento con los conocimientos que contienen y las actividades donde son generadas o utilizadas. Por lo tanto, diseñar un sistema que permita construir mapas de conocimiento de forma automática tomando la información proporcionada en los modelos podría ayudar en el desarrollo de sistemas de apoyo a la AC que utilicen dichos mapas para prestar servicios a sus usuarios; por ejemplo identificando necesidades de conocimiento con base en las actividades realizadas por los usuarios, y detectando fuentes de conocimiento útiles para dichas actividades. Cabe destacar que posibles aplicaciones de este tipo de mapas ya fueron exploradas al inicio de este trabajo de tesis, y fueron publicados en diversos congresos y en una revista [Rodríguez-Elias *et al.*, 2005a; Rodríguez-Elias *et al.*, 2005b; Rodríguez *et al.*, 2004a].

VII.3.5 Realización de más casos de estudio

Finalmente, con el objetivo de evaluar la aplicación de la metodología para detectar sus ventajas y limitaciones, se propone la realización de más casos de estudio en contextos distintos. De esta manera se podría determinar que tan factible es que la metodología apoye no sólo en procesos de software, sino en otros tipos de procesos organizacionales. En este

sentido, el segundo caso de estudio es un avance en esta línea. Adicionalmente, existe el interés de un estudiante de la Maestría en Tecnologías de la Información y la Comunicación de la UABC, por aplicar la metodología KoFI para desarrollar un sistema de administración del conocimiento para la dirección de información académica de la UABC, en el campus Ensenada. De lograrse este proyecto, contribuiría en la evaluación de la utilidad de la metodología en contextos distintos al desarrollo de software, y junto con el segundo caso descrito en esta tesis, refuerza la necesidad de un enfoque metodológico que apoye en el estudio de procesos organizacionales con el fin de desarrollar sistemas o estrategias de AC para mejorar dichos procesos.

^{vi} Victor Hugo (1802-1885) fue un novelista francés del siglo XIX, quizá su obra más famosa sea “Los Miserables”.

Referencias

- Abdullah, M. S., Benest, I., Evans, A., y Kimble, C. 2002. Knowledge modelling techniques for developing knowledge management systems. En: memorias de European Conference on Knowledge Management, Dublin, Ireland. Septiembre 2002. 15-25 p.
- Abran, A., Moore, J. W., Bourque, P., Dupuis, R., y Tripp, L. L. (Eds.). 2004. Swebok: Guide to the software engineering body of knowledge: 2004 version. IEEE Computer Society. Los Alamitos, California. 98 p.
- Acuña, S. T., Juristo, N., y Moreno, A. M. 2006. Emphasizing human capabilities in software development. *IEEE Software*, 23(2): 94-101 p.
- Alavi, M., y Leidner, D. E. 2001. Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly*, 25(1): 107-136 p.
- Alavi, M., y Tiwana, A. 2002. Knowledge integration in virtual teams: The potential role of kms. *Journal of the American Society for Information Science and Technology*, 53(12): 1029-1037 p.
- Althoff, K.-D., Becker-Kornstaedt, U., Decker, B., Klotz, A., Leopold, E., Rech, J., y Voß, A. 2002. Enhancing experience management and process learning with moderated discourses: The indigo approach. *Lecture Notes in Computer Science*, 2569: 114-125 p.
- Althoff, K.-D., Birk, A., Hartkopf, S., Müller, W., Nick, M., Surmann, D., y Tautz, C. 1999. Managing software engineering experience for comprehensive reuse. En: memorias de Eleventh International Conference on Software Engineering and Knowledge Engineering, Kaiserslautern, Germany. 17-19 June 1999. 10-19 p.
- Anquetil, N., de Oliveira, K. M., de Sousa, K. D., y Batista Dias, M. G. 2007. Software maintenance seen as a knowledge management issue. *Information and Software Technology*, 49(5): 515-529 p.
- Aurum, A., Daneshgar, F., y Ward, J. 2007. Investigating knowledge management practices in software development organizations - an Australian experience. *Information and Software Technology*. DOI: 10.1016/j.infsoft.2007.05.005. Mayo, 2007.
- Aurum, A., Jeffery, R., Wohlin, C., y Handzic, M. (Eds.). 2003. Managing software engineering knowledge. Springer. Berlin, Germany. 406 p.

- Avison, D., Lau, F., Myers, M., y Nielsen, P. A. 1999. Action research. *Communications of the ACM*, 42(1): 94-97 p.
- Banerjee, R. 2005. A fool with a tool is still a fool. En: M. Rao (Ed.). *Knowledge management tools and techniques*. Elsevier, Amsterdam, 283-292 p.
- Basili, V. R., y Caldiera, G. 1995. Improve software quality by reusing knowledge & experience. *Sloan Management Review*, 37(1): 55-64 p.
- Basili, V. R., Caldiera, G., y Rombach, H. D. 1994. The experience factory En: J. J. Marciniak (Ed.). *Encyclopedia of software engineering*. John Willey & Sons, 469-476 p.
- Beavers, G., y Hexmoor, H. 2003. Types and limits of agent autonomy. En: *memorias de Agents and Computational Autonomy at AAMAS 2003*, Melbourne, Australia. Julio 14, 2007. 95-102 p.
- Bera, P., Nevo, D., y Wand, Y. 2005. Unravelling knowledge requirements through business process analysis. *Communications of the Association for Information Systems*, 16: 814-830 p.
- Bézivin, J., y Breton, E. 2004. Applying the basic principles of model engineering to the field of process engineering. *UPGRADE*, V(5): 27-33 p.
- Binney, D. 2001. The knowledge management spectrum -understanding the km landscape. *Journal of Knowledge Management*, 5(1): 33-42 p.
- Bleistein, S. J., Cox, K., Verner, J., y Phalp, K. T. 2006. B-scp: A requirements analysis framework for validating strategic alignment of organizational it based on strategy, context, and process. *Information and Software Technology*, 48(9): 846-868 p.
- Bonifacio, M., Bouquet, P., y Cuel, R. 2002a. Knowledge nodes: The building blocks of a distributed approach to knowledge management. *Journal of Universal Computer Science*, 8(6): 652-661 p.
- Bonifacio, M., Bouquet, P., y Traverso, P. 2002b. Enabling distributed knowledge management: Managerial and technological implications. *Novatica and Informatik/Informatique*, 3(1): 23-29 p.
- Bontis, N., Fearon, M., y Hishon, M. 2003. The e-flow audit: And evaluation of knowledge flow within and outside a high tech firm. *Journal of Knowledge Management*, 7(1): 6-19 p.

- Borghoff, U. M., y Pareschi, R. 1997. Information technology for knowledge management. *Journal of Universal Computer Science*, 3(8): 835-842 p.
- Borghoff, U. M., y Pareschi, R. (Eds.). 1998. Information technology for knowledge management. Springer. Berlin, Germany. 244 p.
- Brainov, S., y Hexmoor, H. 2001. Quantifying relative autonomy in multiagent interaction. En: memorias de IJCAI-01, Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents, Seattle, WA. 27-35 p.
- Carabelea, C., Boissier, O., y Florea, A. 2003. Autonomy in multi-agent systems: A classification attempt. En: memorias de Agents and Computational Autonomy at AAMAS 2003, Melbourne, Australia. Julio 14, 2003. 103-113 p.
- Carlsson, S. A. (2002). Towards an understanding and conceptualization of knowledge managing within the context of inter-organizational networks, Third European Conference on Organizational Knowledge, Learning, and Capabilities (OKLC'2002). Atenas, Grecia. 5 y 6 de Abril de 2002.
- Carroll, J. M., y Rosson, M. B. 1992. Getting around the task-artifact cycle: How to make claims and design by scenario. *ACM Transactions on Information Systems*, 10(2): 181-212 p.
- Chapin, N. 1987. The job of software maintenance. En: memorias de Conference on Software Maintenance-1987. 4-12 p.
- Chapin, N. 2003. Software maintenance and organizational health and fitness En: M. Polo, M. Piattini & F. Ruiz (Eds.). *Advances in software maintenance management: Technologies and solutions*. Idea Group Inc., Hershey, PA. USA, 1-31 p.
- Chapin, N., Hale, J. E., Khan, K. M., Ramil, J. F., y Tan, W.-G. 2001. Types of software evolution and software maintenance. *Journal of Software Maintenance and Evolution: Research and Practice*, 13(1): 3-30 p.
- Checkland, P. 1972. Towards a systems-based methodology for real-world problem solving. *Journal of Systems Engineering*, 3(2): 87-116 p.
- Checkland, P., y Scholes, J. 1999. *Soft system methodology in action*. John Wiley and Sons. 418 p.

- Chin, G. J., Rosson, M. B., y Carroll, J. M. 1997. Participatory analysis: Shared development of requirements from scenarios. En: memorias de Conference on Human Factors in Computing Systems (CHI97), Atlanta, GA, USA. 162-169 p.
- Choo, C. W. 1999. The knowing organization: How organizations use information to construct meaning, create knowledge, and made decisions. Oxford University Press. Oxford, USA. 346 p.
- Conradi, R., y Jaccheri, L. 1999. Process modelling languages En: J. C. Derniame, B. A. Kaba & D. Wastell (Eds.). Software process. Springer, Berlin, 27-52 p.
- Cox, K., Hall, J. G., y Rapanotti, L. 2005a. A roadmap of problem frames research. Information and Software Technology, 46(14): 891-902 p.
- Cox, K., Phalp, K. T., Bleistein, S. J., y Verner, J. M. 2005b. Deriving requirements from process models via the problem frames approach. Information and Software Technology, 47(5): 319-337 p.
- Cuel, R. 2003. A new methodology for distributed knowledge management analysis. En: memorias de International Symposium on Knowledge Management (I-KNOW '03), Graz, Austria. 2-4 Julio. 531-537 p.
- Curia, R., Gallucci, L., y Ruffolo, M. 2005. Knowledge management in health care: An architectural framework for clinical process management systems. En: memorias de 16th International Workshop on Database and Expert Systems Applications (DEXA'05), Copenhagen, Denmark. 22-26 Agosto de 2005. 393-397 p.
- Curtis, B., Kellner, M. I., y Over, J. 1992. Process modeling. Communications of the ACM, 35(4): 75-90 p.
- Curtis, B., Krasner, H., y Iscoe, N. 1988. A field study of the software design process for large systems. Communications of the ACM, 31(11): 1268-1287 p.
- Dalkir, K. 2005. Knowledge management in theory and practice. Elsevier. Amsterdam. 356 p.
- Dart, S., Christie, A. M., y Brown, A. W. 1993. A case study in software maintenance (Technical Report No. CMU/SEI-93-TR-8, ESC-TR-93-185). Pittsburgh: Software Engineering Institute, Carnegie Mellon University.

- Davenport, T. H. 2007. Information technologies for knowledge management En: K. Ichijo & I. Nonaka (Eds.). Knowledge creation and management: New challenges for managers. Oxford University Press, New York, NY., 97-117 p.
- Davenport, T. H., De Long, D. W., y Beers, M. C. 1998. Successful knowledge management projects. Sloan Management Review, 39(2): 43-57 p.
- Davenport, T. H., y Prusak, L. 1997. Ecología de la información. Oxford University Press. New York. 297 p.
- Davenport, T. H., y Prusak, L. 2000. Working knowledge: How organizations manage what they know. Harvard Business School Press. Boston, Massachusetts. 199 p.
- Davis, F. D. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly, 13(3): 319-340 p.
- de Sousa, K. D., Anquetil, N., y Oliveira, K. M. d. 2004. Learning software maintenance organizations. En: memorias de 6th International Workshop on Learning Software Organization (LSO 2004), Banff, Canada. 20-21 Junio. 67-77 p.
- Desouza, K. C. 2003. Barriers to effective use of knowledge management systems in software engineering. Communications of the ACM, 46(1): 99-101 p.
- Ding, Y., y Foo, S. 2002. Ontology research and development, part 1 - a review of ontology generation. Journal of Information Science, 28(2): 123-136 p.
- Dingsøyr, T., y Conradi, R. 2002. A survey of case studies of the use of knowledge management in software engineering. International Journal of Software Engineering and Knowledge Engineering, 12(4): 391-414 p.
- Edwards, J. S. 2003. Managing software engineers and their knowledge En: A. Aurum, R. Jeffery, C. Wohlin & M. Handzic (Eds.). Managing software engineering knowledge. Springer, Berlin, 5-27 p.
- Fensel, D. 2001. Ontologies: A silver bullet for knowledge management and electronic commerce. Springer. Berlin. 138 p.
- Folkens, F., y Spiliopoulou, M. 2004. Towards an evaluation framework for knowledge management systems. En: memorias de Practical Aspects of Knowledge Management (PAKM). 23-34 p.

- Franklin, S., y Graesser, A. 1996. Is it an agent, or just a program? A taxonomy for autonomous agents. En: memorias de Workshop on Intelligent Agents III, Agent Theories, Architectures, and Languages. Agosto ,12-13. Budapest, Hungría. 21-35 p.
- García, S., Graettinger, C., y Kost, K. 2006. Proceedings of the first international workshop for process improvement in small settings, 2005 (Special Report No. CMU/SEI-2006-SR-001). Pittsburgh, PA: Carnegie Mellon, Software Engineering Institute. 309 p.
- Gilbert, X. 2007. Globalizing local knowledge in global companies En: K. Ichijo & I. Nonaka (Eds.). Knowledge creation and management: New challenges for managers. Oxford University Press, New York, NY., 215-228 p.
- Glass, R. L. 2001. Frequently forgotten fundamental facts about software engineering. IEEE Software, 18(3): 112-111 p.
- Google. Búsquedas por directorio de google. Accedido en 10 de Agosto, 2007, desde <http://www.google.com.mx/dirhp?hl=es>, Agosto 2007.
- Grant, R. M. 1996. Toward a knowledge-based theory of the firm. Strategic Management Journal, 17: 109-122 p.
- Grehan, R. 2005. Managing your content with xml. InfoWorld, 27(28): 16-18 p.
- Gruber, T. R. 1995. Toward principles for the design of ontologies used for knowledge sharing. International Journal on Human-Computer Studies, 43(5/6): 907-928 p.
- Haggie, K., y Kingston, J. 2003. Choosing your knowledge management strategy. Electronic Journal of Knowledge Management Practice, 4(5): 1-5 p.
- Hansen, B. H., y Kautz, K. 2004. Knowledge mapping: A technique for identifying knowledge flows in software organizations. En: memorias de European Conference on Software Process Improvement (EuroSPI 2004), Trondheim, Norway. Noviembre 10-12. 126-137 p.
- Hexmoor, H. 2000. A cognitive model of situated autonomy. En: memorias de PRICAI 2000 Workshop Reader, Four Workshops held at PRICAI 2000; Advances in Artificial Intelligence, Melbourn, Australia. Agosto 28 - Septiembre 1, 2000. 325-334 p.
- Hilburn, T. B., Hirmanpour, I., Khajenoori, S., Turner, R., y Qasem, A. 1999a. A software engineering body of knowledge version 1.0 (No. CMU/SEI-99-TR-004, ESC-TR-99-004). Pitsburg, PA: Software Engineering Institute, Carnegie Mellon.

- Hilburn, T. B., Hirmanpour, I., Khajenoori, S., Turner, R., y Qasem, A. 1999b. A software engineering body of knowledge version 1.0 (No. CMU/SEI-99-TR-004, ESC-TR-99-004). Pitsburg, PA: Software Engineering Institute, Carnegie Mellon.
- Holsapple, C. W., y Joshi, K. D. 1999. Description and analysis of existing knowledge management frameworks. En: memorias de 32nd Hawaii Int. Conf. on Systems Sciences (HICSS), Maui, Hawaii. Enero 5-8.
- Ichijo, K., y Nonaka, I. (Eds.). 2007. Knowledge creation and management: New challenges for managers. Oxford University Press. New York, NY. 335 p.
- IEEE. 1995. Std 1074-1995: Ieee standard for developing software life cycle processes.
- ISO/IEC. 1999. Iso/iec fdis 14764:1999, software engineering - software maintenance (Standard): Secretariat: Standard Council of Canada.
- Jackson, M. 2005. Problem frames and software engineering. *Information and Software Technology*, 47(14): 903-912 p.
- Jacobson, I., Booch, G., y Rumbaugh, J. 2000. El proceso unificado de desarrollo de software. Addison Wesley. Madrid. 464 p.
- Jennex, M. E., y Olfman, L. 2005. Assessing knowledge management success. *International Journal of Knowledge Management*, 1(2): 33-49 p.
- Kankanhalli, A., y Tan, B. C. Y. 2004. A review of metrics for knowledge management systems and knowledge management initiatives. En: memorias de 37th Hawaii International Conference on Systems Science (HICSS), Big Island, Hawaii. Enero 5-8.
- Kankanhalli, A., Tanudidjaja, F., Sutanto, J., y Tan, B. C. Y. 2003. The role of it in successful knowledge management initiatives. *Communications of the ACM*, 46(9): 69-73 p.
- Kim, S., Hwang, H., y Suh, E. 2003. A process-based approach to knowledge flow analysis: A case study of a manufacturing firm. *Knowledge and Process Management*, 10(4): 260-276 p.
- Kitchenham, B., y Pickard, L. 1995. Case studies for method and tool evaluation. *IEEE Software*, 12(4): 52-62 p.
- Klint, P., y Verhoef, C. 2002. Enabling the creation of knowledge about software assets. *Data & Knowledge Engineering*, 41(2-3): 141-158 p.

- Komi-Sirviö, S., Mäntyniemi, A., y Seppänen, V. 2002. Toward a practical solution for capturing knowledge for software projects. *IEEE Software*, 19(3): 60-62 p.
- Kucza, T., y Komi-Sirviö, S. 2001. Utilising knowledge management in software process improvement - the creation of a knowledge management process model. En: *memorias de 7th international Conference on Concurrent Enterprising (ICE 2001)*, Nottingham, UK. 241-249 p.
- Kucza, T., Nättinen, M., y Parviainen, P. 2001. Improving knowledge management in software reuse process. En: *memorias de International Conference on Product Focused Software Process Improvement*, Kaiserslautern, Germany. Septiembre 10-13. 141-152 p.
- Kühn, O., y Abecker, A. 1998. Corporate memories for knowledge management in industrial practice: Prospects and challenges En: U. M. Borghoff & R. Pareschi (Eds.). *Information technology for knowledge management*. Springer, Berlin, 183-206 p.
- Lethbridge, T. C. 2000. What knowledge is important to a software professional? *IEEE Computer*, 33(5): 44-50 p.
- Lethbridge, T. C., Singer, J., y Forward, A. 2003. How software engineers use documentation: The state of the practice. *IEEE Software*, 20(6): 35- 39 p.
- Lientz, B. P. 1983. Issues in software maintenance. *Computing Surveys*, 15(3): 271-278 p.
- Lindvall, M., y Rus, I. 2003. Knowledge management for software organizations En: A. Aurum, R. Jeffery, C. Wohlin & M. Handzic (Eds.). *Managing software engineering knowledge*. Springer, Berlin, 73-94 p.
- Lutters, W. G., y Seaman, C. B. 2007. Revealing actual documentation usage in software maintenance through war stories. *Information and Software Technology*, 49(6): 576-587 p. DOI: 10.1016/j.infsof.2007.02.013.
- Maedche, A., Motik, B., Stojanovic, L., Studer, R., y Volz, R. 2003. Ontologies for enterprise knowledge management. *IEEE Intelligent Systems*, 18(2): 26-33 p.
- Maes, P. 1994. Agents that reduce work and information overload. *Communications of the ACM*, 37(7): 31-40 p.
- Maier, R., y Remus, U. 2002. Defining process-oriented knowledge management strategies. *Knowledge and Process Management*, 9(2): 103-118 p.

- Marwick, A. D. 2001. Knowledge management technology. *IBM Systems Journal*, 40(4): 814-830 p.
- Mayrhauser, A. v., y Vans, A. M. 1995. Program comprehension during software maintenance and evolution. *IEEE Computer*, 28(8): 44-55 p.
- McElroy, M. W. 2000. The new knowledge management. *Knowledge and Innovation: Journal of the KMCI*, 1(1): 43-67 p.
- Meehan, B., y Richardson, R. 2002. Identification of software process knowledge management. *Software Process Improvement and Practice*, 7: 47-55 p.
- Meyer, B. 2006. Testable, reusable units of cognition. *IEEE Computer*, 39(4): 20-24 p.
- Monk, A., y Howard, S. 1998. The rich picture: A tool for reasoning about work context. *Interactions*, 5(2): 21-30 p.
- Munson, J. 1998. Software lives too long. *IEEE Software*, 15(4): 18-20 p.
- Newman, B. B., y Conrad, K. W. 2000. A framework for characterizing knowledge management methods, practice, and technologies. En: *memorias de Practical Aspects of Knowledge Management (PAKM 2000)*, Basel, Switzerland. 30-31 Oct.
- Nissen, M. E. 2002. An extended model of knowledge-flow dynamics. *Communications of the Association for Information Systems*, 8: 251-266 p.
- Nissen, M. E., y Levitt, R. E. 2002. Dynamic models of knowledge-flow dynamics (Working paper No. 76). Stanford, CA, USA: Center for Integrated Facility Engineering (CIFE), Stanford University.
- Nissen, M. E., y Levitt, R. E. 2004. Agent-based modeling of knowledge flows: Illustration from the domain of information systems design. En: *memorias de Hawaii International Conference on System Science (HICSS 2004)*, Big Island, Hi, USA.
- Nonaka, I. 1991. The knowledge-creating company. *Harvard Business Review on Knowledge Management*: 26-46 p.
- Nonaka, I., y Konno, N. 1998. The concept of "ba": Building a foundation for knowledge creation. *California Management Review*, 40(3): 40-54 p.
- Nonaka, I., y Takeuchi, H. 1995. *The knowledge-creation company: How japanese companies create the dynamics of innovation*. Oxford University Press. New York, NY. 304 p.

- O'Leary, D. E. 1998. Using ai in knowledge management: Knowledge bases and ontologies. *IEEE Intelligent Systems*, 13(3): 34-39 p.
- Oliveira, K. M., Anquetil, N., M.G, D., Ramal, M., y Meneses, R. 2003. Knowledge for software maintenance. En: *memorias de Fifteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'03)*, San Francisco, CA. 61-68 p.
- OMG. 2002. Software process engineering metamodel specification (spem). Accedido en Octubre 29, 2004, desde <http://www.omg.org/technology/documents/formal/spem.htm>
- Papavassiliou, G., y Mentzas, G. 2003. Knowledge modelling in weakly-structured business processes. *Journal of Knowledge Management*, `7(2): 18-33 p. DOI: DOI 10.1108/13673270310477261.
- Peachey, T. 2005. Knowledge management and the leading is journals: An analysis of trends and gaps in published research. En: *memorias de 38th Hawaii International Conference on System Science (HICSS)*, Big Island, HI, USA. 3-6 Enero 2005.
- Phillips, E. M., y Pugh, D. S. 2001. *Cómo obtener un doctorado: Manual para estudiantes y tutores*. Gedisa editorial. Barcelona. 271 p.
- Polo, M., Piattini, M., Ruiz, F., y Calero, C. 1999. Roles in the maintenance process. *Software Engineering Notes, Special Interest Group on Software Engineering, ACM*, 24, N°4: 84-86 p.
- Polo, M., Piattini, M., y Ruiz, F. 2003a. A methodology for software maintenance En: M. Polo, M. Piattini & F. Ruiz (Eds.). *Advances in software maintenance management: Technologies and solutions*. Idea Group Inc., Hershey, 228-254 p.
- Polo, M., Piattini, M., y Ruiz, F. (Eds.). 2003b. *Advances in software maintenance management: Technologies and solutions*. Idea Group Inc. Hershey, PA, USA. 310 p.
- Qureshi, S., Hlupic, V., y Briggs, R. O. 2004. On the convergence of knowledge management and groupware. En: *memorias de 10th International Workshop on Groupware (CRIWG'2004)*, San Carlos, Costa Rica. 25-33 p.
- Rajlich, V. 2006. Changing the paradigm of software engineering. *Communications of the ACM*, 49(8): 67-70 p.
- Rao, M. 2005a. Overview: The social life of km tools En: M. Rao (Ed.). *Knowledge management tools and techniques*. Elsevier, Amsterdam, 1-73 p.

- Rao, M. (Ed.). 2005b. Knowledge management tools and techniques: Practitioners and experts evaluate km solutions. Elsevier. Amsterdam. 438 p.
- Rhem, A. J. 2006. Uml for developing knowledge management systems. Auerbach Publications. Boca Raton, FL. 269 p.
- Richardson, I., y von Wangenheim, C. G. 2007. Why are small software organizations different? *IEEE Software*, 24(1): 18-22 p.
- Robillard, P. N. 1999. The role of knowledge in software development. *Communications of the ACM*, 42(1): 87-92 p.
- Rodriguez-Elias, O. M., Martínez-García, A. I., Vizcaíno, A., Favela, J., y Piattini, M. 2005a. Constructing a knowledge map for a software maintenance organization. En: memorias de Poster Session of the 21st IEEE International Conference on Software Maintenance (ICSM 2005), Budapest, Hungary. Septiembre 26-29. 51-54 p.
- Rodriguez-Elias, O. M., Martínez García, A. I., Vizcaíno Barceló, A., Favela Vara, J., y Piattini Velthuis, M. 2005b. Facilitando el acceso a las fuentes de información de un grupo de mantenimiento de software por medio de un mapa de conocimiento. *Gerencia Tecnológica Informática*, 4(10): 31-41 p.
- Rodríguez Elias, O. M. 2003. Administración del conocimiento como soporte al proceso de mantenimiento de software. Tesis de Maestría. CICESE. Ensenada, B.C., México.
- Rodríguez, O. M., Martínez, A. I., Favela, J., Vizcaíno, A., y Piattini, M. 2004a. Understanding and supporting knowledge flows in a community of software developers. *Lecture Notes in Computer Science*, 3198: 52-66 p.
- Rodríguez, O. M., Martínez, A. I., Vizcaíno, A., Favela, J., y Piattini, M. 2004b. Identifying knowledge management needs in software maintenance groups: A qualitative approach. En: memorias de Fifth Mexican International Conference on Computer Science (ENC'2004), Colima, México. 72-79 p.
- Rus, I., y Lindvall, M. 2002. Knowledge management in software engineering. *IEEE Software*, 19(3): 26-38 p.
- Rus, I., Lindvall, M., y Sinha, S. S. 2001. Knowledge management in software engineering: A state of the art report. Rome, NY: Data & Analysis Center for Software: ITT Industries.

- Schneider, K., von Hunnius, J.-P., y Basili, V. R. 2002. Experience in implementing a learning software organization. *IEEE Software*, 19(3): 46-49 p.
- Scholl, W., König, C., Meyer, B., y Heisig, P. 2004. The future of knowledge management: An international delphi study. *Journal of Knowledge Management*, 8(2): 19-35 p.
- Schreiber, G., Wielinga, B., de Hoog, R., Akkermans, H., y de Velde, W. V. 1994. Commonkads: A comprehensive methodology for kbs development. *IEEE Intelligent Systems*, 9(6): 28-37 p.
- Seaman, C. 2002. The information gathering strategies of software maintainers. En: *memorias de International Conference on Software Maintenance (ICSM'2002)*, Montreal, Canada. Octubre 2002. 141-149 p.
- Seaman, C. B., Mendonca, M., Basili, V. R., y Kim, Y.-M. 2003. User interface evaluation and empirically-based evolution of a prototype experience management tool. *IEEE Transactions on Software Engineering*, 29(9): 838-850 p.
- Serenko, A., y Bontis, N. 2004. Meta-review of knowledge management and intellectual capital literature: Citation impact and research productivity rankings. *Knowledge and Process Management*, 11(3): 185-198 p.
- Singer, J. 1998. Practices of software maintenance. En: *memorias de International Conference on Software Maintenance*, Bethesda, Maryland, USA. 16-19 Noviembre. 139-145 p.
- Smith, H. A., y McKeen, J. D. 2004. Knowledge-enabling business processes. *Communications of the Association for Information Systems*, 13: 25-38 p.
- Smith, H. A., y McKeen, J. D. 2005. Development in practice xvii: A framework for km evaluation. *Communications of the Association for Information Systems*, 16: 233-246 p.
- Sparrow, J. 2001. Knowledge management in small firms. *Knowledge and Process Management*, 8(1): 3-16 p.
- Stenmark, D., y Lindgren, R. 2004. Integrating knowledge management systems with everyday work: Design principles leveraging user practice. En: *memorias de Hawaii International Conference on System Science (HICSS)*. 80245b p.
- Stewart, T. A. 2002. The case against knowledge management. *Business 2.0*, 3: 80 p.

- Strohmaier, M., y Tochtermann, K. 2005. B-kide: A framework and a tool for business process-oriented knowledge infrastructure development. *Journal of Knowledge and Process Management*, 12(3): 171-189 p.
- Teevan, J., Jones, W., y Bederson, B. B. 2006. Personal information management. *Communications of the ACM*, 49(1): 40-43 p.
- Thomsett, R. 1998. The year 200 bug: A forgotten lesson. *IEEE Software*, 15(4): 91-95 p.
- Tiwana, A. 2000. *The knowledge management toolkit: Practical techniques for building a knowledge management system*. Prentice Hall. USA. 608 p.
- Tiwana, A. 2004. An empirical study of the effect of knowledge integration on software development performance. *Information and Software Technology*, 46(13): 899-906 p.
- Tiwana, A., y Ramesh, B. 2001. Integrating knowledge on the web. *IEEE Internet Computing*, 5(3): 32-39 p.
- Tuomi, I. 1999. Data is more than knowledge: Implications of the reversed knowledge hierarchy for knowledge management and organizational memory. *Journal of Management Information Systems*, 16(3): 103-117 p.
- Valente, A., y Housel, T. 2002. A framework to analyze and compare knowledge management tools. En: *memorias de Knowledge-Based Intelligent Information Engineering and Allied Technologies (KES'2002)*, Crema, Italy. 16-18 Septiembre. 291-295 p.
- van Elst, L., Dignum, V., y Abecker, A. 2003. Towards agent-mediated knowledge management. En: *memorias de International Symposium AMKM 2003*, Stanford, CA, USA. March 24-26. 1-30 p.
- Vizcaíno, A., Piattini, M., Martínez, M., y Aranda, G. 2005. Evaluating collaborative applications from a knowledge management approach. En: *memorias de Evaluation of Collaborative Information Systems and Support for Virtual Enterprises (ECE)*, Linköping, Sweden.
- W3C-HTML. Hyper text markup language (html). Accedido en 10 de Agosto, 2007, desde <http://www.w3.org/html/>
- W3C-XML. Extensible markup language (xml). Accedido en 10 de Agosto, 2007, desde <http://www.w3.org/XML/>

- Walz, D. B., Elam, J. J., y Curtis, B. 1993. Inside a software design team: Knowledge acquisition, sharing, and integration. *Communications of the ACM*, 36(10): 63-77 p.
- Ward, J., y Aurum, A. 2004. Knowledge management in software engineering - describing the process. En: *memorias de 15th Australian Software Engineering Conference (ASWEC 2004)*, Melbourne, Australia. 137-146 p.
- Wastell, D. G., White, P., y Kawalek, P. 1994. A methodology for business process redesign: Experiences and issues. *Journal of Strategic Information Systems*, 3(1): 23-40 p.
- Wenger, E. 1998. *Communities of practice: Learning, meaning, and identity*. Cambridge University Press. Cambridge, U.K. 318 p.
- Wiig, K. 1993. *Knowledge management foundations*. Schema Press. Arlington, TX.
- Wiig, K. 2004. *People-focused knowledge management: How effective decision making leads to corporate success*. Elsevier. Amsterdam. 365 p.
- Woitsch, R., y Karagiannis, D. 2002. Process-oriented knowledge management systems based on km-services: The promote approach. *International Journal of Intelligent Systems in Accounting, Finance & Management*, 11: 253-267 p.
- Wong, K. Y. 2005. Critical success factors for implementing knowledge management in small and medium enterprises. *Industrial Management & Data Systems*, 105(3): 261-279 p.
- Yahoo. Búsquedas por directorio en yahoo. Accedido en 10 de Agosto, 2007, desde <http://dir.yahoo.com/>
- Zhuge, H. 2002. Knowledge flow management for distributed team software development. *Knowledge-Based Systems*, 15(8): 465-471 p.

APÉNDICE A

Experimento

A.1 Introducción

En este apéndice se reportan los resultados obtenidos del experimento realizado para evaluar la facilidad con que estudiantes de maestría entienden los modelos representados por medio de las extensiones realizadas a SPEM.

El reporte se presenta de la siguiente manera, primero se da una descripción sobre la forma en que se preparó el experimento, abarcando los objetivos perseguidos por el mismo, las preguntas de investigación que se busca responder, las variables que fueron medidas, y el material empleado durante el experimento. Después se describe el proceso seguido durante la realización del experimento, tomando en cuenta los sujetos que fueron tomados como muestra, y la metodología que se siguió. Posteriormente se detallan los resultados del experimento, para terminar con una discusión sobre los mismos.

A.2 Preparación: descripción del experimento

El experimento se preparó con el fin de ayudar a evaluar si los modelos desarrollados usando las extensiones hechas a SPEM, para modelar flujos de conocimiento en procesos de ingeniería de software son fáciles de entender. Para la realización del mismo, se determinó trabajar con estudiantes del posgrado de Ciencias de la Computación de CICESE, en particular, los que llevaron el curso de Ingeniería de Procesos.

Se decidió realizar la evaluación teniendo en cuenta la opinión de los participantes del estudio con respecto a un conjunto de diagramas que modelan parte de un proceso de mantenimiento de software. Además, se determinó medir el tiempo que les lleva entender los modelos y se realizaron preguntas con respecto a los modelos para medir la cantidad de

aciertos de los participantes y evaluar si realmente entendieron la información que se intentó representar en dichos modelos.

En esta sección se describen los principales objetivos del experimento, las preguntas de investigación que se buscó responder, las variables a medir para intentar responder estas preguntas y lograr los objetivos propuestos, y finalmente, una descripción del material empleado durante el experimento.

A.2.1 Objetivo del experimento

El objetivo del experimento es identificar si los modelos desarrollados con la nomenclatura propuesta permiten la identificación de lo siguiente:

- el conocimiento requerido y generado por las actividades;
- 4) el conocimiento requerido por los participantes en el proceso modelado;
- 5) las fuentes de información involucradas en las actividades;
- 6) el conocimiento que puede ser obtenido en las fuentes participantes en el proceso;
- 7) transferencias o flujos de conocimiento entre actividades y fuentes.

Con base en estos objetivos se formularon las siguientes hipótesis.

A.2.2 Hipótesis del experimento

Con el fin de representar de manera explícita el conocimiento involucrado en las actividades de un grupo de mantenimiento de software, así como la forma en que este fluye dentro del grupo, se definieron algunos elementos que se agregaron a la nomenclatura propuesta en el lenguaje de modelado de SPEM. Esto partiendo de la suposición de que la representación explícita de estos elementos dentro de los modelos, facilita su identificación por parte de las personas que analicen dichos modelos. Así también, debido a que SPEM está basado en UML, creemos que el tener experiencia en UML puede resultar benéfico para entender los modelos. Siguiendo estas premisas, se plantearon las siguientes hipótesis:

H1.- Las extensiones realizadas a SPEM permiten identificar:

- a. el conocimiento e información utilizada y generada durante las actividades involucradas en un proceso de IS;
- b. las fuentes donde este conocimiento e información es obtenido o almacenado;
- c. transferencias de conocimiento entre actividades;
- d. transferencias de conocimiento entre fuentes de conocimiento.

H2.- La nomenclatura definida para representar los elementos aquí descritos es adecuada.**H3.- La experiencia con el uso de UML puede facilitar el entendimiento de los modelos.**

Siguiendo estas hipótesis se definió un conjunto de variables con el fin de medir si dichas hipótesis son cumplidas. Estas variables se describen a continuación.

A.2.3 Variables a medir

La variable principal a medir durante el experimento, corresponde a la facilidad con que los participantes identifican los elementos representados en los modelos desarrollados con la nomenclatura de las extensiones realizadas a SPEM. Para esto se evaluarán varios aspectos:

- 1) Primero, se presentará a los participantes una serie de diagramas correspondientes a una parte de un proceso de mantenimiento de software, y se harán una serie de preguntas con respecto a la información contenida en el modelo. Dentro de esto, se llevará cuenta de lo siguiente:
 - a. el tiempo empleado por los participantes para contestar estas preguntas;

- b. el grado de acierto de las respuestas proporcionadas;
- c. grado de entendimiento del proceso descrito en los modelos, por parte de los participantes;
- d. la opinión de los participantes con respecto a los modelos del experimento, y la facilidad para entender los diagramas y la nomenclatura.

- 2) Dado que SPEM ha sido desarrollado tomando como base UML, queremos medir que tanto ayuda el tener conocimiento sobre UML a la hora de interpretar los modelos, por lo que se compararán los datos anteriormente mencionados con la experiencia de los participantes en el uso de UML, tanto a nivel académico como profesional.
- 3) A la par de lo anterior, se medirá si la experiencia profesional y académica tienen algún efecto en los resultados de los puntos evaluados.
- 4) La nomenclatura empleada para modelar flujos de conocimiento, proporciona dos modos de representar el conocimiento requerido por las actividades de un proceso, una representando directamente los temas o tópicos de conocimiento involucrados, y la otra es agrupando estos tópicos, por ejemplo el conocimiento almacenado en un determinado documento, y representando esos tópicos por medio de un paquete de temas de conocimiento. Dado que en el primer caso los diagramas podrían resultar confusos debido a la cantidad de elementos que los pueden conformar, se decidió evaluar si existe alguna diferencia en la identificación del conocimiento involucrado en las actividades si los tópicos o temas se presenta de forma explícita, o por medio de un paquete que los contenga.

Para evaluar los criterios anteriormente mencionados, se preparó un experimento que consiste en un ejercicio y dos cuestionarios descritos a continuación.

A.2.4 Material empleado

Para la realización del experimento, se desarrolló un cuestionario inicial, y un ejercicio compuesto por una serie de diagramas que describen una parte de un proceso de mantenimiento de software.

El cuestionario inicial (anexo a este apéndice) se desarrolló con el objetivo de conocer ciertos datos generales sobre los participantes del experimento. En particular, se obtuvieron datos relacionados con su área de estudio, su nivel académico, experiencia laboral, y conocimientos sobre UML. En este último punto, sólo se hicieron preguntas sobre la cantidad de veces que han utilizado UML durante sus estudios, o su trabajo, esto último en caso de que la persona contara con experiencia laboral. Así también, se les pidió a los participantes que evaluaran ellos mismos el grado de experiencia que consideran tienen en el dominio de UML.

El ejercicio (anexo a este apéndice), consistió de una serie de diagramas que representan el modelo de una parte de un proceso de mantenimiento de software, así como una serie de preguntas agrupadas en cuatro secciones principales. La primera sección consiste en un total de 22 preguntas abiertas sobre la información presentada por los modelos, las cuales fueron contestadas tomando como base los mismos. La segunda sección consiste en la descripción del proceso por parte de los participantes, con el fin de evaluar si entendieron el proceso representado en los diagramas. La tercera sección se compone de un conjunto de preguntas sobre la apreciación que tuvieron los usuarios con respecto al experimento, los diagramas, y la nomenclatura. Finalmente, la cuarta sección se compone de seis preguntas que buscan evaluar cuáles fueron los diagramas que los participantes encontraron más útiles para obtener la información que se les solicitó en el ejercicio.

Finalmente, el ejercicio presentado a los participantes fue de dos tipos, los cuales sólo difirieron en uno de los diagramas. Esto se hizo con el fin de identificar si alguna de estas representaciones presentaba mayor dificultad para ser entendida. En uno de los

ejercicios, en el diagrama de actividades titulado: “Definir actividades” (ver anexos) se presentó cada tema de conocimiento de forma independiente, mientras que en el otro se presentaron agrupados en un paquete. A continuación se describe cómo se aplicaron estos ejercicios a los participantes.

A.3 Realización del experimento

Para la realización del mismo, primeramente se llevó a cabo un experimento preliminar para evaluar los cuestionarios y ejercicios que serían presentados. Este experimento preliminar se llevó a cabo con cuatro personas, y sirvió para mejorar la redacción de algunas de las preguntas, y ajustar los niveles en las respuestas de tipo ordinal. En esta sección se describe la forma en que fueron realizados ambos experimentos, iniciando con la descripción de los participantes que formaron parte de la muestra, y siguiendo con la de la metodología empleada.

A.3.1 Descripción de la muestra

Los sujetos de la muestra fueron estudiantes del posgrado de ciencias de la computación de CICESE. Para el experimento preliminar, se escogió a cuatro estudiantes de maestría en su fase de elaboración de tesis, los cuales habían tomado con anterioridad el curso de Ingeniería de Procesos, y se encontraban aplicando técnicas de ingeniería de procesos como parte de sus proyectos de tesis de maestría.

Para el experimento principal, se eligieron diez estudiantes de reciente ingreso que se encontraban llevando el curso de Ingeniería de Procesos; dos con grado de maestría, y el resto de licenciatura. De los participantes, tres fueron de sexo femenino, y el resto masculino. La edad promedio fue de entre 25 y 26 años, siendo 21 la menor edad, y 33 la máxima. Tres de los participantes reportaron no haber llevado cursos de UML durante sus estudios. Cinco de los participantes reportaron contar con experiencia laboral en el siguiente orden: menos de un año (1), dos años (1), cuatro años (2), y 10 años (1). De los participantes con experiencia laboral, sólo tres de ellos reportaron haber empleado UML en

su trabajo. Con respecto a qué tan expertos consideran ser los participantes en el uso de UML, en una escala de 1 a 7 ellos reportaron lo siguiente: dos se consideran en el nivel 3, cuatro en el nivel 4, dos en el 5, y otros dos en el 6. Finalmente, sólo uno de los participantes reportó haber participado en el modelado de procesos en una ocasión, el resto sólo contaba con los conocimientos obtenidos durante el curso de Ingeniería de Procesos.

A.3.2 Metodología

La realización del experimento consistió de cuatro etapas principales: el experimento preliminar, una plática sobre SPEM y las extensiones realizadas para el modelado de flujos de conocimiento, la aplicación del ejercicio, y finalmente el análisis de los resultados. A continuación se describen las tres primeras etapas, dejando el análisis de los resultados para la siguiente sección.

A.3.2.1 Experimento preliminar

En la primera etapa se llevó a cabo el experimento preliminar, para lo cual a los participantes se les dio una plática general sobre SPEM y las extensiones hechas al mismo para modelar flujos de conocimiento. Se presentaron algunos ejemplos de cómo usar la nomenclatura para modelar procesos. Lo anterior con el fin de que los participantes tuvieran conocimientos básicos sobre la nomenclatura empleada, su significado, y forma de emplearla para modelar procesos. Al final de la plática, se les dio el ejercicio a los participantes para que lo contestaran. También se proporcionó copia de las diapositivas empleadas durante la plática, esto por si alguno de los participantes requería consultar información sobre el lenguaje usado en los diagramas presentados en el ejercicio. La realización de esta etapa llevó alrededor de 2 horas, incluyendo la plática y la realización del ejercicio.

A.3.2.2 Pláticas sobre SPEM y las extensiones hechas

Para la realización del segundo experimento, se dieron dos pláticas a los participantes del mismo. La primera como parte del curso de Bases de Datos, donde se

trataron aspectos básicos de administración del conocimiento y bases de conocimiento, junto con una descripción general de la metodología KoFI, desarrollada para identificar flujos de conocimiento en grupos de desarrollo de software. La segunda plática consistió en los detalles del uso de SPEM para el modelado de procesos de desarrollo de software, así como de las extensiones desarrolladas para el modelado de flujos de conocimiento en procesos de desarrollo de software. Esta segunda plática se dio como parte del curso de Ingeniería de Procesos una vez que los estudiantes habían visto la teoría básica sobre el modelado de procesos, así como distintas técnicas y lenguajes para este fin.

Las pláticas tomaron alrededor de una hora y media cada una. Para asegurar que los estudiantes asistieran a ambas pláticas, se decidió tomar los temas tratados, como parte del curso de Ingeniería de Procesos. Para esto se tomaron algunas preguntas, relacionadas a los temas vistos en ambas pláticas, como parte de uno de los exámenes que fueron aplicados a los estudiantes del curso. Así también, se tomó la participación de los estudiantes como un extra en la calificación final del curso. Finalmente, a todos los asistentes a ambas pláticas, se les dio copia de las diapositivas usadas durante las mismas, y se les pidió que las estudiaran tanto para el examen, como para la realización del ejercicio del experimento.

A.3.3 Aplicación del experimento

Para la aplicación del experimento se dividió a los participantes en dos grupos, tomando en cuenta su grado académico, el tiempo de haber egresado de la licenciatura, su experiencia laboral, así como en el uso de UML. Se buscó distribuir de forma homogénea a los participantes en ambos grupos, como se muestra en la Tabla XVI.

Tabla XVI. Datos generales de los participantes en el experimento. La edad está definida en años, así como el tiempo de egreso de la licenciatura, y la experiencia laboral. En grado académico, l equivale a licenciatura, y m a maestría. La experiencia reportada en el uso de UML va de un rango de 1 a 7, donde 1 equivale a ninguna experiencia y 7 a experto. En la parte izquierda de la tabla, se presentan los promedios por grupo y el total para los campos en que aplica.

	Grupo 1					Grupo 2					Promedio		
	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7	Part 8	Part 9	Part 10	G1	G2	Total
Edad	33	26	24	24	22	30	25	21	23	29	25.8	25.6	25.7
Grado académico	m	l	l	l	l	m	l	l	l	l			
Tiempo de egreso	11	3	2	1	1	7	3	1	1	7	3.6	3.8	3.7
Experiencia laboral	10	0	2	0.75	0	4	0	4	0	0	2.55	1.6	2.075
Experiencia reportada en el uso de UML	6	3	4	6	4	4	4	5	3	5	4.6	4.2	4.4

Como mencionamos, el ejercicio fue de dos tipos, y se aplicó un ejercicio distinto a cada grupo, el cual se asignó de forma aleatoria. Todos los participantes realizaron el ejercicio al mismo tiempo. El tiempo total utilizado por los participantes varió entre una hora y cuarto, y una hora y 3 cuartos. La primera sección del ejercicio, la cual corresponde a las preguntas sobre la información presentada en los diagramas, fue dividida en cuatro sub-secciones. Se pidió a los participantes que anotaran la hora antes de iniciar cada una de estas sub-secciones, y al finalizar las mismas. En la siguiente sección se presentan los resultados de la aplicación de los ejercicios.

A.4 Análisis de resultados

En esta sección se presenta el análisis de los resultados obtenidos de la aplicación de los ejercicios del experimento, así como una discusión sobre los mismos. Sin embargo, antes de entrar en el análisis de estos resultados, se presenta el método empleado para evaluar las respuestas a las preguntas presentadas en el ejercicio.

A.4.1 Método de evaluación

Como ya hemos mencionado, el ejercicio se dividió en cuatro secciones principales, las dos primeras para evaluar si los participantes entendieron los modelos presentados. En estas, la primera consistió en un conjunto de 22 preguntas divididas en cuatro series, y la segunda de una descripción del proceso por parte de los participantes. La evaluación de estas dos secciones se realizó por medio de una escala ordinal de 0 a 6, con el significado mostrado en la Tabla XVII.

Tabla XVII. Valores para la medición del grado de acierto a las preguntas del ejercicio.

0	<i>la pregunta no fue contestada</i>
1	<i>la respuesta es incorrecta</i>
2	<i>la respuesta es incorrecta pero de alguna forma está relacionada con la respuesta correcta</i>
3	<i>la respuesta es en parte incorrecta y en parte correcta</i>
4	<i>la respuesta es correcta pero faltan detalles importantes</i>
5	<i>la respuesta es correcta pero no completa</i>
6	<i>la respuesta es la esperada</i>

Con esta escala, se pretende medir el grado de acierto en las respuestas dadas por los participantes, tanto en preguntas individuales, como por secciones y de forma general. En este último caso, se busca comparar si la experiencia de los participantes, el tiempo empleado para responder, el tipo de ejercicio, o algún otro parámetro parece influir o estar relacionado con el grado de acierto en las preguntas. Así también, evaluar si los participantes lograron identificar la información que se buscó representar en los diagramas.

La tercera sección está orientada a medir la apreciación de los participantes con respecto al ejercicio, los diagramas y la nomenclatura empleada. En general, esta sección consiste en una serie de preguntas para medir que tan positiva o negativa es la apreciación de los participantes. La sección está dividida en dos partes, la primera consiste en una serie de preguntas cuyas respuestas son para medir que tan de acuerdo están los participantes con respecto a varios temas. La escala usada es la mostrada en la Tabla XVIII.

Tabla XVIII. Escala para medir la apreciación de los participantes.

1	<i>Totalmente en desacuerdo</i>
2	<i>En Desacuerdo</i>
3	<i>Ligeramente desacuerdo</i>
4	<i>Neutral</i>
5	<i>Ligeramente de acuerdo</i>
6	<i>De acuerdo</i>
7	<i>Totalmente de acuerdo</i>

La segunda parte de la tercera sección del ejercicio, consistió en que los participantes evaluaran diversos aspectos de la nomenclatura empleada, asignando un valor dentro de un rango de 1 a 7, dependiendo de una serie de calificativos, donde un extremo indicaba un aspecto negativo, y el otro su contraparte positiva.

Finalmente, en la cuarta sección se le pidió a los participantes que ordenaran los cinco tipos de diagramas presentados en el ejercicio, según como consideraron ellos su utilidad para obtener distintos tipos de información, con el fin de evaluar qué tipos de diagramas pueden resultar más útiles.

A.4.2 Resultados

A continuación se presentan los resultados obtenidos en el experimento, iniciando con un análisis de los tiempos empleados por los participantes, siguiendo con los resultados de las preguntas del ejercicio, y finalizando con la opinión que los participantes tuvieron con respecto al ejercicio, y el lenguaje de modelado empleado en los diagramas.

A.4.2.1 Tiempos empleados por los participantes

Como ya se ha mencionado, a los participantes se les pidió que anotaran la hora de inicio y fin en cada una de las series de preguntas de la primer parte del ejercicio. Con base en estos datos, se calculó la cantidad de minutos empleados por cada participante en dichas series, así como el total de minutos empleados para responder todas las preguntas del ejercicio. Principalmente, el objetivo de esta medición fue saber si existe alguna diferencia

en el tiempo empleado para responder los distintos ejercicios aplicados. Además, de si hay diferencia entre el tiempo empleado, y el grado de acierto en las preguntas.

Como se puede observar en la Tabla XIX, los tiempos empleados por los participantes son muy variados; abarcan desde 37 minutos quien empleó menos tiempo (participante 6), hasta 75 quien requirió mayor tiempo (participante 10), ambos en el grupo 2. Sin embargo, el promedio de tiempo empleado en cada grupo fue prácticamente el mismo. Así también, en promedio, el mayor porcentaje del tiempo total de los participantes fue utilizado durante la cuarta serie de preguntas, la cual se relacionó con las transferencias de conocimiento entre actividades y fuentes. Dado que la cantidad de preguntas así como la complejidad de las respuestas en ésta cuarta serie es similar al resto, esta diferencia la podemos atribuir a que los participantes emplearon mayor tiempo analizando los diagramas. Quizá la razón de esto se deba a que, al parecer, fue en esta serie de preguntas que los participantes analizaron con mayor detalle los diagramas de transferencia de conocimiento, y de paquetes; mientras que para el resto de las preguntas se basaron principalmente en los diagramas de actividad y casos de uso. Esto se verá más adelante al analizar la percepción de los usuarios.

Tabla XIX. Tiempos empleados por los participantes en la realización de la primera parte del ejercicio. En la parte superior se presentan los tiempos en minutos agrupados por serie de preguntas y por participantes, en la parte inferior se presenta el porcentaje del tiempo total empleado en cada serie por cada participante. En la parte izquierda se presentan los promedios de estos datos por grupo, y del total de participantes.

	Tiempos										Promedios		
	Grupo 1					Grupo 2					G1	G2	Total
	Part 1	Part 2	Part 3	Part 4	Part 5	Part 6	Part 7	Part 8	Part 9	Part 10			
Serie 1	8	16	12	10	7	4	10	10	9	16	10.6	9.8	10.2
Serie 2	11	18	9	9	9	10	7	17	13	14	11.2	12.2	11.7
Serie 3	11	16	9	14	13	9	8	13	17	16	12.6	12.6	12.6
Serie 4	23	20	14	18	17	14	13	21	15	29	18.4	18.4	18.4
Totales	53	70	44	51	46	37	38	61	54	75	52.8	53	52.9
	Porcentajes del tiempo												
Serie 1	15.09	22.86	27.27	19.61	15.22	10.81	26.32	16.39	16.67	21.33	20.08	18.49	19.28
Serie 2	20.75	25.71	20.45	17.65	19.57	27.03	18.42	27.87	24.07	18.67	21.21	23.02	22.12
Serie 3	20.75	22.86	20.45	27.45	28.26	24.32	21.05	21.31	31.48	21.33	23.86	23.77	23.82
Serie 4	43.40	28.57	31.82	35.29	36.96	37.84	34.21	34.43	27.78	38.67	34.85	34.72	34.78

A.4.2.2 Grado de acierto en las respuestas del ejercicio

Como ya hemos mencionado, para la evaluación de las respuestas dadas por los participantes a las preguntas del ejercicio, se tomó una escala de 0 a 6, donde el valor 0 se usó para indicar preguntas no contestadas. Dado que todos los participantes respondieron la totalidad de las preguntas, ninguna fue calificada como 0. Esto se puede apreciar en la Tabla XXI, donde se presentan el grado de acierto de las respuestas dadas por los participantes (la Tabla XX muestra las preguntas a las que corresponden las calificaciones de las respuestas de los participantes). Como se puede observar, el promedio de calificación fue muy similar en ambos grupos, lo que sugiere que no existió una diferencia notable en el grado en que los participantes entendieron los dos tipos de ejercicios. Si se analiza pregunta por pregunta, los promedios de calificación por grupo también son semejantes, con excepción de las preguntas 4 y 5, donde se registraron diferencias de dos o más puntos entre ambos grupos.

Tabla XX. Lista de preguntas hechas a los participantes en el ejercicio.

No.	Pregunta
1	¿Cuáles son las fuentes de información o conocimiento que participan dentro del proceso?
2	¿Qué roles están involucrados en el proceso?
3	¿Qué documentos específicos son utilizados o generados durante el proceso?
4	¿Cuál es el rol principal dentro del proceso?
5	¿Cuál es la principal fuente de información utilizada o generada durante el proceso?
6	¿En qué actividades específicas participa el jefe del departamento?
7	¿En qué actividades o flujos de trabajo se generan fuentes de información?
8	¿Qué actividades debe realizar el líder del proyecto?
9	¿Para qué actividades puede ser útil el plan del proyecto?
10	¿En qué actividades se requiere la solicitud de mantenimiento?
11	¿En general, qué requiere saber el líder del proyecto para desempeñar sus actividades?
12	¿A qué fuentes puede acudir el líder del proyecto para obtener información y conocimiento que requiere para sus actividades?
13	¿Qué información se puede obtener del plan del proyecto?
14	Además de lo anterior, ¿qué información puede proporcionar la documentación del proyecto?
15	¿Qué conocimientos de los ingenieros de software es útil para el proceso?
16	¿Qué debe considerar el líder del proyecto para decidir qué ingeniero se hará cargo de cada actividad?
17	¿Qué información se considera para estimar el tiempo que llevará cada actividad?
18	¿Qué actividades generan información o conocimiento utilizado en otras actividades?
19	¿Qué actividades generan información o conocimiento útil para los ingenieros de software?
20	¿Qué conocimiento obtiene el líder del proyecto al participar en el proceso?
21	¿Qué actividades requieren el intercambio de información o conocimiento entre los participantes del proceso?
22	Toma una de las actividades anteriores, y describe los roles participantes, el conocimiento que aportan y obtienen, y en qué actividad se da la transferencia de conocimiento

Tabla XXI. Grado de acierto en las respuestas dadas por los participantes en las preguntas. En la parte inferior, se presenta el promedio de calificación por participante, y en la parte izquierda, el promedio por grupo y del total de los participantes, tanto por pregunta como en calificación general.

	Grupo 1					Grupo 2					Promedios		
	P1	P	P3	P4	P5	P6	P7	P8	P9	P10	G1	G2	T
Pregunta 1	5	5	5	5	5	6	5	5	5	5	5.00	5.20	5.10
Pregunta 2	6	6	6	1	4	6	1	6	1	6	4.60	4.00	4.30
Pregunta 3	5	5	5	5	5	5	5	5	5	5	5.00	5.00	5.00
Pregunta 4	6	6	6	1	6	6	1	6	1	1	5.00	3.00	4.00
Pregunta 5	2	2	2	2	2	6	5	4	1	5	2.00	4.20	3.10
Pregunta 6	6	6	5	5	5	5	5	5	5	6	5.40	5.20	5.30
Pregunta 7	3	6	5	5	5	6	1	4	5	5	4.80	4.20	4.50
Pregunta 8	5	5	4	4	4	5	4	5	5	6	4.40	5.00	4.70
Pregunta 9	5	3	2	5	5	5	4	5	6	5	4.00	5.00	4.50
Pregunta 10	6	6	6	6	6	6	4	6	1	6	6.00	4.60	5.30
Pregunta 11	6	6	5	4	6	5	4	4	6	6	5.40	5.00	5.20
Pregunta 12	5	2	4	2	2	5	4	2	5	2	3.00	3.60	3.30
Pregunta 13	6	6	6	5	5	6	6	6	2	4	5.60	4.80	5.20
Pregunta 14	6	6	6	6	4	6	5	6	1	6	5.60	4.80	5.20
Pregunta 15	6	5	5	6	5	6	5	6	5	5	5.40	5.40	5.40
Pregunta 16	6	6	6	6	6	4	6	6	5	6	6.00	5.40	5.70
Pregunta 17	6	4	4	4	5	4	5	5	3	6	4.60	4.60	4.60
Pregunta 18	5	3	6	6	5	4	6	6	5	6	5.00	5.40	5.20
Pregunta 19	6	4	5	6	5	6	5	6	6	6	5.20	5.80	5.50
Pregunta 20	4	4	5	4	4	4	5	4	4	4	4.20	4.20	4.20
Pregunta 21	6	4	4	4	4	4	4	6	4	6	4.40	4.80	4.60
Pregunta 22	6	6	6	4	5	6	6	6	6	5	5.40	5.80	5.60
Promedios	5.32	4.82	4.91	4.36	4.68	5.27	4.36	5.18	3.95	5.09	4.82	4.77	4.80

En el caso de la pregunta 4, la cual solicita que el participante mencione el rol principal del proceso, al parecer hubo una confusión por parte de los participantes al definir roles. La respuesta esperada en este caso era Líder del Proyecto, la cual contestaron correctamente 6 de los participantes (4 en el grupo 1 y 2 en el grupo 2). Sin embargo, los otros cuatro participantes definieron el sub-proceso de “definir actividades” como el rol principal. La razón de esta confusión no es del todo clara. Puede relacionarse con la forma de nombrar los roles en uno de los lenguajes de modelado de procesos que los participantes vieron en clase, en específico los diagramas de rol-actividad (RADs), en los cuales se suele definir el rol desempeñado por los actores del proceso, en relación a las actividades que desempeñan. Así, en el caso del proceso mostrado en el ejercicio, el actor sería el Líder del

Proyecto, y el rol sería “Definiendo actividades”. Un problema similar se dio en el caso de la pregunta 2.

En el caso de la pregunta 5, a los participantes se les pidió nombrar la principal fuente de información usada o generada en el proceso. El objetivo de esta pregunta estuvo relacionado con la identificación de fuentes de información dentro del proceso, específicamente documentos o personas. Al parecer hubo una confusión entre los participantes cuando se les pidió identificar fuentes de información, dado que problemas similares se vieron en las respuestas a las preguntas 1 y 12. Para explicar la posible razón de esto, consideremos lo siguiente: en la nomenclatura usada en los diagramas, se utilizó un icono que representa un paquete de conocimientos o información (ver Figura 45), este icono puede ser usado para agrupar los temas o tópicos de conocimiento que pueden ser obtenidos, o son almacenados en una determinada fuente, como un documento. Debido a que éste fue el caso en los ejercicios, creemos que los participantes consideraron este elemento como la fuente de información en sí, ya que fue el nombre asignado a estos iconos los que los participantes respondieron cuando se les pidió nombrar fuentes de información. Para ver esto de forma más claro se reproducen los diagramas presentados en el experimento en la Figura 46 y 47.



Figura 45. Ejemplo de un paquete de conocimientos o información.

Considerando lo anterior, la razón de la diferencia en las respuestas a la pregunta 5 entre el grupo 1 y el 2, puede estar relacionada con el tipo de diagramas presentados en ambos ejercicios. En el diagrama de actividad principal del ejercicio para el grupo 1 (ver Figura 46), del cual se deriva la respuesta a la pregunta en cuestión, se utilizó el icono mostrado en la Figura 45, para representar la información y conocimiento que se estaba generando en dicha actividad, y que posteriormente pasaría a formar parte del documento

del plan de proyecto. Esto puede ser la razón de que en el grupo 1, los participantes contestaran que la principal fuente de información era la “Información del proyecto”. En cambio, en el grupo 2, en vez de mostrar los temas de conocimiento agrupados en el paquete mencionado, se presentaron de forma individual (ver Figura 47), esto pudo haber ocasionado que los miembros del grupo 2 buscaran dónde se estaba guardando dicha información (esto requiere revisar el diagrama del paquete “Información del proyecto”, y el diagrama sobre la estructura de las fuentes de conocimiento), lo que pudo ser la causa de que la mayoría de los miembros de este grupo dieran una respuesta correcta.

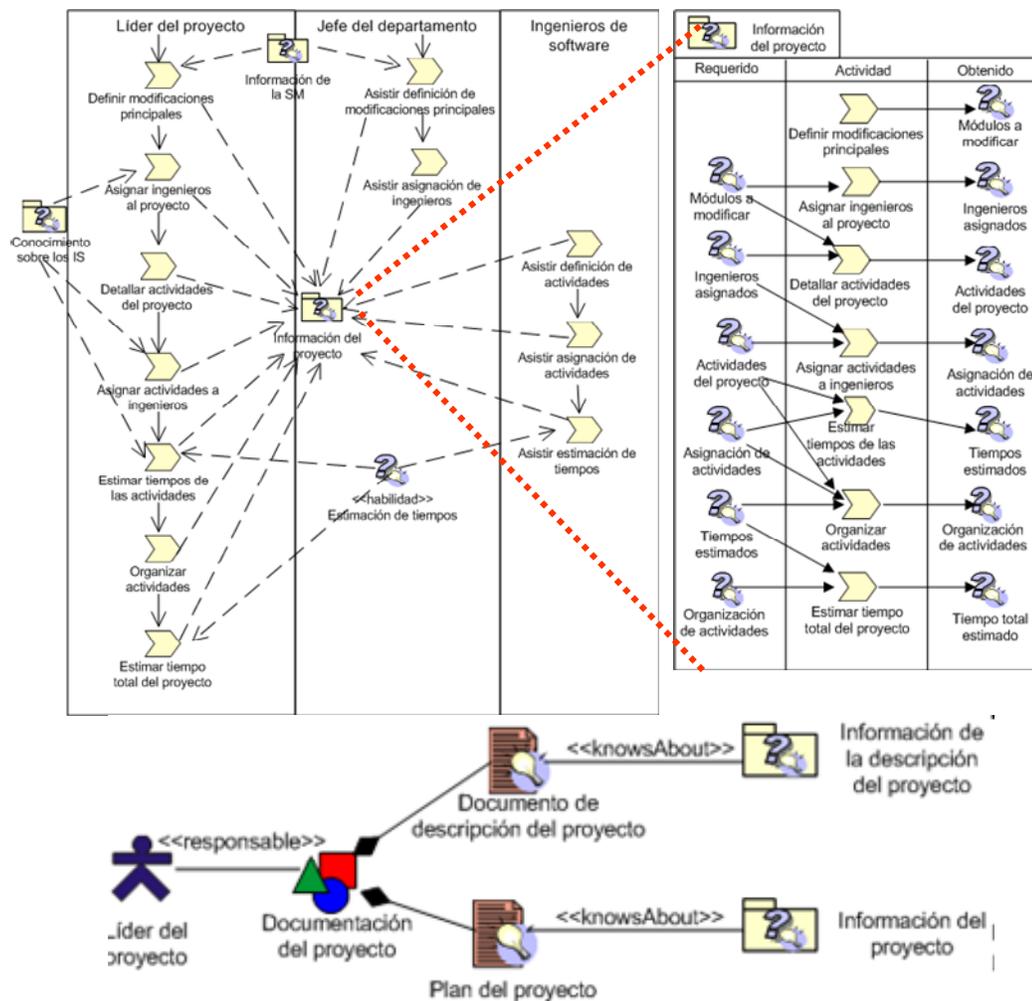


Figura 46. Diagramas mostrados al grupo 1.

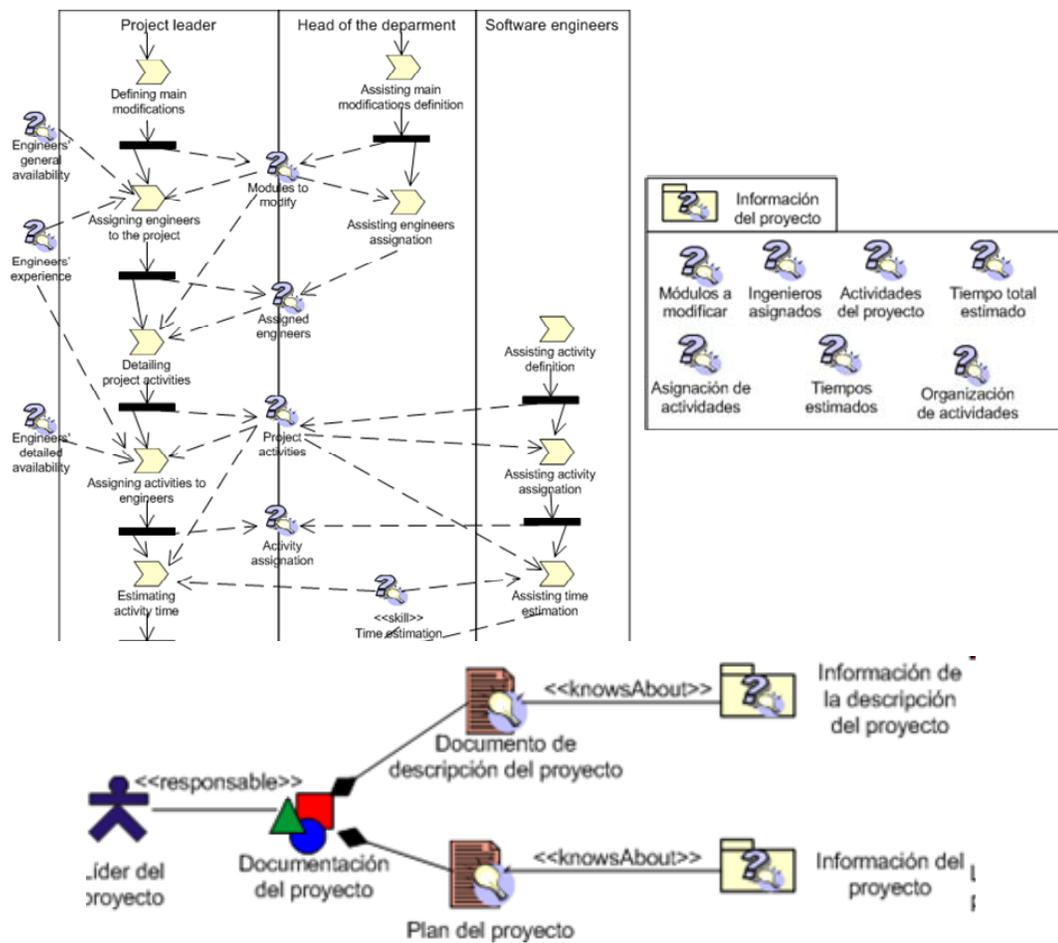


Figura 47. Diagramas mostrados al grupo 2.

Otro aspecto interesante observado fue en relación a la pregunta 21, dónde se le pidió a los participantes identificar flujos de conocimiento entre las personas participantes en el proceso. En este caso, la mayoría de los participantes contestaron las dos representadas en los diagramas de transferencia de conocimientos, mientras que tres lograron identificar las transferencias que se encuentran de forma implícita en el resto de los diagramas. En cambio, cuando se les solicitó, en la pregunta 22, que describieran los elementos participantes en una transferencia de conocimiento distinta a las dos mencionadas, todos los participantes lograron identificar una transferencia distinta a estas, aun cuando algunos dieron una respuesta menos completa que otros. Con base en esto, podemos decir que varios diagramas además de los de transferencia de conocimiento, y en

particular los diagramas de actividad, pueden ayudar a identificar flujos de conocimiento entre fuentes. Sin embargo, el proporcionar diagramas específicos para este fin puede reducir el esfuerzo cognitivo de quienes analizan los modelos.

Además del análisis realizado a las respuestas de las distintas preguntas, hemos realizado una comparación de los promedios de calificaciones con respecto a los datos de experiencia laboral, y en el uso de UML de los participantes. La Tabla XXII muestra esta comparación, donde los datos se encuentran ordenados con base en la calificación promedio de los participantes. Como se puede observar, los participantes con grado de maestría fueron quienes mejor calificación obtuvieron. También se puede observar que quienes cuentan con mayor tiempo de egreso, y con mayor experiencia laboral, se encuentran entre los mejor calificados. Esto que puede indicar que la experiencia laboral y académica en el área del desarrollo de software puede facilitar la interpretación del tipo de diagramas empleados en el experimento.

Tabla XXII. Promedio de calificación de los participantes, comparado con el tiempo total empleado en el ejercicio, y datos generales de los participantes, como su grado académico, experiencia laboral, y en el manejo de UML. Los datos se encuentran ordenados con respecto a la calificación total, la cual es un promedio de las calificaciones del total de respuestas.

	Part 9	Part 4	Part 7	Part 5	Part 2	Part 3	Part 10	Part 8	Part 6	Part 1
Calificación promedio	3.95	4.36	4.36	4.68	4.82	4.91	5.09	5.18	5.27	5.32
Tiempos total empleado en el ejercicio	54	51	38	46	70	44	75	61	37	53
Tiempo de egreso	1	1	3	1	3	2	7	0.5	7	11
Experiencia laboral	0	0.75	0	0	0	2	0	4	4	10
Grado	l	l	l	l	l	l	l	l	m	m
Estudió UML	s	s	n	s	s	s	s	s	n	n
Cuántas veces aplicó UML en la carrera	<5	<5	0	<5	<5	<5	5 a 10	>10	0	0
Usó UML en el trabajo	0	<5	0	0	0	0	0	5 a 10	0	5 a 10
Experto en UML	3	6	4	4	3	4	5	5	4	6

En lo referente al tiempo empleado por los participantes, parece no haber un factor de influencia en la calificación entre los que ocuparon mayor o menor tiempo. Así mismo,

no se ve que exista una influencia en la calificación, el conocimiento o nivel de experiencia en UML. Por ejemplo, los dos participantes mejor calificados no estudiaron UML durante su carrera. Sin embargo, uno de ellos lo ha usado durante su vida profesional. También se puede observar que quienes reportaron un mayor uso de UML tanto durante sus estudios como en su vida profesional se encuentran entre los que obtuvieron mejor calificación. Sin embargo, estos datos no reflejan una influencia de la ventaja de conocer o tener experiencia en UML. Por lo tanto, se requieren más experimentos para determinar si el conocer UML puede resultar en una ventaja para entender los modelos generados por medio de SPEM y las extensiones evaluadas en este trabajo.

Tabla XXIII. Comparación entre la calificación dada a la descripción del proceso hecha por los participantes, y su apreciación con respecto a la similitud con la descripción real del proceso, y de ésta última con los diagramas. La calificación de la descripción está dada con base en la escala usada para las preguntas del ejercicio, es decir de 0 a 6. Las calificaciones de los otros 4 puntos está en la escala usada en las preguntas de apreciación, es decir de 1 a 7, donde 1 = totalmente en desacuerdo, 7 = totalmente de acuerdo, etc.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Calificación de la descripción del proceso	6	6	6	4	6	6	6	5	4	6
La descripción proporcionada se asemeja a la que escribiste	6	6	4	5	5	6	7	7	2	6
Los diagramas son suficientes para poder entender el proceso descrito	6	6	6	4	6	3	7	5	6	7
La descripción del proceso no se asemeja a lo que entendiste de los diagramas	3	1	1	6	2	1	1	4	7	1
La información proporcionada por los diagramas es insuficiente para poder entender el proceso que se trata de representar	6	1	7	3	4	2	1	6	6	1

Con el objetivo de evaluar si los participantes habían entendido correctamente el proceso representado en los diagramas del ejercicio, se les pidió que escribieran una descripción del mismo. Posteriormente, se les presentó la descripción del proceso que fue tomada como base para realizar los diagramas, y se les pidió que indicaran qué tan de acuerdo estaban con respecto a la similitud de esta descripción con la que ellos habían escrito, así como con los diagramas proporcionados. La Tabla XXIII muestra una

comparativa entre la calificación dada a la descripción hecha por los participantes, y otros cuatro puntos relacionados con el entendimiento general del proceso por parte de los participantes. Es preciso aclarar que la escala para la calificación va de 0 a 6, mientras que para el resto de los puntos es de 1 a 7.

Como es posible observar en la Tabla XXIII, la mayoría de los participantes proporcionaron una descripción del proceso que fue considerada correcta. Así también, en su mayoría, los participantes estuvieron de acuerdo en que la descripción que ellos escribieron se asemeja a la proporcionada en el ejercicio. Sólo el participante 9 estuvo en desacuerdo en este punto, el cual fue el que proporcionó la descripción más incompleta de todos. Así también, 8 de los participantes estuvieron de acuerdo en que los diagramas proporcionados fueron suficientes para poder entender el proceso. Sólo un participante estuvo ligeramente en desacuerdo en este punto, y otro fue neutral. Como también se puede observar, los dos participantes con la calificación más baja, fueron quienes estuvieron de acuerdo en que no entendieron adecuadamente el proceso que se trató de representar. Sin embargo, con respecto a si la información proporcionada por los diagramas era suficiente o no, hubo algunas diferencias importantes entre los distintos participantes, cuatro de ellos consideraron que era insuficiente, aunque estuvieron de acuerdo en que los diagramas fueron suficientes. Quizá esto se deba a qué consideraron que hizo falta mayor información para conocer algunos de los detalles del proceso presentados en la descripción del mismo; aunque esto no es posible determinarlo con los datos recabados.

A.4.2.3 Opinión de los participantes

En esta sección presentamos los resultados de las preguntas realizadas para medir la apreciación de los participantes con respecto a la técnica de modelado empleada en los ejercicios, lo cual incluye la nomenclatura, así como los distintos tipos de diagramas. Debido a que en la mayoría de los casos es fácil apreciar la tendencia de los participantes, sólo se tratarán las preguntas cuyas respuestas presentaron aspectos de interés para los objetivos principales del experimento.

Las preguntas realizadas en esta sección del experimento, estuvieron divididas en tres tipos, en la primera serie de preguntas se les pidió a los participantes que definieran en qué grado estaban de acuerdo o en desacuerdo con ciertos aspectos. Los resultados a estas preguntas son presentados en la Tabla XXIV. Como se puede observar, en su mayoría, los participantes estuvieron de acuerdo en que la nomenclatura empleada es fácil de asociar con los elementos que se busca representar, y que permite la identificación de fuentes de información y conocimiento, así como de flujos de conocimiento dentro de un proceso.

Tabla XXIV. Respuestas a las preguntas de apreciación de los participantes. En la parte izquierda se muestra la cantidad de participantes que dieron la misma respuesta; donde 1 = totalmente en desacuerdo, 2 = en desacuerdo, 3 = ligeramente en desacuerdo, 4 = neutral, 5 = ligeramente de acuerdo, 6 = de acuerdo, y 7 = totalmente de acuerdo.

Pregunta	Respuestas						
	1	2	3	4	5	6	7
5 Es fácil asociar los íconos de la nomenclatura utilizada con los elementos que representan	0	1	0	0	2	4	3
6 La información presentada por las distintas vistas es redundante	1	1	1	2	1	1	3
7 La variedad de los diagramas incrementa el entendimiento del proceso	1	0	1	2	0	3	3
8 La información representada en los diagramas se complementa	0	0	1	0	2	5	2
9 La utilización de paquetes para separar el conocimiento de las fuentes facilita la identificación del conocimiento involucrado en las actividades	0	0	0	2	1	1	6
10 El tener conocimiento de UML facilita el entendimiento de los diagramas	1	1	0	0	1	3	4
11 Preferirías la utilización de otras técnicas de modelado de procesos para representar el tipo de información del ejemplo presentado en el ejercicio	1	2	1	4	1	1	0
12 Los diagramas usados para modelar un proceso siguiendo el lenguaje de modelado son demasiado detallados	0	3	0	1	2	3	1
13 Los diagramas generados con la nomenclatura pueden ayudar a entender cómo fluye el conocimiento y la información dentro de un proceso	0	0	0	0	1	6	3
14 Se requieren diagramas más detallados para poder representar correctamente los flujos de conocimiento	1	3	2	1	0	1	2
15 Los modelos ayudan a identificar las fuentes de información y conocimiento involucradas en el proceso	0	0	1	0	0	2	7

Un aspecto interesante es que hubo discrepancias entre los participantes al determinar si la información presentada en los distintos diagramas es redundante (pregunta 6), ya que 3 estuvieron en desacuerdo, cinco de acuerdo, y 2 fueron neutrales. Estas respuestas contrastan con las de las preguntas 7 y 8, ya que 6 de los participantes consideraron que la variedad de los diagramas incrementa el entendimiento del proceso, y 9 de ellos que la información de los distintos diagramas se complementa. Por otro lado, 6 de

los participantes consideraron que los diagramas generados con el lenguaje de modelado son demasiado detallados (pregunta 12), lo que concuerda con la pregunta 14, donde también 6 participantes no creen que se requieran diagramas más detallados.

Tabla XXV. Muestra la apreciación de los participantes con respecto a distintos adjetivos asignados a los distintos elementos del lenguaje de modelado empleado en el ejercicio. Las respuestas muestran la suma de los participantes que asignaron un determinado nivel. Los adjetivos se encuentran en la parte superior de cada tabla, donde la equivalencia con respecto al nivel es la siguiente: para el calificativo de confusa o clara, 1 = confusa, 7 = clara, 4 = neutral, y así sucesivamente.

	Respuestas						
	1	2	3	4	5	6	7
	Confusa o Clara						
Nomenclatura	0	0	1	1	2	3	3
Casos de uso	0	0	0	0	3	6	1
Diag. de actividad	0	0	1	0	1	4	4
Paquetes	0	0	1	1	2	3	3
Transf. de conocimiento	0	0	0	3	3	1	3
Est. de fuentes de con.	0	0	1	2	2	2	3
Todo en conjunto	0	0	0	2	0	8	0

	Respuestas						
	1	2	3	4	5	6	7
	Fácil o Difícil de entender						
Nomenclatura	4	2	1	0	1	2	0
Casos de uso	3	3	0	1	1	2	0
Diag. de actividad	3	3	0	0	2	2	0
Paquetes	3	3	1	1	1	1	0
Transf. de conocimiento	0	3	2	3	0	1	1
Est. de fuentes de con.	1	2	3	1	0	2	1
Todo en conjunto	0	6	0	1	2	1	0

	Respuestas						
	1	2	3	4	5	6	7
	Completa o Incompleta						
Nomenclatura	3	4	1	0	0	2	0
Casos de uso	4	2	2	0	0	2	0
Diag. de actividad	3	4	1	0	0	2	0
Paquetes	4	1	1	2	0	2	0
Transf. de conocimiento	2	2	2	2	0	1	1
Est. de fuentes de con.	3	1	2	2	0	1	1
Todo en conjunto	2	4	2	0	0	2	0

	Respuestas						
	1	2	3	4	5	6	7
	Insuficiente o Suficiente						
Nomenclatura	0	0	0	0	1	3	6
Casos de uso	0	0	0	0	2	1	7
Diag. de actividad	0	0	0	1	1	5	3
Paquetes	0	0	0	0	2	4	4
Transf. de conocimiento	0	0	0	4	1	0	5
Est. de fuentes de con.	0	0	0	3	1	2	4
Todo en conjunto	0	0	0	1	0	6	3

	Respuestas						
	1	2	3	4	5	6	7
	Difícil o Fácil de aprender						
Nomenclatura	0	0	0	1	1	4	4
Casos de uso	0	0	0	0	1	4	5
Diag. de actividad	0	0	0	0	2	3	5
Paquetes	0	0	0	0	2	4	4
Transf. de conocimiento	0	0	0	0	2	6	2
Est. de fuentes de con.	0	0	0	1	1	6	2
Todo en conjunto	0	0	0	0	2	4	4

	Respuestas						
	1	2	3	4	5	6	7
	Innecesaria o Necesaria						
Nomenclatura	0	0	1	0	2	3	4
Casos de uso	0	0	0	0	2	3	5
Diag. de actividad	0	0	0	0	2	2	6
Paquetes	0	0	0	2	2	1	5
Transf. de conocimiento	0	0	1	0	3	4	2
Est. de fuentes de con.	0	0	0	1	2	4	3
Todo en conjunto	0	0	0	0	2	5	3

	Respuestas						
	1	2	3	4	5	6	7
	Fácil o Difícil de seguir						
Nomenclatura	3	2	0	0	2	2	1
Casos de uso	2	4	0	1	0	1	2
Diag. de actividad	1	3	3	0	0	2	1
Paquetes	3	2	0	1	2	1	1
Transf. de conocimiento	0	3	2	1	2	0	2
Est. de fuentes de con.	1	3	1	1	2	0	2
Todo en conjunto	1	5	0	0	2	1	1

	Respuestas						
	1	2	3	4	5	6	7
	Rebuscada o Adecuada						
Nomenclatura	1	0	0	0	3	3	3
Casos de uso	0	0	0	1	2	4	3
Diag. de actividad	0	0	0	1	2	5	2
Paquetes	0	0	0	1	2	3	4
Transf. de conocimiento	0	0	1	2	3	2	2
Est. de fuentes de con.	1	3	1	1	2	0	2
Todo en conjunto	0	0	0	1	1	5	3

En la segunda serie de preguntas de esta sección, se les pidió a los participantes que calificaran distintos aspectos del lenguaje de modelado según una serie de adjetivos. La Tabla XXV muestra las respuestas de los participantes a esta pregunta. Como se puede observar, en términos generales los participantes dieron una evaluación positiva, dado que

hubo una mayor tendencia a considerar que el lenguaje de modelado es claro, fácil de entender y de aprender, necesario, suficiente y adecuado. Sin embargo, hubo varios participantes que consideraron varios de los aspectos del lenguaje como difíciles de seguir.

Tabla XXVI. Evaluación de los participantes con respecto a la utilidad de los distintos tipos de diagrama. Se presenta la pregunta, los tipos de diagramas, y el orden de importancia dado por los participantes. El número 1 fue dado al más importante, 2 al segundo, y así sucesivamente. En el campo “importancia”, se presenta el número total de participantes que asignaron el grado de importancia indicado por el número en la parte superior de la tabla.

Pregunta		Importancia					Pregunta		Importancia				
		1	2	3	4	5			1	2	3	4	5
1- ¿Qué tan útiles los consideras para el entendimiento general del proceso?	Casos de uso	4	3	2	1	0	2- ¿Qué diagramas te ayudaron a identificar los flujos de conocimiento entre actividades?	Casos de uso	2	0	3	2	3
	Actividad	6	2	1	1	0		Actividad	5	2	1	0	2
	Paquetes	0	3	1	3	3		Paquetes	1	5	1	3	0
	Transferencia	0	2	3	3	2		Transferencia	2	1	4	1	2
	Clases	0	0	3	2	5		Clases	0	2	1	4	3
3- ¿Qué diagramas te ayudaron a identificar las fuentes de información o conocimiento?	Casos de uso	1	2	4	2	1	4- ¿Qué diagramas te ayudaron a identificar los flujos de conocimiento entre fuentes?	Casos de uso	0	2	4	1	3
	Actividad	4	1	1	2	2		Actividad	2	0	4	3	1
	Paquetes	2	3	3	2	0		Paquetes	3	1	1	4	1
	Transferencia	1	1	2	3	3		Transferencia	4	3	0	1	2
	Clases	2	3	0	1	4		Clases	1	4	1	1	3
5- ¿Qué diagramas te ayudaron a identificar el conocimiento utilizado y generado en las actividades?	Casos de uso	1	1	2	5	1	6- ¿Qué diagramas te ayudaron a identificar los roles participantes en el proceso?	Casos de uso	8	2	0	0	0
	Actividad	5	2	1	0	2		Actividad	2	5	1	2	0
	Paquetes	1	4	3	2	0		Paquetes	0	0	7	1	2
	Transferencia	1	1	3	1	4		Transferencia	0	3	0	4	3
	Clases	2	2	1	2	3		Clases	0	0	2	3	5

La Tabla XXVI presenta las respuestas a la tercera serie de preguntas de esta sección, en la cual se les pidió a los participantes que ordenaran los distintos tipos de diagramas según qué tan útiles los consideraron para la identificación de los distintos datos solicitados en el ejercicio. Como se puede observar, entre los diagramas más útiles se encuentran los de actividad y casos de uso. Sin embargo, para la identificación de aspectos como los flujos de conocimiento entre fuentes y actividades, los diagramas de paquetes y de transferencia de conocimientos también resultaron ser útiles. Los diagramas de clases, que corresponden a los de la estructura de fuentes de conocimiento, fueron los considerados

menos útiles, aunque hubo participantes que los consideraron en primer o segundo lugar a la hora de identificar fuentes de información o conocimiento, flujos de conocimiento entre fuentes, así como el conocimiento que pueden contener esas fuentes.

Los resultados de las distintas partes del experimento, descritos en esta sección, fueron analizados y comparados para evaluar si proporcionan información útil para aceptar o refutar las hipótesis planteadas, así como para medir si los objetivos del experimento fueron cumplidos. En la siguiente sección se presenta una discusión sobre éste análisis.

A.4.3 Discusión de los resultados

Como se mencionó en el inicio de este documento, el experimento se diseñó con el objetivo de mediar si las extensiones hechas a SPEM permiten la identificación del conocimiento e información utilizada y generada durante las actividades involucradas en un proceso de IS; las fuentes donde este conocimiento e información es obtenido o almacenado; transferencias de conocimiento entre actividades; y transferencias de conocimiento entre fuentes de conocimiento. Así mismo, si la nomenclatura empleada es adecuada para este propósito. Para definir si los datos recabados en el experimento pueden ayudarnos a medir si se cumplen estos objetivos, a continuación se presenta una discusión de cada uno de estos puntos.

A.4.3.1 Identificación del conocimiento usado en las actividades

Para la representación del conocimiento usado y generado en las actividades de un proceso de IS, por medio de las extensiones realizadas a SPEM, se hace principalmente por medio de diagramas de actividad y una adaptación hecha a los diagramas de paquetes. Los diagramas de actividad se definieron con el fin de permitir representar de forma explícita el conocimiento que se genera y el que es utilizado en las actividades de un proceso. La modificación de los diagramas de paquetes es utilizada para representar el conocimiento que puede ser obtenido o almacenado en una determinada fuente de información o conocimiento, así como la actividad donde ese conocimiento es usado o generado. Para definir si estos diagramas realmente cumplen su propósito, podemos tomar tanto las

calificaciones del experimento, como la apreciación de los participantes con respecto a la utilidad de estos diagramas.

Para el primer caso hemos considerado las respuestas de las preguntas de la 11 a la 20 del ejercicio, mostradas en la Tabla XXI, las cuales están relacionadas con el conocimiento usado o generado en las actividades. Salvo en las preguntas 12, 17 y 20, el promedio general de calificación fue mayor a 5 puntos, lo que indica que los participantes sí lograron identificar el conocimiento involucrado en el proceso. En el caso de la pregunta 12, como ya se mencionó, existió una confusión entre las fuentes de información, y la información contenida en esa fuente (ver sección A.4.2.2). En el caso de las preguntas 17 y 20, los participantes contestaron de manera muy general, omitiendo algunos detalles. Sin embargo, las respuestas dadas por los participantes con respecto al conocimiento involucrado en las actividades fueron en su mayoría correctas, lo que sugiere que realmente lograron identificar estos aspectos por medio de los diagramas presentados.

Por otra parte, estos dos tipos de diagramas fueron de los que los participantes consideraron más importantes para la identificación del conocimiento involucrado en las actividades, como se observó en los resultados mostrados en la pregunta 5 de la Tabla XXVI, donde en cada caso, 8 de los participantes los consideraron entre los tres primeros grados de importancia. Lo anterior sugiere que los diagramas de actividades y de paquetes realmente ayudan en la identificación del conocimiento involucrado en las actividades de un proceso de IS.

A.4.3.2 Identificación de fuentes de conocimiento

En el caso de la identificación de fuentes de conocimiento, hemos mencionado que existió una confusión por parte de los participantes en el experimento. Debido a esta confusión, el promedio de calificación de algunas de las preguntas relacionadas con este tema fue bajo. Sin embargo, creemos que dejando bien claro lo que es una fuente de información, y lo que es un paquete que agrupa temas de conocimiento e información, el problema debe solucionarse.

Por otro lado, como se observa en las respuestas a la pregunta 15 de la Tabla XXIV, los participantes están de acuerdo en que los modelos presentados en el ejercicio ayudan a identificar las fuentes de información o conocimiento involucradas en el proceso. Lo que sugiere que también este objetivo se está cumpliendo. Sin embargo, con respecto a qué diagramas fueron los más útiles para identificar fuentes de conocimiento, no existió una clara tendencia de los participantes al respecto, aunque los diagramas de actividad fueron los que más participantes consideraron en primer lugar, como se puede observar en la pregunta 3 de la Tabla XXVI.

A.4.3.3 Identificación de transferencias de conocimiento

Con el objeto de facilitar la identificación de flujos de conocimiento, se definieron mecanismos dentro de las extensiones hechas a SPEM, para representar transferencias de conocimiento e información entre actividades, y fuentes de información. En particular, se definió un nuevo tipo de diagrama para este fin, el cual permite representar de manera explícita las transferencias de conocimiento que se dan entre dos o más fuentes de conocimiento durante un flujo de trabajo o una actividad específica. Las transferencias de conocimiento entre actividades están representadas en los diagramas de actividad, los cuales permite definir los conocimientos generados en una actividad, y usados en otras.

Las respuestas a las preguntas 18, 19, 21 y 22, presentadas en la Tabla XXI, fueron las utilizadas para evaluar si los participantes podían identificar transferencias de conocimiento en los diagramas presentados en el ejercicio. Como se observa, las respuestas de las preguntas 18 y 19 obtuvieron un promedio de calificación mayor a 5, lo que sugiere que los participantes sí lograron identificar transferencias de conocimiento. Más aun, todos los participantes estuvieron de acuerdo en que los diagramas generados con la nomenclatura pueden ayudar a entender como fluye el conocimiento en un proceso, lo que se puede observar en la respuesta a la pregunta 13 de la Tabla XXIV.

Las preguntas 21 y 22 arrojaron datos interesantes. En la pregunta 21 se pidió identificar los flujos de conocimiento que se dan entre los participantes en el proceso. En su

mayoría, proporcionaron sólo las dos representadas en los diagramas de transferencias de conocimiento. Por otro lado, cuando en la pregunta 22 se les pidió que describieran una transferencia distinta a estas dos, todos los participantes lograron identificar una. Esto indica que no solo los diagramas de transferencia de conocimiento pueden ser útiles para identificar flujos de conocimiento entre fuentes, lo que también se puede observar en las respuestas a la pregunta 4 de la Tabla XXVI, donde varios participantes también consideraron los diagramas de paquetes, clases y actividad, como útiles para este propósito. Aun así, el proporcionar diagramas específicos para representar transferencias de conocimiento explícitamente, puede reducir el esfuerzo de quienes analizan los modelos, de modo que no tenga que recurrir a analizar otros diagramas. Además, como se observó en las respuestas a la pregunta 21, quizá el no definir explícitamente estas transferencias puede provocar que algunas de ellas no sean tomadas en cuenta.

Con base en todo lo anterior, consideramos que los mecanismos proporcionados para representar transferencias o flujos de conocimiento entre actividades y fuentes también cumplen su propósito, y permiten identificar información que de otra manera podría no ser tomada en cuenta.

A.4.3.4 Nomenclatura

El último punto del experimento, fue conocer si la nomenclatura empleada por las extensiones realizadas es adecuada. Para esto se hicieron varias preguntas a los participantes, primero, como se puede observar en las respuestas a la pregunta 5 en la Tabla XXIV, 9 de los participantes consideraron que los iconos usados en la nomenclatura son fáciles de asociar con los elementos que representan. También, hubo una mayor tendencia de los participantes a considerar que la nomenclatura y los tipos de diagramas empleados, son adecuados, además de claros, fáciles de entender y aprender.

Si consideramos que la sesión para explicar la nomenclatura y la forma de representar los diagramas fue de menos de hora y media, y que los participantes no tuvieron problemas graves para entender los diagramas, esto puede sugerir que la apreciación de que

es fácil de entender y aprender es cierta. Sin embargo, como ya se ha mencionad, se presentaron algunas confusiones que es preciso tener en cuenta, como la de las fuentes de información, y la asignación de roles.

Por otro lado, tanto durante el experimento preliminar, como en el final, se dieron algunos comentarios de los participantes con respecto a la nomenclatura que hay que tomar en cuenta. Primero, en los casos de uso se usaron dependencias para asociar el conocimiento generado y requerido por las actividades, con los flujos de trabajo. En la nomenclatura de UML, las dependencias son flechas con líneas punteadas, cuya dirección va del elemento dependiente, al elemento del que éste depende. Esto causó confusión entre los participantes, sobre todo en los diagramas de actividad, ya que SPEM define que las conexiones entre los objetos generados y usados en las actividades, son líneas semejantes a las usadas en las dependencias, sólo que en este caso la dirección va de la actividad donde se generan al objeto generado, o del objeto usado a la actividad donde se usa. Es decir, en el sentido inverso en el que este tipo de flechas son utilizadas en los casos de uso. Si bien la experiencia podría evitar este tipo de confusiones, con el fin disminuirlas estamos considerando que las dependencias usadas en los casos de uso también indiquen qué tipo de conexión son, esto es, si se refieren a conocimiento usado, generado, o modificado.

Finalmente, aun cuando el experimento fue realizado con un número de participantes muy pequeño, los datos muestran una tendencia que sugiere que las extensiones hechas a SPEM cumplen con los objetivos para los que fueron diseñadas. Sin embargo, existen algunos resultados que requieren de un estudio más amplio.

A.4.3.5 Experiencia en el uso de UML

El último punto a evaluar en el experimento fue si el tener conocimientos sobre UML puede facilitar el entendimiento de los modelos desarrollados con las extensiones hechas a SPEM. Como se observó en el análisis de la Tabla XXII, no existió una influencia clara entre la experiencia en UML que los participantes reportaron, y el promedio de calificación que obtuvieron. En contraste, en las respuestas a la pregunta 10 de la Tabla

XXIV, la mayoría de los participantes consideró que el tener conocimientos de UML sí facilita el entendimiento de los modelos presentados en el experimento. Esto puede sugerir que sí puede existir una influencia entre el tener conocimientos de UML, y la facilidad para entender los diagramas desarrollados con el lenguaje de modelado propuesto, sin embargo, es necesario realizar más experimentos para ver si efectivamente esto es cierto.

A.5 Conclusiones

En este apéndice se ha presentado el reporte de los resultados obtenidos en el experimento realizado para evaluar el lenguaje de modelado propuesto en la metodología KoFI, desarrollada para facilitar la identificación de flujos de conocimiento en organizaciones de desarrollo de software. Este lenguaje está basado en SPEM, y consiste en una serie de extensiones diseñadas para facilitar la representación de flujos de conocimiento en procesos de desarrollo de software. Esto incluye la representación del conocimiento o información generado o usado en las actividades del proceso, las fuentes donde es obtenido o almacenado, y las transferencias de conocimiento entre fuentes y actividades. Con base en los resultados obtenidos en el experimento, podemos concluir que los objetivos para los cuales se definieron las extensiones se están cumpliendo.

Sin embargo, hay algunos puntos que es necesario tratar, como lo son las confusiones que existieron entre los participantes con respecto a las fuentes de información, y a los roles. Así también, no se pudo determinar con certeza si el tener experiencia en el uso de UML facilita el entendimiento del lenguaje de modelado. Aunque en su mayoría los participantes consideraron que sí, creemos que se requieren más datos para poder afirmar esto con certeza. Sin embargo, se observó que la experiencia profesional en el desarrollo de software sí puede tener un factor de influencia en la facilidad con que se entienden los diagramas desarrollados con la nomenclatura propuesta.

A.6 Anexos

A continuación se presentan el cuestionario inicial que se le aplicó a los participantes, los dos ejercicios usados en el experimento, los diagramas de ambos

ejercicios son prácticamente iguales, la principal diferencia se encuentra en el diagrama “Definir actividades”; y las preguntas hechas a los participantes después de aplicarles el ejercicio.

A.6.1 Cuestionario Inicial

I.- Datos generales

Nombre: _____

Edad: _____

Grado académico: _____

Área de estudios profesionales: _____

Áreas de estudio actual: _____

1.- ¿Hace cuanto tiempo saliste de la carrera? _____

2.- ¿Viste o estudiaste UML durante tu carrera?:

Si No

3.- ¿Cuántas veces aplicaste UML durante tus trabajos escolares?

Ninguna 1 vez menos de 5 5 -10 más de 10

4.- ¿Tienes experiencia laboral en el área de ingeniería de software?

Si Tiempo: _____ No

En caso de responder Sí a la pregunta anterior contesta las siguientes dos, en caso contrario ir a la pregunta 7:

5. ¿Qué puestos desempeñaste y por cuanto tiempo?

1 _____

2 _____

3 _____

6. ¿Cuántas veces has usado lenguajes de modelado en tú trabajo?

Ninguna 1 vez menos de 5 5 -10 más de 10

¿Cuáles? _____

7.- ¿Has realizado análisis y diseño de sistemas?

No 1 vez menos de 5 5 -10 más de 10

8.- De los siguientes tipos de diagramas de UML, menciona cuantas veces has usado cada uno, con base en la siguiente distribución: Ninguna, 1 vez, menos de 5, 5 a 10, más de 10.

Casos de uso _____ Clases _____

Secuencia _____ Colaboración _____

Estados _____ Actividades _____

Distribución _____ Componentes _____

9.- ¿Qué tato consideras que sabes sobre la utilización de UML?

Nada Experto

1 2 3 4 5 6 7

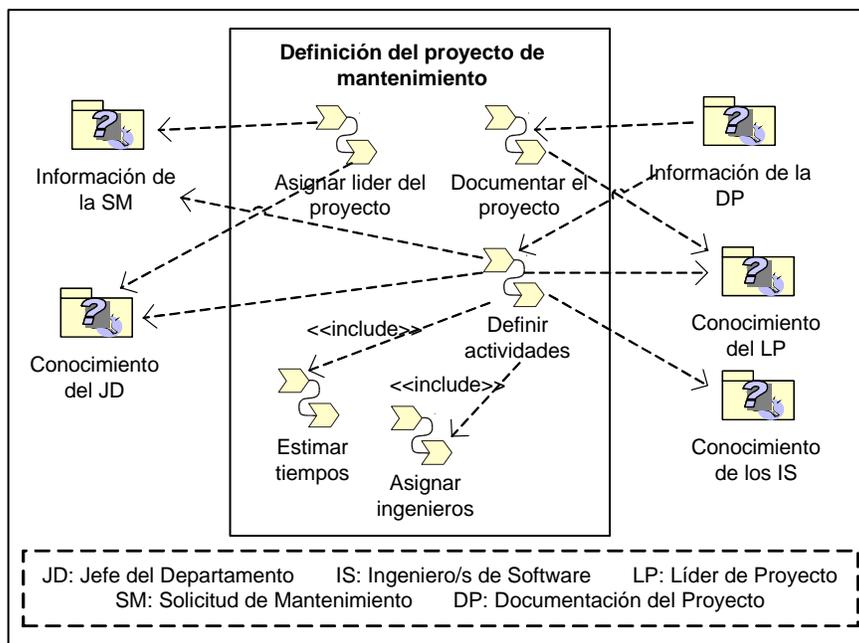
10.- ¿Has trabajado con modelado de procesos fuera del curso de ingeniería de procesos?
 No 1 vez menos de 5 5 -10 más de 10

Observaciones:

A.6.2 Ejercicio primer grupo (Ejercicio 1a)

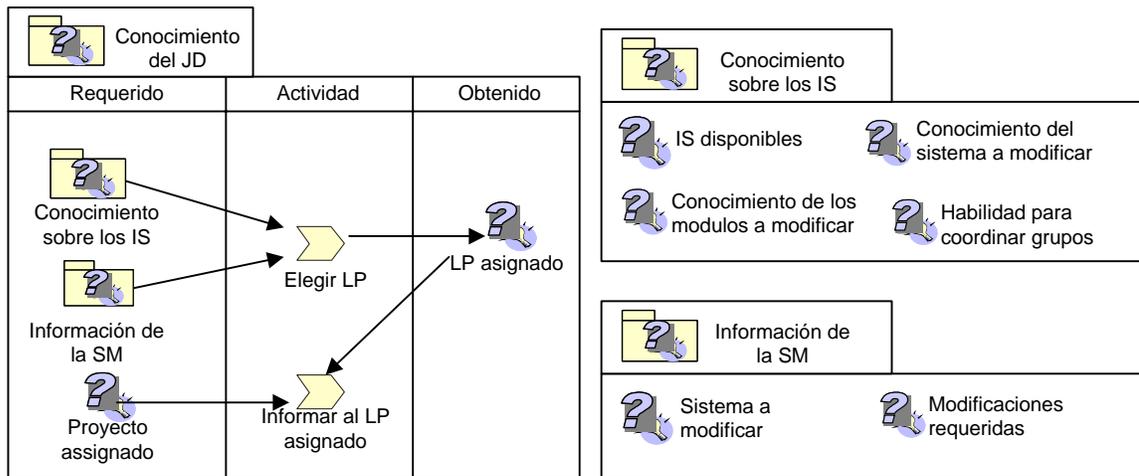
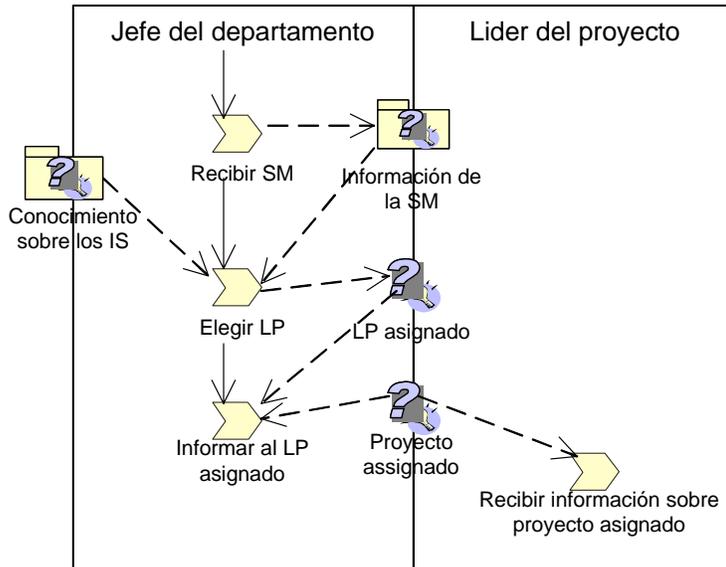
Nombre: _____

Casos de uso

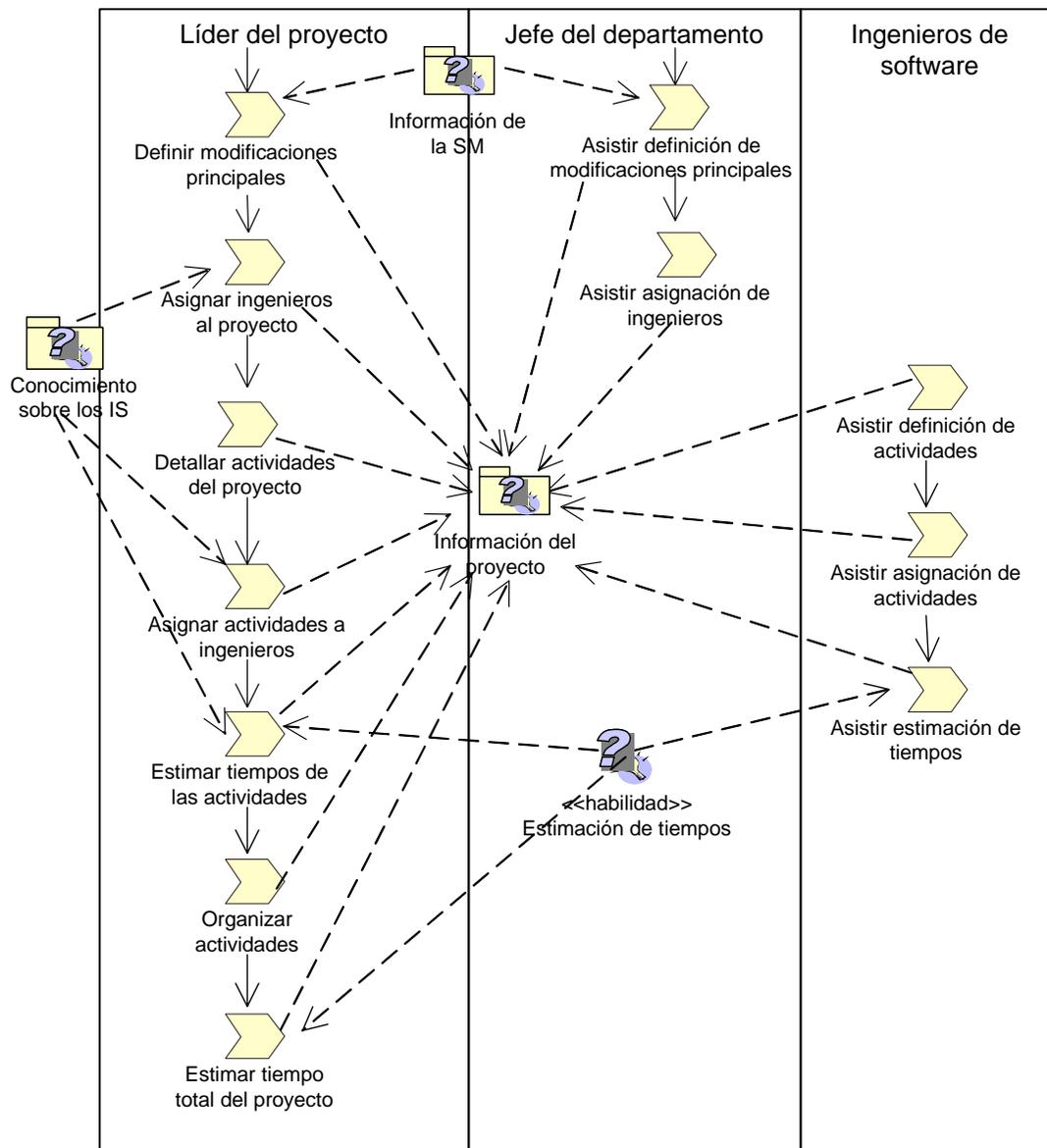


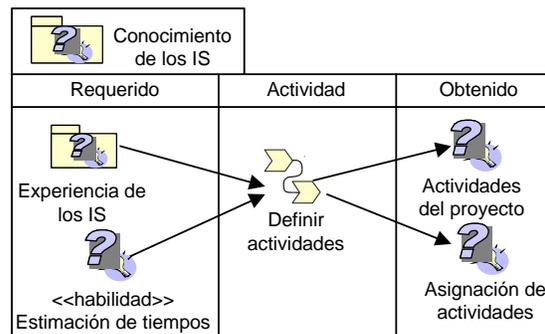
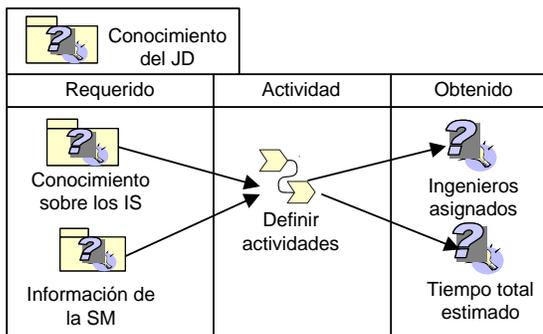
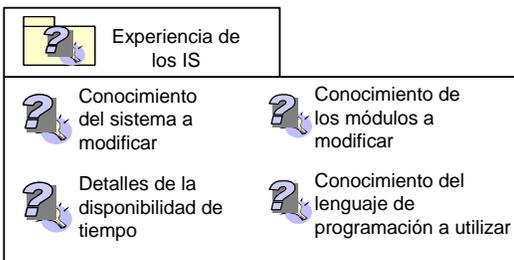
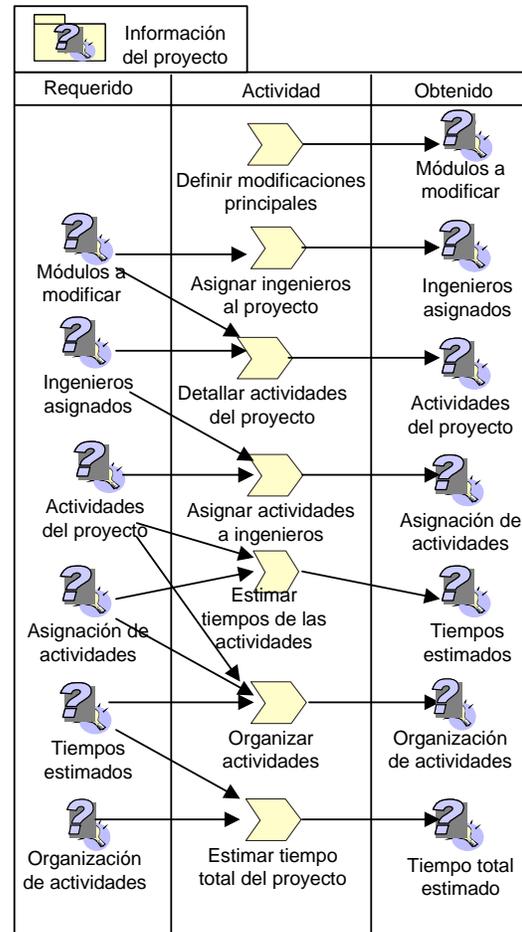
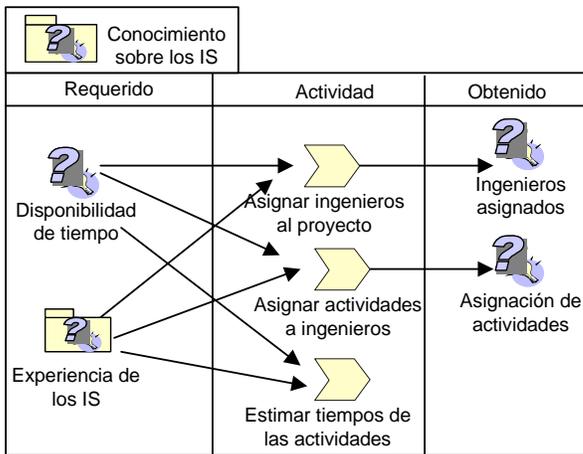
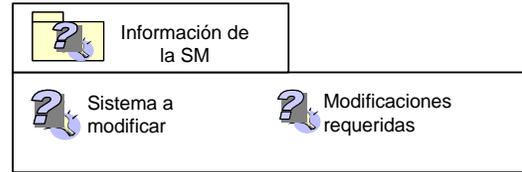
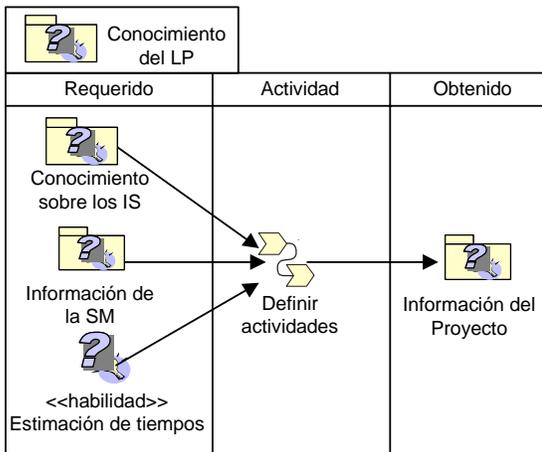
Actividades:

Asignar líder del proyecto

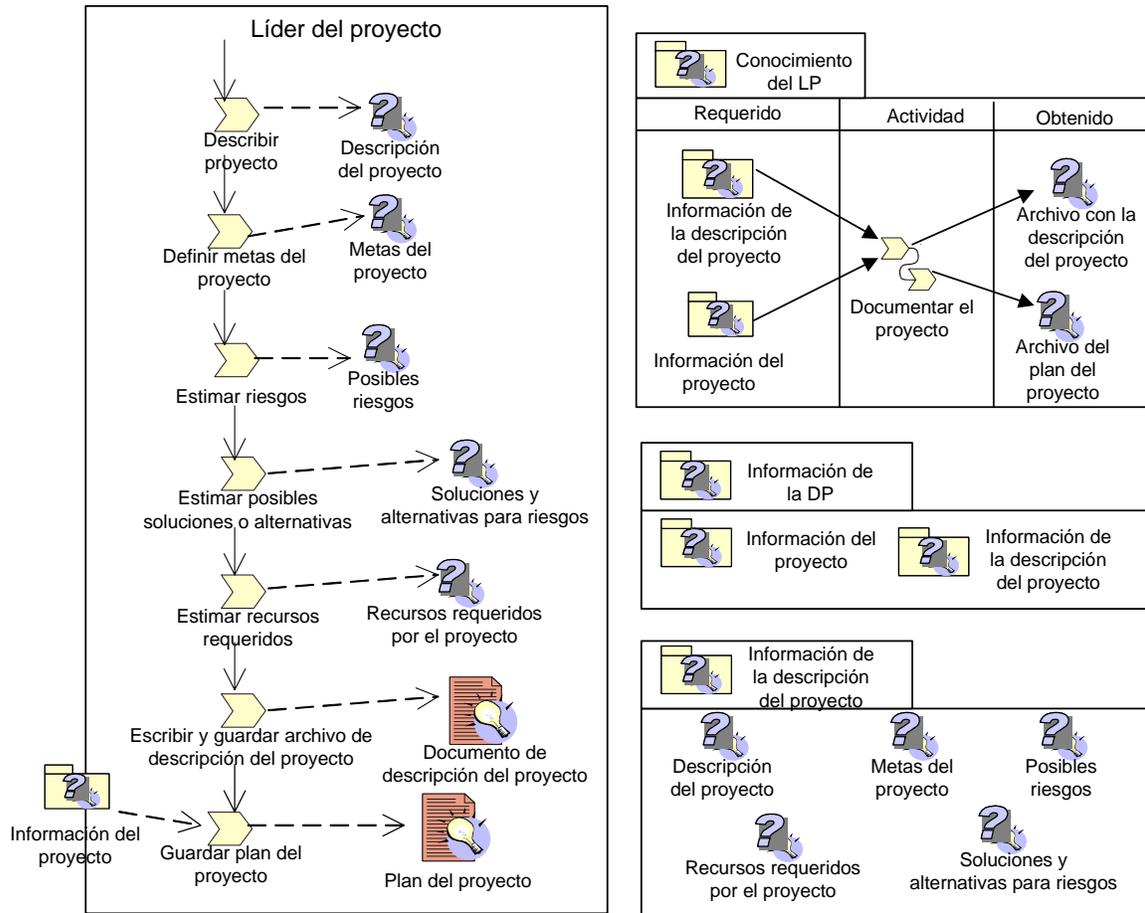


Definir actividades

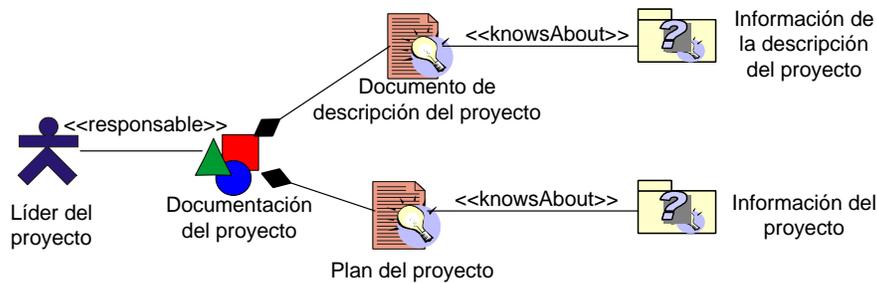




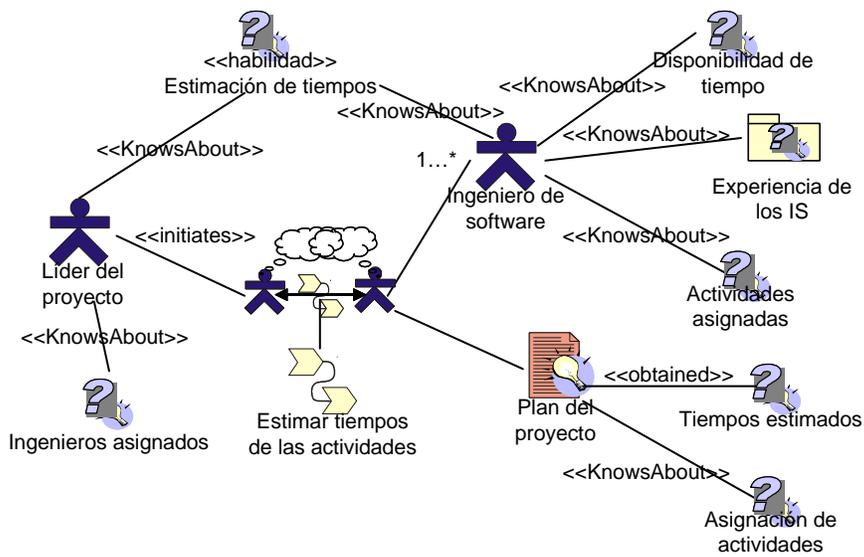
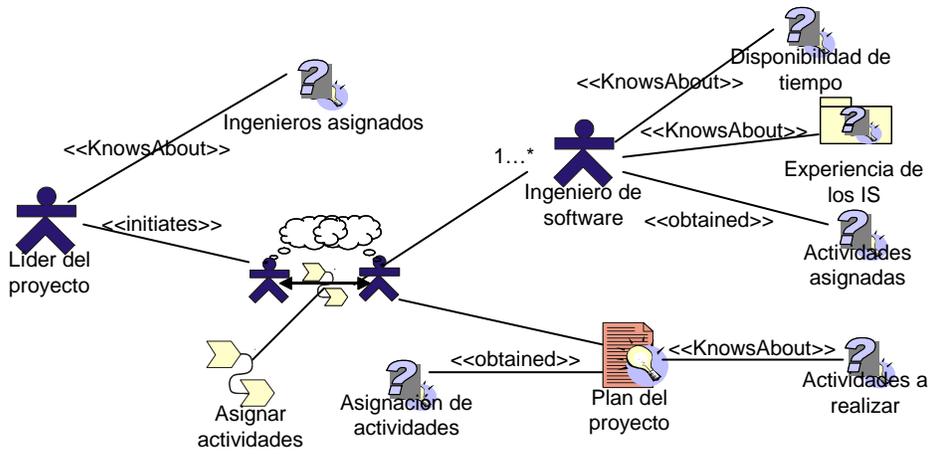
Documentar el proyecto



Estructura de las fuentes de conocimiento



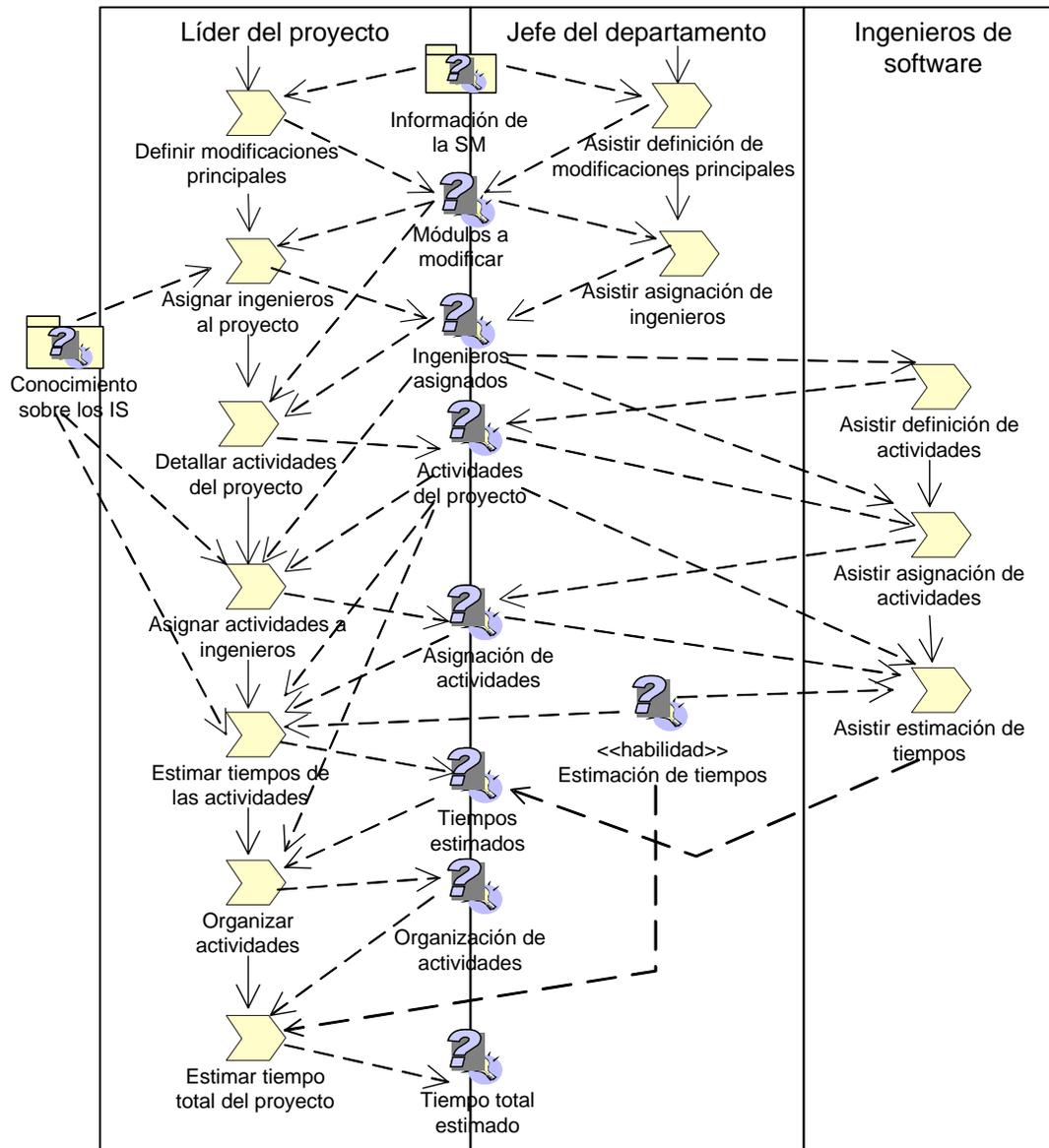
Transferencias de información y conocimiento

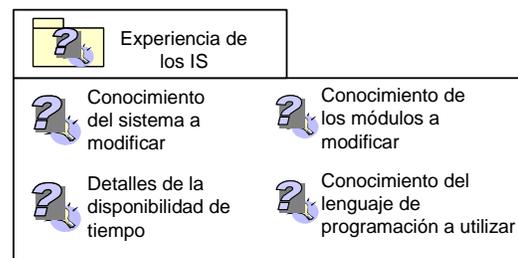
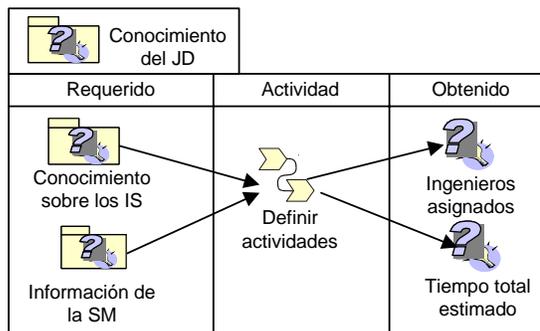
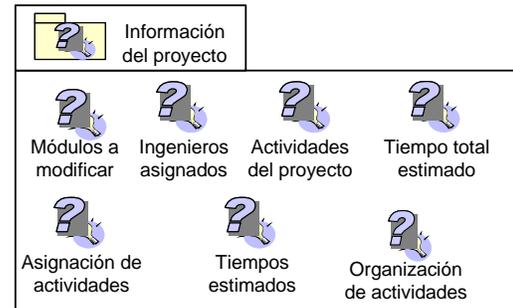
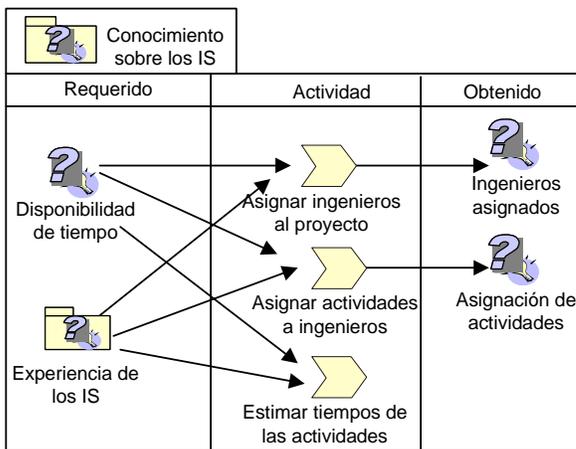
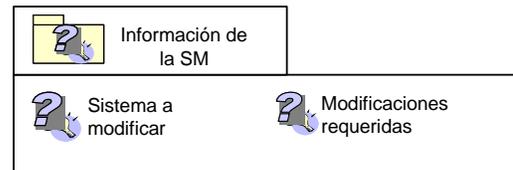
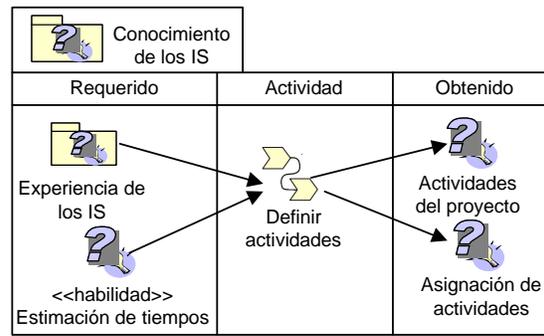
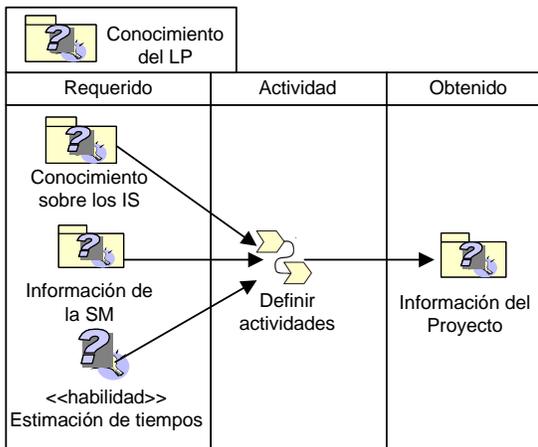


A.6.3 Ejercicio segundo grupo (Ejercicio 1b)

En el caso del ejercicio del segundo grupo, lo único que varió fueron los diagramas de la sección “Definir actividades”, por lo que sólo estos diagramas son mostrados.

Definir actividades





A.6.4 Cuestionario final

Nombre: _____

Ejercicio: [] 1a [] 1b

Procedimiento:

1. Con base en los modelos mostrados por los diagramas, contesta las series de preguntas que se presentan.
2. Contesta las preguntas en el orden en que aparecen, sin saltar ninguna.
3. Antes de iniciar la lectura de cada serie de preguntas, apunta el tiempo de inicio (hh:mm).
4. Al finalizar de contestar cada serie de preguntas, apunta el tiempo (hh:mm).

I.- Primera serie

Tiempo inicial: _____

- 1.- ¿Cuáles son las fuentes de información o conocimiento que participan dentro del proceso?
- 2.- ¿Qué roles están involucrados en el proceso?
- 3.- ¿Qué documentos específicos son utilizados o generados durante el proceso?
- 4.- ¿Cuál es el rol principal dentro del proceso?
- 5.- ¿Cuál es la principal fuente de información utilizada o generada durante el proceso?

Tiempo final: _____

II.- Segunda serie

Tiempo inicial: _____

- 6.- ¿En qué actividades específicas participa el jefe del departamento?
- 7.- ¿En qué actividades o flujos de trabajo se generan fuentes de información?
- 8.- ¿Qué actividades debe realizar el líder del proyecto?
- 9.- ¿Para qué actividades puede ser útil el plan del proyecto?
- 10.- ¿En qué actividades se requiere la solicitud de mantenimiento?

Tiempo final: _____

III.- Tercera serie

Tiempo inicial: _____

- 11.- ¿En general, qué requiere saber el líder del proyecto para desempeñar sus actividades?
- 12.- ¿A qué fuentes puede acudir el líder del proyecto para obtener la información y conocimiento que requiere para realizar sus actividades?
- 13.- ¿Qué información se puede obtener del plan del proyecto?
- 14.- Además de lo anterior, ¿Qué información puede proporcionar la documentación del proyecto?
- 15.- ¿Qué conocimientos de los ingenieros de software es útil para el proceso?
- 16.- ¿Qué debe considerar el líder del proyecto para decidir qué ingeniero se hará cargo de cada actividad del proyecto?
- 17.- ¿Qué información se considera para estimar el tiempo que llevará cada actividad?

Tiempo final: _____

IV.- Cuarta serie

Tiempo inicial: _____

- 18.- ¿Qué actividades generan información o conocimiento que es utilizado en otras actividades? Menciona también la información o conocimiento que es generado.
- 19.- ¿Qué actividades generan información o conocimiento que es útil para los ingenieros de software? Menciona también qué información o conocimiento.
- 20.- ¿Qué conocimiento obtiene el líder del proyecto al participar en el proceso?
- 21.- ¿Qué actividades requieren el intercambio de información o conocimientos entre los participantes en el proceso?
- 22.- Excluyendo las mostradas en los dos últimos diagramas, toma una de las actividades anteriores y describe los roles participantes, el conocimiento que aportan y obtienen, y en qué actividad se da esta transferencia (toma como base los dos últimos diagramas)

Tiempo final: _____

V.- Describe el proceso mostrado por los diagramas. Trata de ser breve, pero considerando las actividades principales, los principales conocimientos e información requeridos y generados, y las fuentes donde estos son obtenidos o almacenados.

VI.- Lee la siguiente descripción del proceso y contesta las preguntas que se hacen posteriormente

Nombre del proceso: Definición del plan de modificaciones

Descripción:

Cuando una solicitud de mantenimiento le es entregada al jefe del departamento (JD), éste debe asignar un líder del proyecto (LP) para que se haga cargo del proyecto de modificaciones. El líder del proyecto es elegido de entre los ingenieros de software, tomando en cuenta su experiencia con el sistema, y en específico los módulos que serán modificados, y en la coordinación de grupos de trabajo.

Ya que el JD ha elegido al LP, entre ambos llevan a cabo la definición del proyecto de modificaciones. Lo primero es definir las principales modificaciones que deberán ser realizadas al sistema, y asignar los ingenieros que trabajaran en el proyecto.

La selección de los ingenieros es realizada por el LP junto con el JD, y depende de la carga de trabajo que cada ingeniero tiene asignado actualmente, y la experiencia con el sistema, en particular con los módulos que podrían requerir ser modificados. Posteriormente, el LP y los ingenieros detallan las actividades específicas que deberán de realizarse, la asignación de los ingenieros a las actividades se basa en la experiencia de los ingenieros y su disponibilidad de tiempo.

Una vez que las actividades han sido asignadas, los ingenieros y el LP estiman el tiempo que cada actividad puede tomar, y con base en estas estimaciones se organizan las actividades buscando reducir el tiempo total que requerirá el proyecto.

Cuando el plan de modificaciones ha sido definido, el LP escribe un reporte describiendo los detalles del proyecto, este reporte debe contener la descripción del proyecto, sus objetivos, el tiempo estimado que se requerirá, personal asignado, y recursos requeridos, como software, equipo de cómputo, etc. El LP es el encargado de estimar estos recursos, así como los principales posibles riesgos y sus posibles soluciones o alternativas a tomar en caso de que estos ocurran.

Para reducir las desviaciones en los tiempos estimados, el LP puede incrementar el tiempo requerido dependiendo de los riesgos previstos. Finalmente, el reporte es enviado al consejo de administración, el cual decide si el proyecto es aceptado, rechazado, o debe modificarse.

Asigna un valor a las siguientes frases siguiendo la escala proporcionada a continuación:

- 1 = Totalmente en desacuerdo**
- 2 = En desacuerdo**
- 3 = Ligeramente en desacuerdo**
- 4 = Neutral**
- 5 = Ligeramente de acuerdo**
- 6 = De acuerdo**
- 7 = Totalmente de acuerdo**

- 1.- ___ La descripción proporcionada se asemeja a la que escribiste
- 2.- ___ Los diagramas son suficientes para poder entender el proceso descrito
- 3.- ___ La descripción del proceso no se asemeja a lo que entendiste de los diagramas
- 4.- ___ La información proporcionada por los diagramas es insuficiente para poder entender el proceso que se trata de representar
- 5.- ___ Es fácil asociar los íconos de la nomenclatura utilizada con los elementos que representan
- 6.- ___ La información presentada por las distintas vistas es redundante
- 7.- ___ La variedad en los diagramas incrementan el entendimiento del proceso
- 8.- ___ La información representada en los distintos diagramas se complementa
- 9.- ___ La utilización de paquetes para separar el conocimiento de las fuentes, facilita la identificación del conocimiento involucrado en las actividades.
- 10.- ___ El tener conocimiento de UML facilita el entendimiento de los diagramas
- 11.- ___ Preferirías la utilización de otras técnicas de modelado de procesos para representar el tipo de información del ejemplo presentado en el ejercicio.
- 12.- ___ Los diagramas usados para modelar un proceso siguiendo el lenguaje de modelado son demasiado detallados.
- 13.- ___ Los diagramas generados con la nomenclatura pueden ayudar a entender cómo fluye el conocimiento y la información dentro de un proceso.
- 14.- ___ Se requieren diagramas más detallados para poder representar correctamente los flujos de conocimiento.
- 15.- ___ Los modelos ayudan a identificar las fuentes de información y conocimiento involucradas en el proceso

16.- La nomenclatura o los tipos de diagramas usados en los ejercicios son:

	Confusa				Clara			
Nomenclatura	1	2	3	4	5	6	7	
Casos de uso	1	2	3	4	5	6	7	
Diag. de actividad	1	2	3	4	5	6	7	
Paquetes	1	2	3	4	5	6	7	
Transferencias de conocimiento	1	2	3	4	5	6	7	
Estructura de fuentes y conocimientos	1	2	3	4	5	6	7	
Todo en conjunto	1	2	3	4	5	6	7	

Nota: Para los siguientes calificativos se usó el mismo esquema anterior

Fácil de entender
Insuficiente
Difícil de aprender
Fácil de seguir
Rebuscada
Completa
Innecesaria

Difícil de entender
Suficiente
Fácil de aprender
Difícil de seguir
Adecuada
Incompleta
Necesaria

Con respecto a las siguientes preguntas, ordena las selecciones que hagas poniendo 1 a la más importante, 2 a la siguiente, y así sucesivamente.

17.- Ordena los siguientes tipos de diagrama dependiendo de qué tan útiles los consideras para el entendimiento general del proceso

- ___ Casos de uso
- ___ Actividad
- ___ Relación entre las actividades y el conocimiento de las fuentes (paquetes)
- ___ Transferencias de conocimiento
- ___ Estructura de las fuentes y conocimientos (clases)

Nota: Para las siguientes preguntas se usó el mismo esquema anterior:

- 18.- ¿Qué diagramas te ayudaron a identificar las fuentes de información o conocimiento?
- 19.- ¿Qué diagramas te ayudaron a identificar el conocimiento utilizado y generado en las actividades?
- 20.- ¿Qué diagramas te ayudaron a identificar los flujos de conocimiento entre actividades?
- 21.- ¿Qué diagramas te ayudaron a identificar los flujos de conocimiento entre fuentes?
- 22.- ¿Qué diagramas te ayudaron a identificar los roles participantes en el proceso?

APÉNDICE B

Caso de Estudio: Análisis del proceso de mantenimiento del Departamento de Informática del CICESE

B.1 Introducción

En este apéndice se presentan los resultados del caso de estudio realizado para analizar el proceso de mantenimiento de software del Departamento de Informática del CICESE (DI), desde la perspectiva del flujo de conocimiento en el proceso. Primero, se presenta la descripción general del proceso siguiendo el modelo general del mismo. Posteriormente se describen las principales observaciones y posibilidades de mejora en el proceso, identificando algunos requerimientos para un sistema de soporte al mismo.

Cabe destacar que este estudio sólo se enfocó en la atención de reportes de problemas en el uso de los sistemas mantenidos por el DI, y en la atención de solicitudes de cambios a los sistemas y módulos ya existentes. Este proceso gira en torno a un sistema de control de proyectos usado dentro del DI, que fue analizado para identificar el papel de dicho sistema en el flujo del conocimiento (como ya se ha descrito en el Capítulo VI). En este sentido, las posibilidades de mejora están enmarcadas en la manera en que dicho sistema, y su uso dentro del proceso, pueden ser modificados para mejorar el flujo del conocimiento dentro del proceso.

B.2 Descripción del proceso de mantenimiento en el Departamento de Informática del CICESE

Esta sección describe el proceso de mantenimiento de software que se lleva a cabo en el DI. Primeramente se describen los principales actores y roles involucrados en el

proceso. En seguida se describe el proceso en términos generales, para después detallar cada uno de los flujos de trabajo principales. Durante esta descripción se presentan los modelos del proceso que fueron desarrollados por medio de las adaptaciones a SPEM propuestas en el Capítulo IV.

B.2.1 Roles en el proceso

Según Polo et al. (1999), los roles de un proceso de mantenimiento pueden agruparse en tres grupos: los clientes, los usuarios, y los miembros del grupo de mantenimiento. Debido a que tanto los usuarios como los clientes del DI comúnmente son las mismas personas, hemos catalogado a los actores principales que participan en el proceso de mantenimiento del DI en dos grupos, los usuarios y los miembros del grupo de mantenimiento, al que denominaremos staff.

B.2.1.1 Usuarios

Los *usuarios* son los que hacen uso de los sistemas mantenidos, y por ende, aquellos a quienes el DI presta servicio. Dentro del esquema observado en el DI, los usuarios pueden ser catalogados en dos subtipos, los operativos y los administrativos.

Los *usuarios operativos* son aquellos que hacen uso del sistema pero no tienen ingerencia directa en la evolución del mismo.

Los *usuarios administrativos* tienen ingerencia en la evolución del sistema dado que son aquellos que pueden hacer solicitudes de cambios o mejoras al sistema. Ya sea solicitando que se corrijan errores, se agreguen nuevas funcionalidades, o se modifiquen las ya existentes.

Se ha definido también un tipo especial de usuario denominado *Usuario de pruebas*, el cual es un miembro del departamento que hace las veces de usuario para probar los sistemas una vez que ya han sido probados por el técnico encargado de modificarlos.

Si tomamos en cuenta que comúnmente al rol encargado de hacer las solicitudes de cambios o mejoras al staff de mantenimiento se le denomina cliente (Polo et al., 1999), podemos considerar que los usuarios administrativos juegan dos roles, el de usuarios, y el de clientes. Siguiendo las definiciones de roles de Polo et al. (1999), los usuarios administrativos también desempeñan los siguientes roles:

- *Solicitante*, dado que se encarga de hacer solicitudes de cambios al grupo de mantenimiento. En el estudio hacemos distinción de dos tipos de solicitantes, *solicitante de atención*, que reportan problemas que no involucran cambios en el sistema, y *solicitante de cambios*, que realiza solicitudes que implican cambios en el sistema. En el modelo de Polo et al. (1999), las actividades del solicitante de atención son asumidas por el usuario, mientras que el solicitante, sería equivalente al solicitante de cambios de nuestro modelo.
- *Organización del sistema*, definido como la organización que tienen un buen conocimiento sobre el sistema que va a ser mantenido, y es responsable de proporcionar los requerimientos de las modificaciones. En el proceso del DI, las actividades de este rol son asumidas por el solicitante de cambios.
- *Soporte*, que son los encargados de atender a los usuarios. En nuestro caso, este rol es denominado *encargado de atención*, y puede ser desempeñado tanto por los usuarios administrativos, como por los técnicos y la secretaria del DI.

B.2.1.2 Staff

Los miembros del staff del DI pueden dividirse en cuatro tipos, jefe del departamento, secretaria, técnicos y becarios.

B.2.1.2.1 Jefe de departamento

El *jefe del departamento* (JD) es responsable del control general del proceso de mantenimiento, por lo que es equivalente al rol de cabeza del mantenimiento definido por Polo et al. (1999), el cual se encarga de preparar la etapa de mantenimiento, esto involucra

establecer estándares y procedimientos a seguir dentro de la metodología o proceso de mantenimiento usados.

El JD es de gran importancia en las etapas iniciales de un proyecto grande de mantenimiento. Debe estar al tanto del estado de todos los proyectos. Participa, junto con los técnicos responsables, en la definición de requerimientos en casos que impliquen grandes cambios en el sistema. Por ejemplo, modificaciones en el proceso al que da soporte el sistema. A su vez, el JD apoya a los técnicos responsables en la definición de estos proyectos y en la planeación de los mismos. Está al tanto de que los proyectos sean realizados en tiempo y forma, y de interactuar con los usuarios solicitantes de los cambios para dar reportes de avances.

Entre las actividades del JD, también se encuentran el analizar las fallas ocurridas durante el desarrollo de los distintos proyectos, con el fin de proponer nuevos métodos o formas de trabajo que permitan reducir dichos problemas en proyectos futuros.

Para el caso del presente estudio, veremos al JD como *gestor de proyectos*, rol que debe encargarse del seguimiento y administración del conjunto de proyectos siendo desempeñados dentro del departamento, así como la obtención de los recursos requeridos para poder realizarlos.

B.2.1.2.2 Técnicos

Los *técnicos* son los ingenieros que trabajan en el DI. Ellos se encargan de desempeñar todas las labores relacionadas con el mantenimiento de los sistemas. Los técnicos a su vez juegan distintos roles dentro del proceso, tales como:

Responsable de sistema, ya que cada técnico está a cargo de un sistema específico, siendo responsable de las solicitudes de cambios que se hagan con respecto a ese sistema particular. Se podría considerar que este rol engloba al *gestor de proyecto*, dado que el responsable del sistema también está a cargo de gestionar cada proyecto definido para modificar dicho sistema. Siguiendo los roles definidos por Polo et al. (1999), el responsable

de sistema, también es el encargado de la calendarización de las actividades de mantenimiento en un sistema o proyecto particular, y de la administración de las solicitudes de mantenimiento de dicho sistema.

Ingenieros de requisitos, dado que son los encargados de obtener los requerimientos para la realización de las modificaciones a los sistemas de los que están a cargo.

Ingenieros de mantenimiento, dado que son los encargados de analizar y definir las modificaciones a los sistemas, así como encargarse de su realización.

Ingenieros de pruebas, ya que se encargan de probar que los cambios no hayan introducido nuevos errores, y que las modificaciones realizadas hayan cumplido con los requerimientos que las causaron.

Programadores, debido a que son los encargados de implementar las soluciones dadas a los problemas o modificaciones hechas en los sistemas.

A su vez, existen técnicos que también desempeñan los roles de *administradores de bases de datos y de archivos*, así como de *instructor de becarios* o nuevos miembros del equipo de desarrollo.

El *administrador de base de datos* es el encargado de hacer los cambios en las bases de datos de los sistemas mantenidos por el DI.

El *administrador de archivos* es el que se encarga de actualizar los archivos en el ambiente de trabajo de los usuarios, una vez que los cambios han sido realizados y probados. Debido a esta responsabilidad, el técnico que desempeña este rol también desempeña los roles de *Gestor de versiones*, dado que es el encargado de que las versiones en el ambiente de ejecución de los usuarios sean las correctas; y el de *Gestor de respaldos*, dado que es el responsable de mantener respaldados de los distintos sistemas mantenidos por el DI.

El rol de *instructor* es desempeñado por uno de los miembros del DI cuando ingresa algún becario, o un nuevo miembro al DI. La persona que desempeña este rol se encarga de informar a éstos sobre las técnicas, procesos y herramientas utilizadas para el desarrollo y mantenimiento de software en el DI. Así mismo, este rol les informa a los nuevos miembros sobre posibles fuentes de información que puedan serles de ayuda en sus actividades. Principalmente, estas fuentes son de información técnica, por ejemplo, el lenguaje de programación a usar, las herramientas de desarrollo, etc.

B.2.1.2.3 *Secretaria*

El DI cuenta entre sus miembros con una *secretaria* que apoya al resto del staff en la atención de ciertas solicitudes de los usuarios. En caso de no poderlas atender, las turna al técnico responsable. Así mismo, la secretaria también apoya en actividades de prueba de los sistemas, en particular, actúa como usuario de pruebas.

B.2.1.2.4 *Becario*

Finalmente, el DI en ocasiones cuenta con los servicios de *becarios*, que son personas que desempeñan labores para apoyar a los técnicos en diversas actividades. Principalmente, los becarios trabajan como *programadores*, implementando soluciones a solicitudes específicas del técnico responsable del sistema al cual el becario pueda estar asignado.

B.2.1.2.5 *Estructura de los roles*

La Figura 48 muestra una vista de la estructura de roles del proceso.

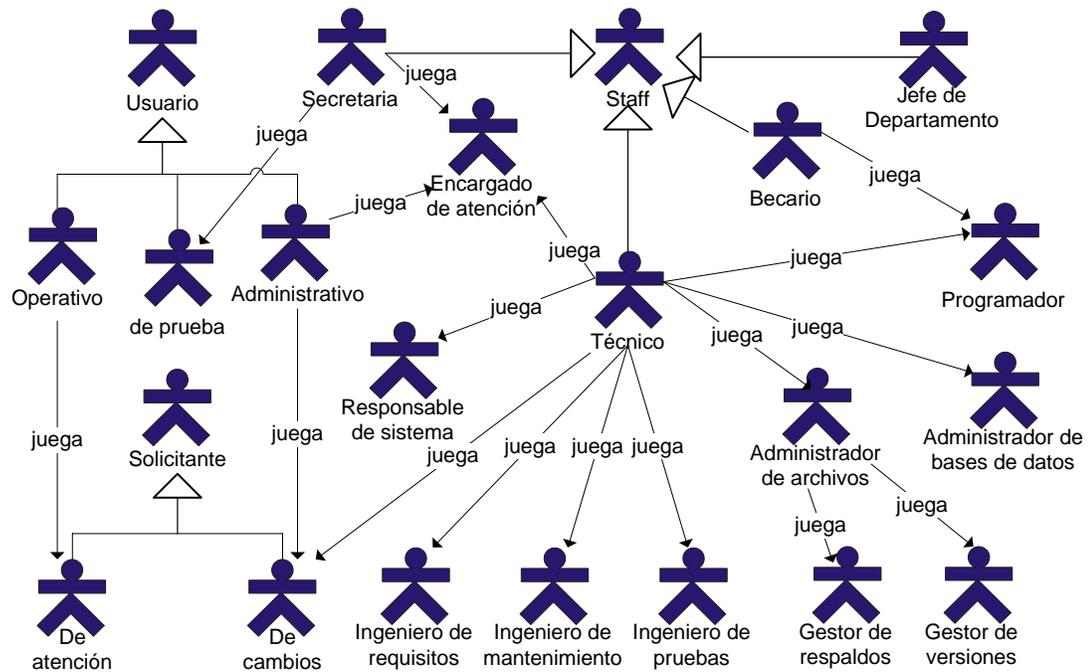


Figura 48. Vista de las relaciones de los distintos roles participantes en el proceso.

B.2.2 Descripción general del proceso

La Figura 49 muestra una vista general del proceso de mantenimiento del DI. Como se puede observar, el proceso consta de tres fases principales: atención a usuarios, implementación de cambios, y liberación de las versiones modificadas del sistema. A continuación se detalla cada fase.

B.2.2.1 Fase de atención a usuarios

La fase de atención a usuarios se ha dividido en dos subprocesos o flujos de trabajo: la atención de reportes de problemas, y la atención de solicitudes de cambio. Las actividades realizadas durante esta fase han sido asignadas a cuatro tipos de roles, como se muestra en las Figuras 48 y 49. En la Figura 50 se muestran los roles que interactúan en cada flujo de trabajo, mientras que en la Figura 51 se presentan también las actividades que cada rol desempeña en esta fase. Un aspecto importante es que durante esta fase del

proceso, en particular durante la atención de solicitudes de cambios, se genera la solicitud de cambios, la cual es uno de los principales documentos o fuentes de información que intervienen en el proceso.

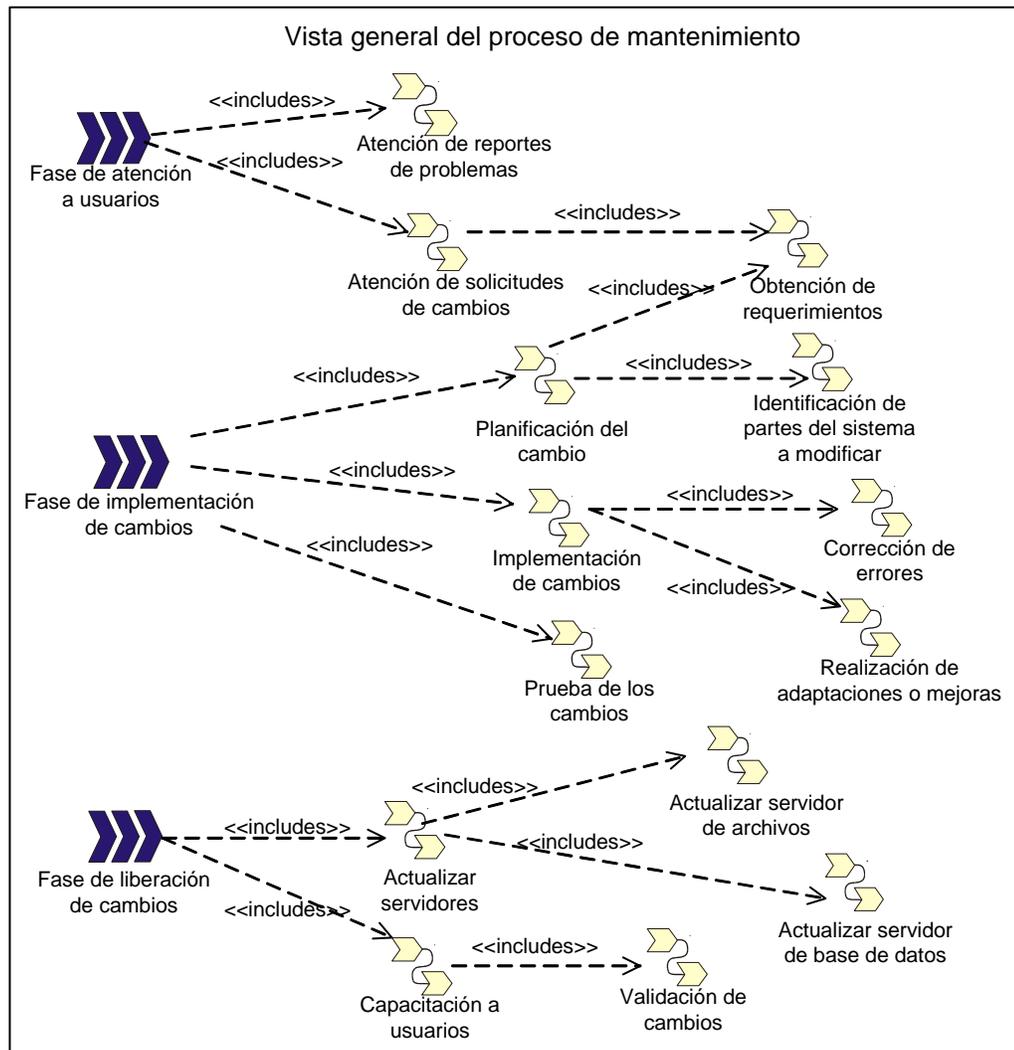


Figura 49. Vista general del proceso de mantenimiento del DI.

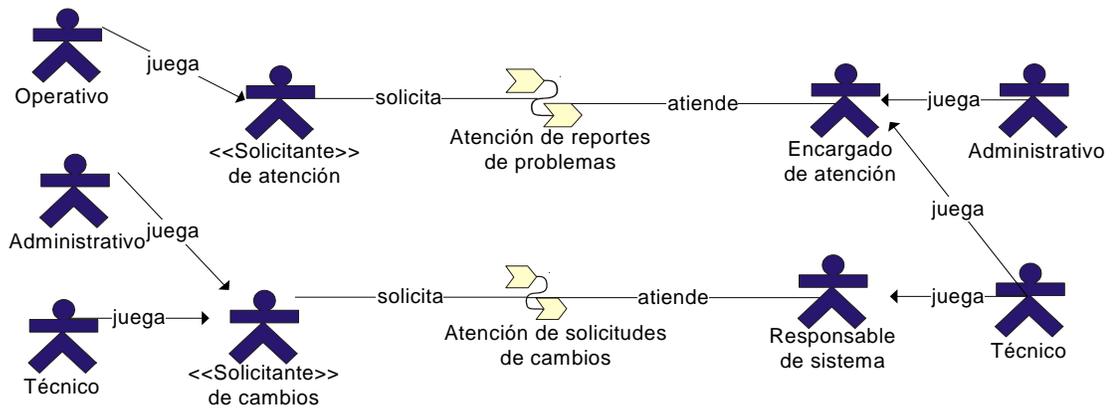


Figura 50. Caso de uso de la fase de atención a usuarios. Muestra los roles que interactúan en los flujos de trabajo que se realizan en esta fase.

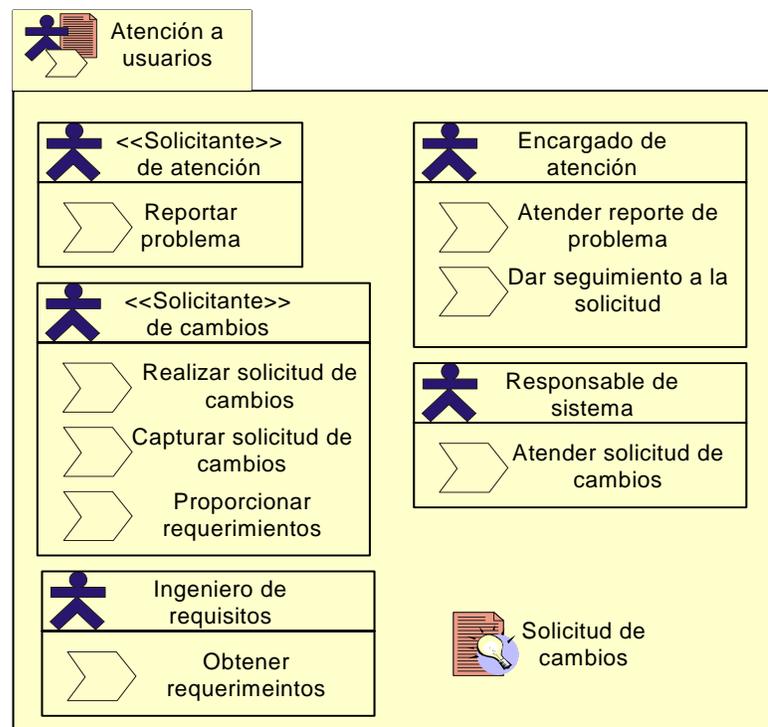


Figura 51. Diagrama de paquetes de la fase de atención a usuarios. Muestra las actividades que desempeña cada rol que participa en dicha fase, así como los principales artefactos que son usados o generados durante la misma.

B.2.2.1.1 Atención de reportes de problemas

La Figura 52 muestra la secuencia de actividades que se da durante la atención de reportes de problemas. Los reportes de problemas son situaciones que no ameritan cambios en el sistema. Por lo general se dan debido a un mal uso del sistema por parte de los usuarios, a errores con el equipo de cómputo o software que no tienen relación directa con el sistema en sí, pero que podría afectar el uso del mismo. Un ejemplo puede ser una mala configuración de la impresora, de tal forma que no se puedan imprimir los reportes proveídos por el sistema.

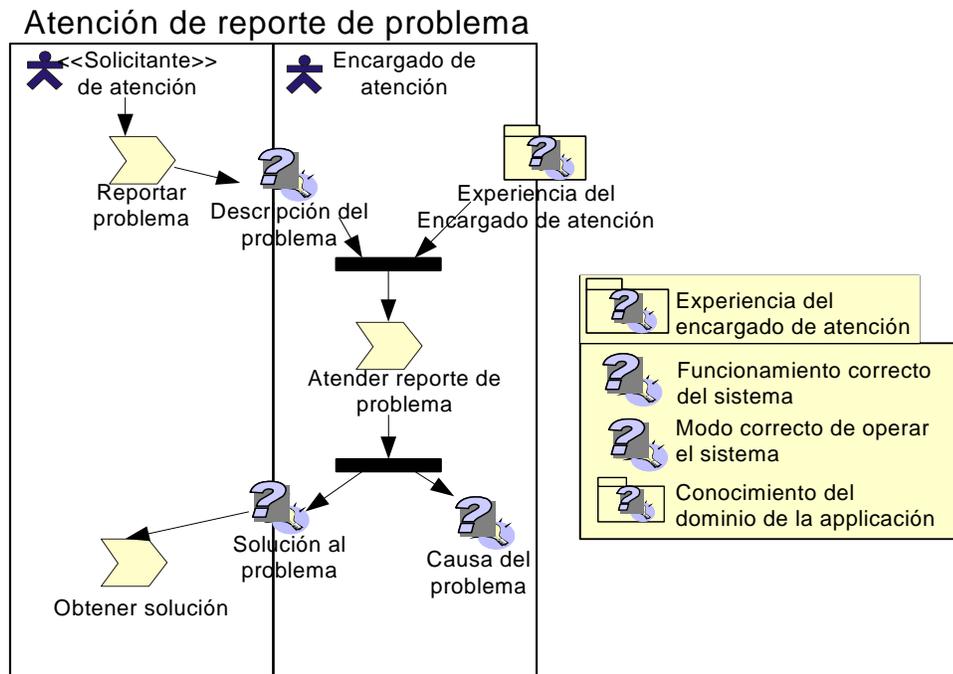


Figura 52. Secuencia de actividades durante el flujo de trabajo de atención de reporte de problema.

Descripción del proceso. Cuando un usuario se topa con un problema, se dirige con el encargado de atención que le corresponde, con el fin de informarle el problema, y solicitarle una solución al mismo. El encargado de atención analiza el problema, y define cuál puede ser la causa. Si conoce la solución, se la proporciona al usuario, para que éste

pueda continuar con sus funciones. En caso contrario, dirige la petición a alguna otra persona que pudiera ayudar a dar solución a la misma. En estos casos, si el problema no está directamente relacionado con alguno de los sistemas mantenidos por el DI, es posible que la solución deba provenir del departamento de Telemática del Centro, por ejemplo, si es un problema relacionado con los servicios de red, con el hardware, etc. En estos casos, el proceso de atención de la solicitud queda fuera del ámbito del DI por lo que no se analizarán estos casos en el presente documento.

Variantes. La atención de problemas es realizada, por lo general, por los usuarios administrativos, que son los usuarios que tienen un mayor conocimiento de la forma en que el sistema debe funcionar, o la forma de uso correcta. Sin embargo, en ocasiones la secretaria y los técnicos también desempeñan el rol de encargados de atención. Cuando los usuarios se comunican con un usuario administrativo, o con la secretaria, éstos últimos tratan de resolver el problema, en caso de no poseer los conocimientos suficientes para resolverlo, dirigen al solicitante con el técnico encargado del sistema para que sea éste quien le resuelva el problema al usuario.

La atención de usuarios se da principalmente de dos formas: por vía telefónica, donde los usuarios se comunican directamente ya sea con algún usuario administrativo, la secretaria del DI, o con el técnico encargado; o por interacción cara a cara.

El DI busca que sea el menor número de reportes de problemas los que lleguen a los técnicos, ya que la atención de los mismos puede generar retrasos en los proyectos de mantenimiento o desarrollo a los que están asignados los miembros del grupo.

Fallas detectadas. La atención a usuarios es realizada principalmente con base en la experiencia de la persona encargada de llevarla a cabo. Por lo general, este tipo de solicitudes no quedan registradas, por lo que es difícil saber la cantidad de solicitudes que se atienden en un período específico, la frecuencia con que se presentan, las causas de las mismas, la forma en que se solucionaron, etc.

Con frecuencia, se reportan problemas comunes que tienen una solución bien definida, pero no hay una descripción de esta solución que esté accesible a los usuarios, o éstos no la conocen. Por ejemplo, en algunos casos, el mismo sistema identifica ciertos errores comunes de los usuarios, y envía mensajes informativos para ayudarle al usuario a realizar sus actividades correctamente. Este caso es común en el llenado de datos en ciertas formas de los sistemas mantenidos por el DI. Sin embargo, no todos los problemas pueden ser abordados de esta manera. Además, debido a que no se lleva un registro de este tipo de solicitudes, no es fácil determinar cuáles son los problemas más comunes, salvo por la misma experiencia de los técnicos o personas involucradas en la atención de usuarios.

B.2.2.1.2 Atención de solicitudes de cambio

La Figura 53 presenta la secuencia de actividades involucradas en la atención de solicitudes de cambio. Como es posible observar, son tres los roles que participan, el encargado de solicitar los cambios, el responsable del sistema a modificar, y el responsable de la obtención de los requerimientos.

Durante este flujo de actividades se genera una de las principales fuentes de información del proceso: la solicitud de cambios. Esta solicitud se genera en el Sistema de Control de Proyectos (SCP) que maneja el DI. Por lo tanto, el SCP ha sido considerado como parte esencial en el flujo del conocimiento en este proceso, dado que es el que almacena las solicitudes, además de que también participa como parte activa en el proceso, al ser este sistema el encargado de informar a los responsables de los sistemas mantenidos por el DI, de las nuevas solicitudes capturadas por los usuarios, como se muestra en la Figura 53.

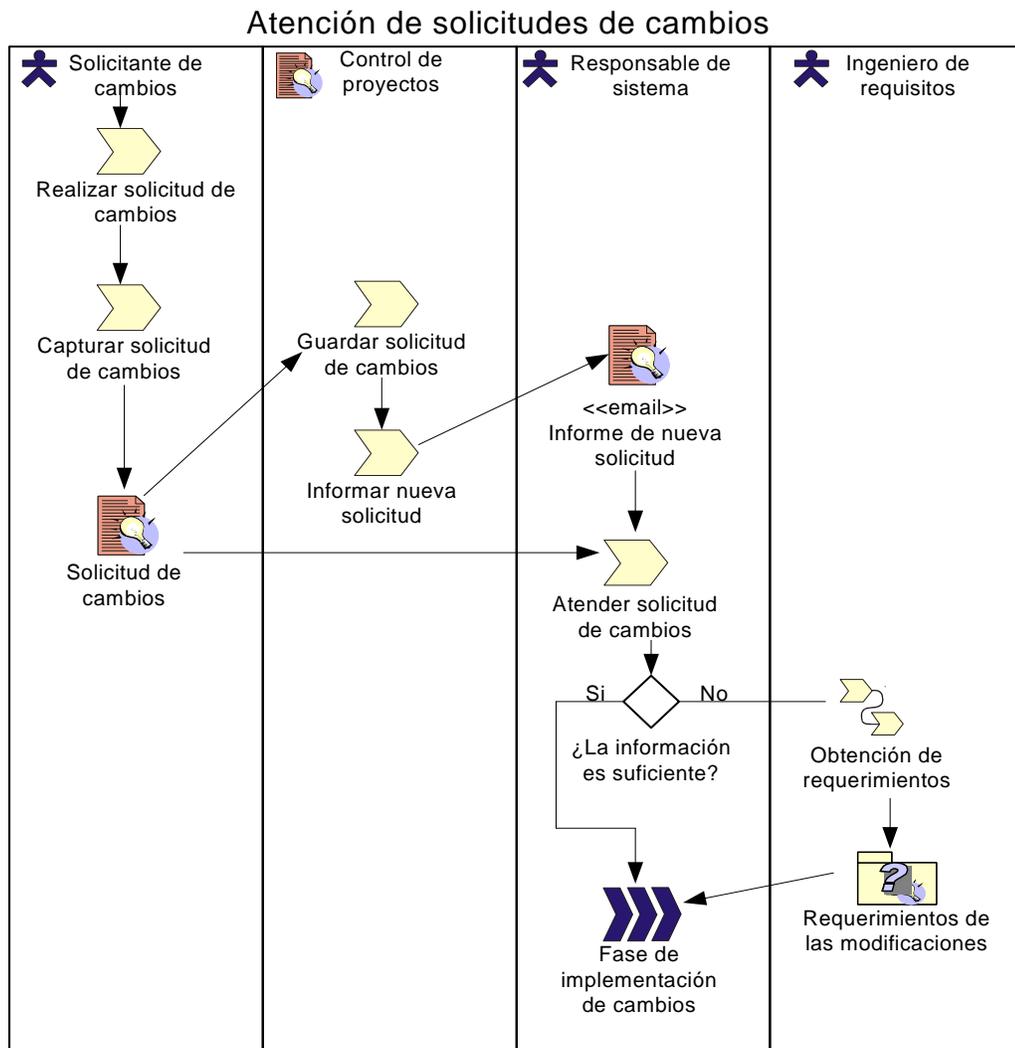


Figura 53. Secuencia de actividades del flujo de trabajo atención de solicitudes de cambios.

Descripción del proceso. Cuando un usuario requiere que se haga algún cambio en el sistema, éste ingresa al SCP. En él, primero indica el sistema donde se requiere el cambio. Por ejemplo, en la Figura 54 se ha elegido el sistema de apoyo a la Dirección de Estudios de Posgrado (DEP).

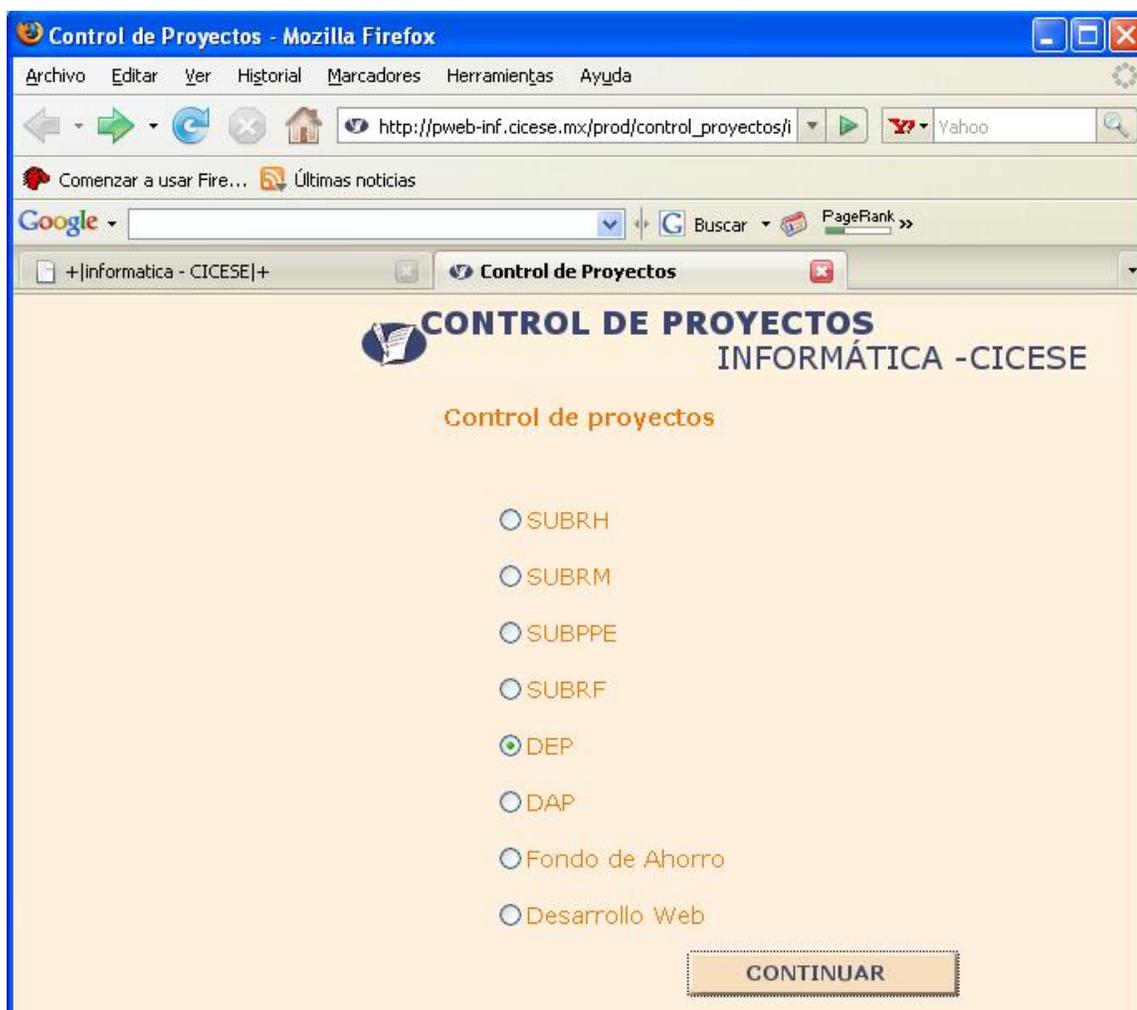


Figura 54. Página de inicio del sistema de Control de Proyectos del DI.

Una vez elegido el sistema donde se solicitará el cambio, el SCP presenta una pantalla con información para dar seguimiento a las distintas solicitudes que se han hecho sobre el sistema elegido. Para agregar una nueva solicitud, sólo hay que presionar la opción “Solicitud” en el menú principal en la parte superior de la ventana, o en el botón “Realizar Solicitud” en la parte inferior de la misma, como se muestra en la Figura 55.

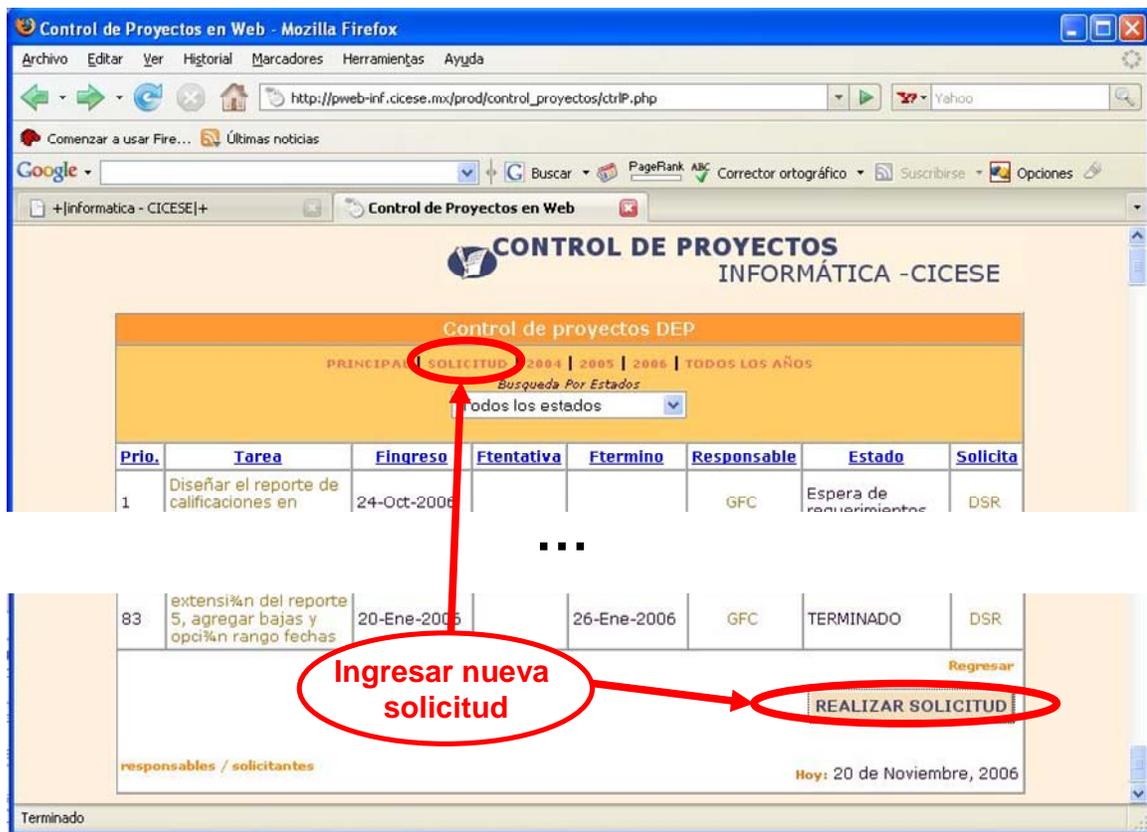


Figura 55. Ventana de seguimiento de solicitudes en el Sistema de Control de Proyectos.

Al presionar cualquiera de las dos opciones, el sistema presenta una forma para la captura de los datos de la solicitud, tal como se muestra en la Figura 56. En esta forma, se asigna un nombre descriptivo a la solicitud, y se da una descripción de la misma. Se indica el solicitante del cambio, y la persona responsable de los mismos. El sistema proporciona de antemano una lista de las personas que pueden ser solicitantes o responsables de los cambios solicitados, los cuales corresponden a los usuarios administrativos del sistema, el jefe del DI, y el técnico responsable del sistema en cuestión. Finalmente, es posible asignar la prioridad en la atención de la solicitud, así como las observaciones que se consideren pertinentes.

Control de Proyectos DEP

PRINCIPAL | PROYECTOS

*Nombre de la tarea: (Por favor utilice un nombre corto y distintivo)

*Descripción de la tarea: (Por favor describa la tarea con lujo de detalle; evite el uso de apóstrofes o comillas [','])

*Solicita: (seleccione al solicitante)

Seleccione al solicitante

Responsable: (seleccione al responsable)

Seleccione al responsable

*Prioridad: **ver tareas existentes**

(coloque la prioridad en relación a las tareas existentes, si desea que su tarea tenga mayor prioridad que alguna de las actuales, haga favor de consultarlo con el solicitante y agregarlo en el campo observaciones) ENTRE MAS PEQUEÑO EL NÚMERO MÁS ALTA LA PRIORIDAD

Observación:

LIMPIAR ENVIAR

* Son campos requeridos

La sección de detalle se va a actualizar en el transcurso del día.

Cualquier duda comuníquese con **Gabriela Flores Campos** ext: 22127 ó **Karin Buckle Lopez** 22120

Última actualización: 12 de Octubre del 2005, Departamento de Informática.

Figura 56. Ventana para la captura de los datos de la solicitud.

Una vez capturados los datos de la solicitud, se presiona el botón de enviar para que esta sea almacenada en el SCP. Posteriormente, el SCP envía un correo electrónico al técnico responsable del sistema en cuestión, con el fin de informarle de la nueva solicitud.

Cuando el técnico responsable decide atender la solicitud, éste determina si es necesario obtener más información para poder llevar a cabo los cambios solicitados. Con base en la información proporcionada, el técnico determina qué cambios se deben hacer, dónde y cómo. Si la información proporcionada no es suficiente para lo anterior, se procede a la obtención de requerimientos más detallados para proceder con la realización de los cambios.

Variantes. Es importante decir que no todas las solicitudes se ingresan al sistema. En ocasiones, estas solicitudes se envían por medio de un correo electrónico, o por medio de un memorando, aun cuando el proceso actualmente definido por el DI establece que se debe usar el SCP.

Otra variante que es importante tomar en consideración, es que las solicitudes de cambios pueden verse de dos formas, las comunes, que se realizan según el proceso ya definido, y las que implican grandes cambios en los sistemas. Solicitudes dentro de esta última categoría, conllevan un proceso más elaborado. En particular, este tipo de solicitudes incluyen un proceso de planeación en el que participan el jefe del departamento, el técnico responsable del sistema a modificar, y los usuarios del mismo. Se realizan reuniones entre los interesados para establecer los requerimientos generales de los cambios, posteriormente se determina un calendario y se estima el tiempo que llevará completar los cambios solicitados. Posteriormente se le informa a los interesados; si se está de acuerdo con dicho calendario, se procede con la inicialización del proyecto de mantenimiento, de lo contrario, se discute para llegar a un acuerdo, ya sea modificando los requerimientos, estableciendo entregas parciales, etc.

En este estudio nos hemos centrado en el proceso normalmente llevado a cabo, por lo que no ahondaremos en detalles con respecto a las solicitudes que impliquen este tipo de cambios. Sin embargo, en lo general, los procesos son similares, con la diferencia de que en el último caso se le da más formalidad al proceso de planeación del proyecto. Esto conlleva que la solicitud de modificaciones, en estos casos, normalmente sea un memorando enviado al jefe del DI, y firmado por el responsable del área que solicita los cambios. Una vez

establecido esto, la captura de los requerimientos específicos, y el desarrollo del proyecto se realiza de manera muy similar al resto de los proyectos.

Fallas detectadas. Los problemas principales que han sido detectados en la fase de atención de solicitudes de cambio son los siguientes:

1.- *Forma de captura de las solicitudes.* En la forma de captura de solicitudes en el SCP, el campo para definir al solicitante de un cambio presenta una lista de usuarios autorizados. Sin embargo, existen usuarios no dados de alta en el sistema que también pueden solicitar cambios. Debido a que estos no aparecen en el sistema, deciden no usarlo y en cambio envían sus solicitudes por otros medios, tales como el correo electrónico, un memorando, o por vía telefónica.

Otro problema relacionado con lo anterior, es que en ocasiones los usuarios que no están dados de alta usan el nombre de alguno de los usuarios registrados, por ejemplo, el jefe del área a la que pertenecen. Cuando el técnico responsable del sistema se comunica con la persona registrada como solicitante, resulta que esta última desconoce de dicha solicitud debido a que no fue realmente quien la hizo. Esto puede complicar el saber con exactitud quien realizó realmente una solicitud.

2.- *Envío de solicitudes.* Si bien el DI busca que todas las solicitudes de cambios sean enviadas por escrito, no todas se ingresan al SCP. Algunas solicitudes son enviadas vía correo electrónico, y otras a través de memorandos. Con frecuencia, los mismos técnicos tratan de ingresar estas solicitudes al SCP, pero no es así en todos los casos. Algunas solicitudes se quedan en los correos electrónicos, o en los memorandos; lo que puede dificultar hacer una recopilación de las solicitudes recibidas en determinado período, por ejemplo.

B.2.2.1.3 *Captura de requerimientos*

Descripción del proceso. La captura de requerimientos, como se muestra en la Figura 57, la realiza el técnico responsable del sistema contactando con el responsable de la

solicitud de cambios. Este contacto se puede dar en persona, o por vía telefónica. Si los cambios son grandes, la obtención de requerimientos puede representar una o varias sesiones de interacción cara a cara con los responsables de la solicitud de cambios.

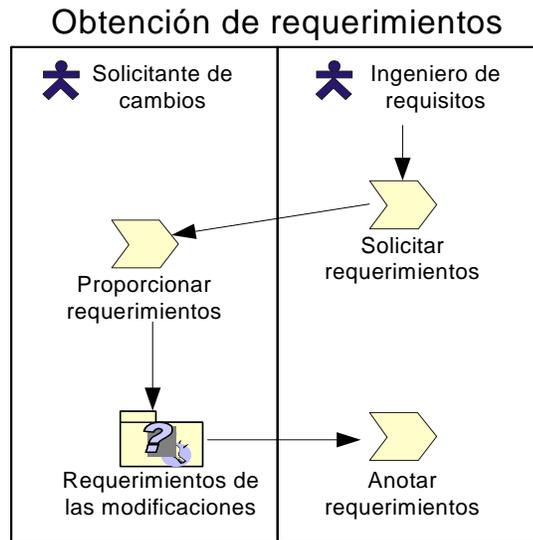


Figura 57. Secuencia de actividades de la etapa de obtención de requerimientos.

Variantes. La principal variante en el proceso de captura de requerimientos, depende de la magnitud del cambio a realizar. Si los cambios son relativamente pequeños, el técnico se encarga de llevar el proceso de captura de requerimientos con los usuarios administrativos del sistema.

Si la solicitud implica cambios mayores, por ejemplo, cambios en las reglas seguidas por el proceso soportado por el sistema, la definición de los requerimientos puede incluir a los jefes del área a la que da soporte el sistema. En estos casos, en el proceso de definición de requerimientos también se involucra al jefe del departamento, quien apoya al responsable del sistema en este proceso. Este caso conlleva una serie de reuniones entre los miembros del DI y los del área al que el sistema da soporte, en las cuales se definen los requerimientos generales a cubrir con las modificaciones. Una vez realizado esto, el técnico se encarga de obtener los requerimientos específicos para llevar a cabo dichos cambios.

Esto último se realiza siguiendo un proceso similar al llevado a cabo con el resto de las solicitudes de cambios.

Como se muestra en la Figura 58, la principal información generada en esta fase, es almacenada en las solicitudes de cambio, ya que estas contienen los datos proporcionados por el solicitante del cambio, y en ocasiones los requerimientos, o parte de ellos. En el caso de que la solicitud no contenga la totalidad de los requerimientos, estos son capturados por los técnicos en una libreta de trabajo al momento de solicitar la información detallada de los requerimientos de los cambios a realizar.

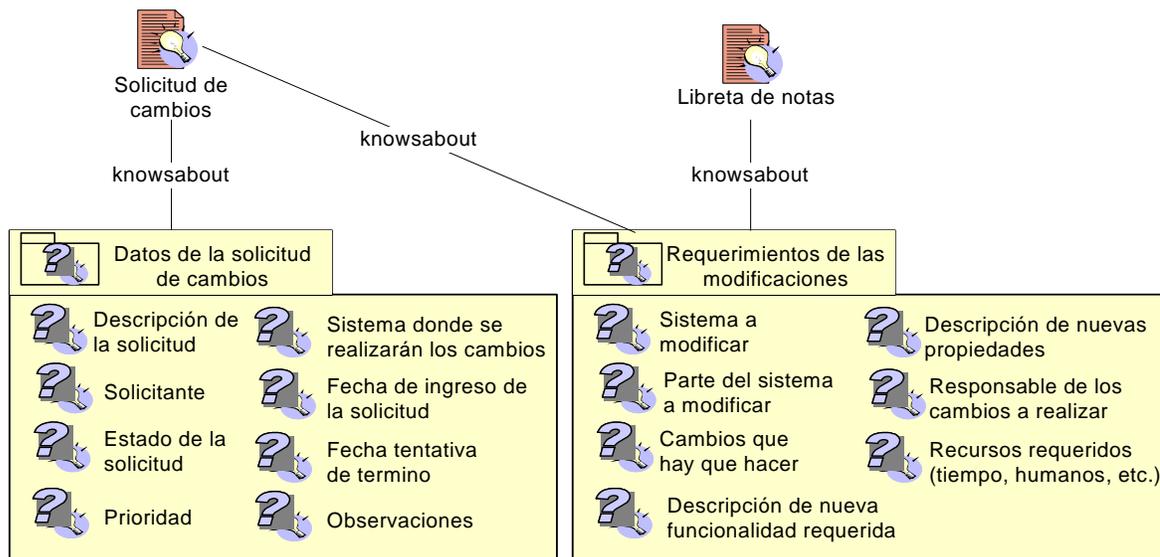


Figura 58. Información contenida en la solicitud de cambios.

Fallas detectadas. No existe un protocolo o proceso predefinido para la captura de requerimientos. Por lo tanto, no existe un estándar con respecto a la información que debe ser obtenida durante la captura de los requerimientos. Así mismo, no existe un proceso de documentación de requerimientos predefinidos. Por lo que la documentación de los requerimientos sólo queda escrita de manera informal como apuntes que realizan los técnicos en sus cuadernos de notas. Esto genera el riesgo de que una vez terminado el

cambio, y desechada la libreta, estos requerimientos se pierden, no pudiendo así, saber en su totalidad las causas que pudieran haber generado un cambio específico; o por ejemplo, medir la evolución de los sistemas en base a los requerimientos de sus modificaciones.

Así mismo, es difícil relacionar una solicitud de cambios con los requerimientos que corresponden con la misma. Para realizar esto, es necesario que el responsable del sistema recuerde en que lugar apuntó los requerimientos de una determinada solicitud. En ocasiones, los técnicos llevan el registro en sus libretas organizados por fechas, por lo que al obtener la fecha de una determinada solicitud, pueden saber desde que punto empezar a buscar los requerimientos que fueron anotados. Sin embargo, este es un proceso que lleva tiempo y no asegura que el técnico vaya a encontrar realmente los requerimientos que anotó en su libreta y que corresponden con dicha solicitud.

B.2.2.2 Fase de implementación de cambios

La fase de implementación de cambios está dividida en tres flujos de trabajo principales, como se muestra en la Figura 59. En el primero, el técnico juega los roles de responsable de sistema y de ingeniero de mantenimiento. Como responsable de sistema, se encarga de planificar la realización de los cambios. Como ingeniero de mantenimiento, se encarga de identificar las partes del sistema que deberán ser modificadas. Posteriormente procede a la implementación de los cambios, donde también actúa como programador. Finalmente, se pone en el papel de ingeniero de pruebas, encargado de verificar que los cambios hayan sido correctamente realizados.

En el modelo de la etapa de Implementación de cambios también se ha considerado la obtención de requerimientos, dado que si el técnico considera que la información que tiene no es suficiente para poder realizar los cambios, procede a la ampliación de la información de los requerimientos.

Asimismo, se han considerado dos opciones para la implementación de cambios: la corrección de errores, y la realización de adaptaciones o mejoras, dado que los procesos específicos para cada una de ellas varía. Sin embargo, en este documento no ahondaremos

en estas diferencias, dado que no detallaremos las actividades específicas en cada una de ellas, más bien, el objetivo es identificar las habilidades y conocimiento que requieren estas opciones de manera general.

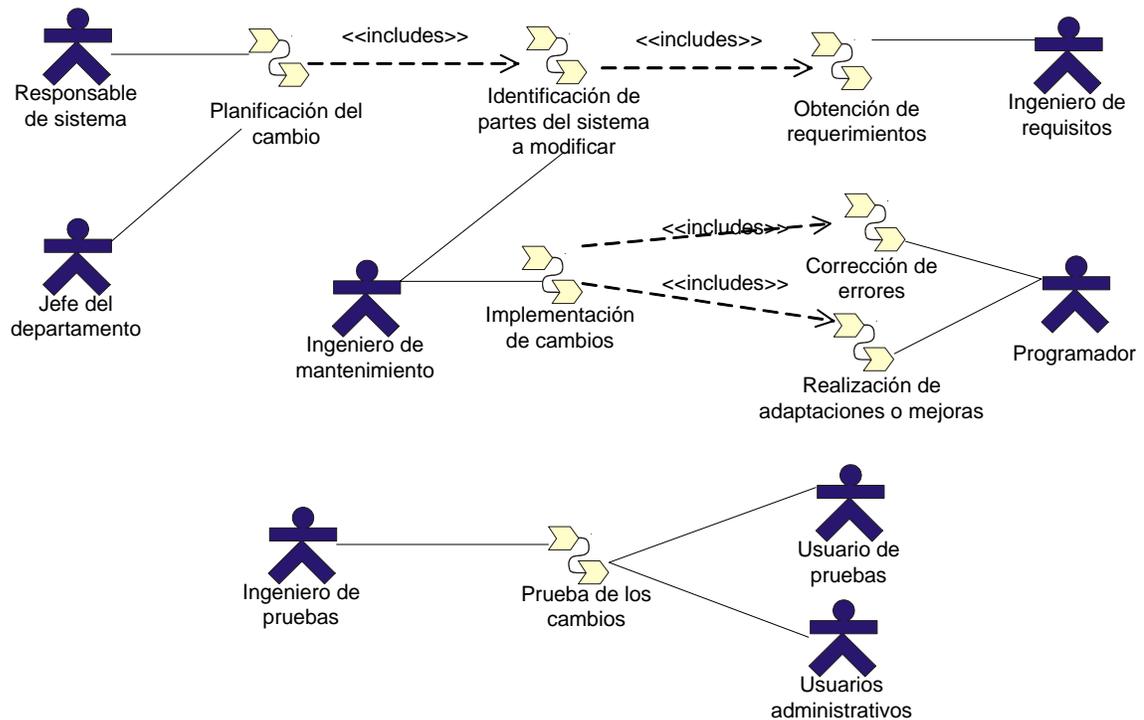


Figura 59. Casos de uso de la fase de implementación de cambios. Muestran los roles que interactúan en los flujos de trabajo que se realizan en esta fase.

La Figura 60 muestra las actividades que realiza cada rol durante la implementación de cambios, así como los artefactos que son usados y modificados durante las mismas. En particular, los artefactos involucrados son la solicitud de cambios, el código fuente de los módulos a modificar, y el sistema mismo.

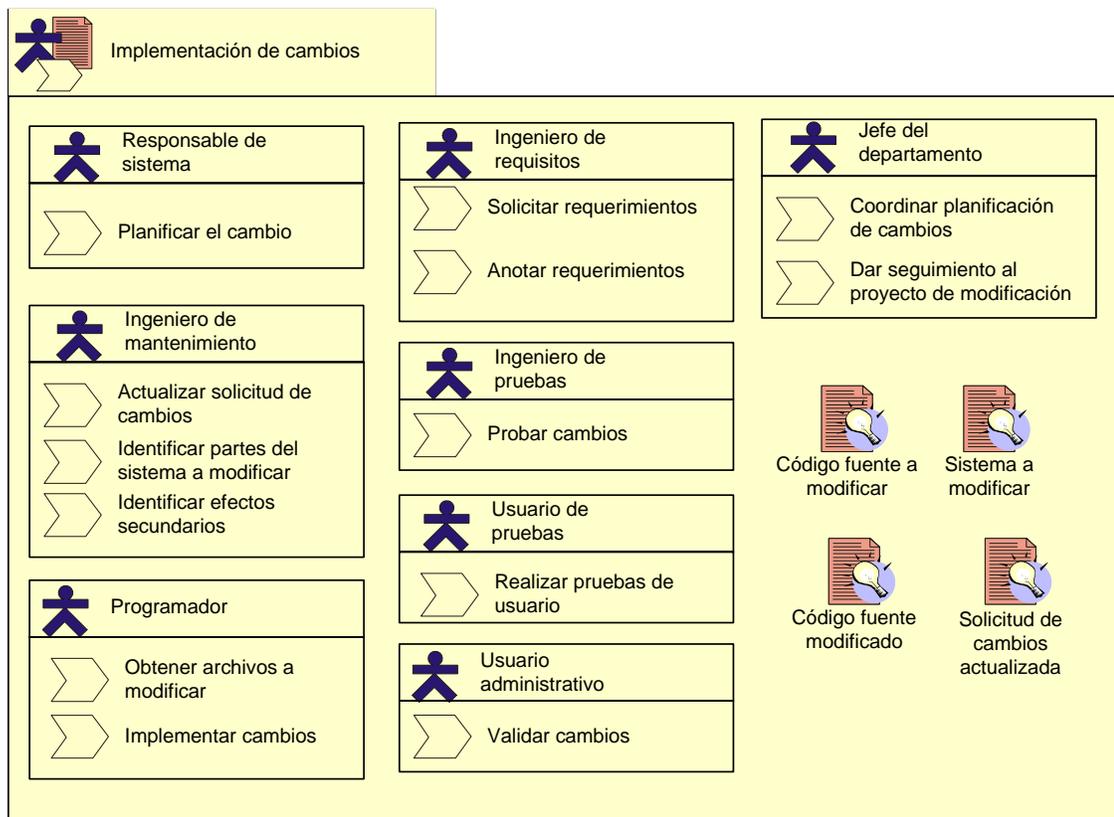


Figura 60. Diagrama de paquetes de la fase de implementación de cambios. Muestra las actividades que desempeña cada rol que participa en dicha fase, así como los principales artefactos que son usados, modificados o generados durante la misma.

Descripción del proceso. La Figura 61 muestra la secuencia de actividades de la fase de implementación de cambios. En este flujo de actividades, cuando el responsable de sistema decide dar solución a una solicitud, éste revisa la solicitud en el SCP. Este sistema muestra una lista de las solicitudes que han sido capturadas (ver Figura 62), de las cuales el ingeniero de mantenimiento elige aquella a la que le dará solución.

La elección de qué solicitud atender, se realiza siguiendo el orden de prioridad, el cual es un número consecutivo asignado a cada solicitud, donde 1 indica la de mayor prioridad, o la que deberá ser atendida primero, el 2 la que será atendida posteriormente, y así sucesivamente.

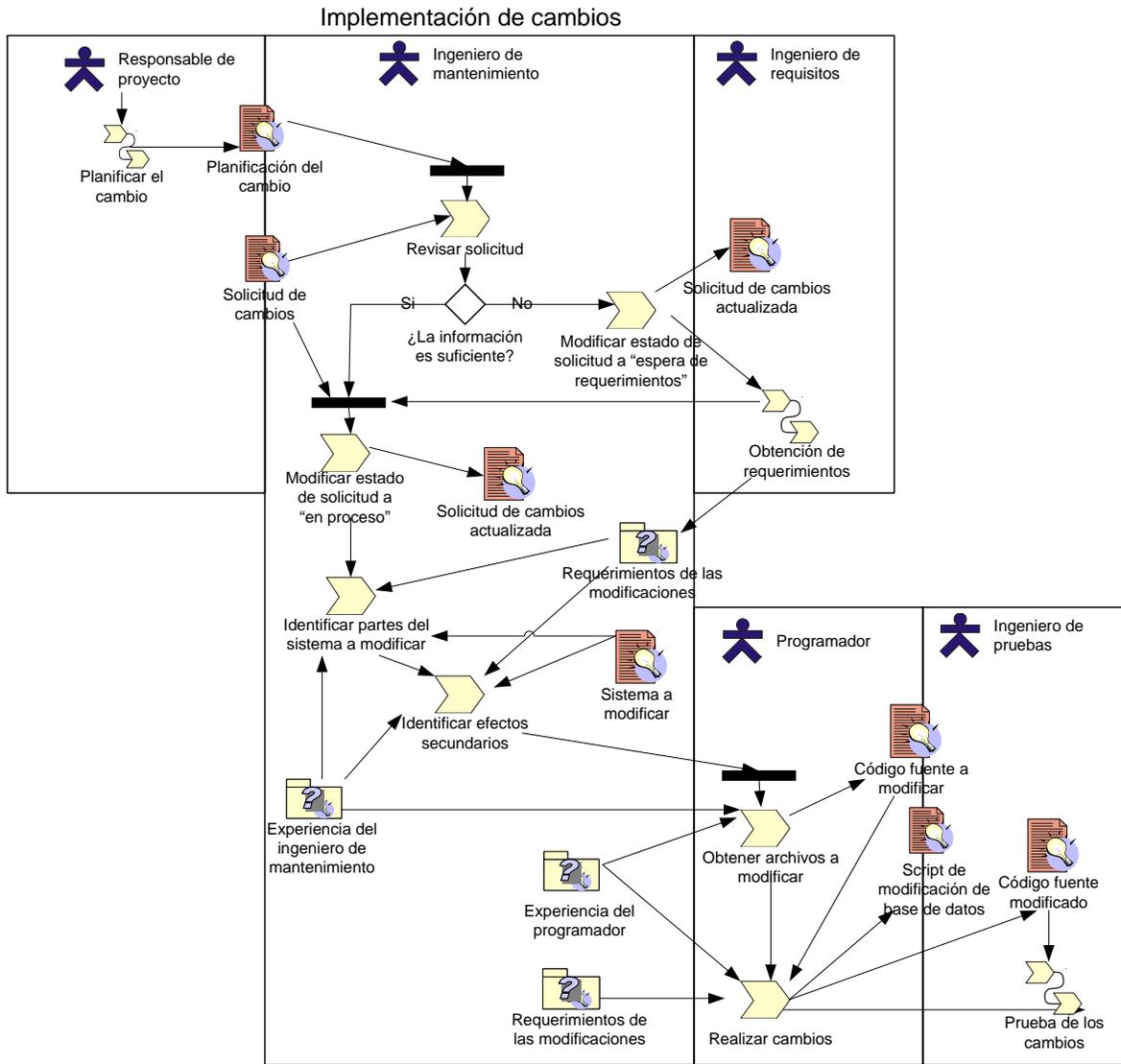


Figura 61. Secuencia de actividades de la fase de implementación de cambios.

Control de proyectos DEP							
PRINCIPAL SOLICITUD 2004 2005 2006 TODOS LOS AÑOS							
Busqueda Por Estados							
Todos los estados							
Prio.	Tarea	Fingreso	Ftentativa	Ftermino	Responsable	Estado	Solicita
1	Diseñar el reporte de calificaciones en Report	24-Oct-2006			GFC	Espera de requerimientos	DSR
2	Navegavilida entre el cardex y la tira de materia...	19-Oct-2006		19-Oct-2006	GFC	TERMINADO	CRM
3	Registro de estudiantes (c/s)	19-Oct-2006		19-Oct-2006	GFC	TERMINADO	CRM
4	Incluir mecanismo a registro de cursos (c/s)	19-Oct-2006		19-Oct-2006	GFC	TERMINADO	DSR

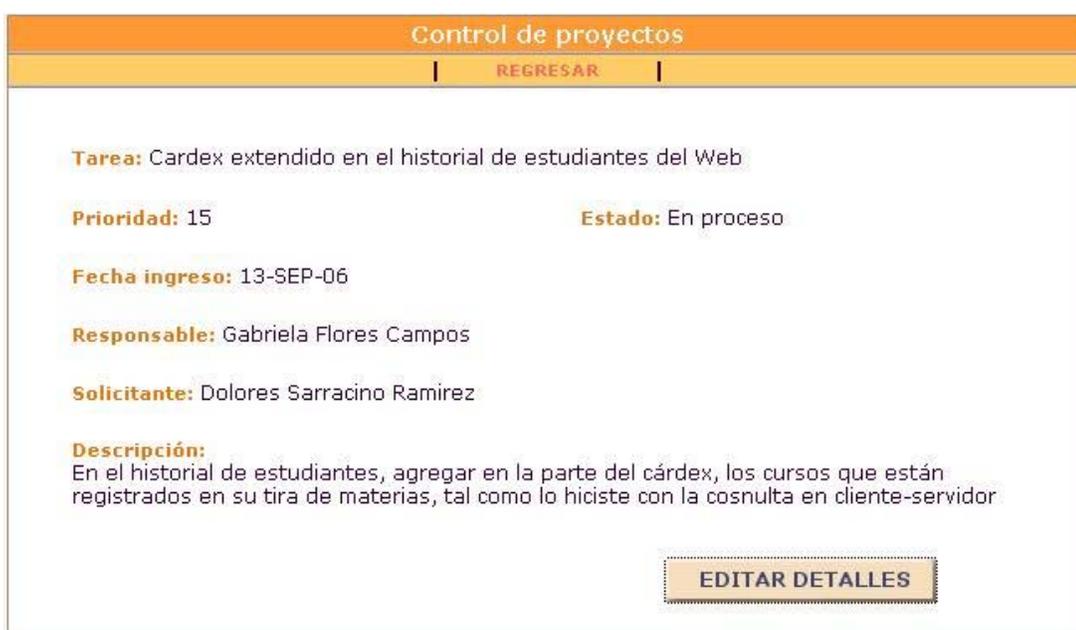
Figura 62. Ventana donde se muestra la lista de solicitudes capturadas en el SCP.

Una vez elegida una solicitud, el SCP muestra los datos de la misma, tal como se ejemplifica en la Figura 63.

Control de proyectos	
REGRESAR	
Tarea:	proceso automático para grabar cardex de egresados y bajas
Prioridad: 25	Estado: Espera de requerimientos
Fecha ingreso:	15-JUN-06
Responsable:	Gabriela Flores Campos
Solicitante:	Citlali Romero Manzano
Descripción:	Crear un proceso por medio del cual, cuando el estatus del estudiante cambie a egresado o bajas, se corra el proceso del promedio y se actualice automáticamente para que se quede grabado en el registro del estudiante. Recordar que el estatus cambia automáticamente a egresado con el avance de tesis es del 100%. De la misma forma cambia automáticamente a baja cuando se corre el proceso de bajas
<input type="button" value="EDITAR DETALLES"/>	

Figura 63. Ejemplo de solicitud de cambio con estado en Espera de Requerimientos.

Al revisar la solicitud, el ingeniero de mantenimiento determina si tiene la información suficiente para proceder con los cambios. En caso de no ser así, cambia el estado de la solicitud a “Espera de Requerimientos”, y procede a la obtención de requerimientos por parte del solicitante del cambio, tal como se mostró en el diagrama 7. En caso de que la información sea suficiente, o una vez que se han obtenido los requerimientos faltantes, entonces se procede a realizar los cambios, modificando el estado de la solicitud a “En Proceso”, la Figura 64 muestra un ejemplo de una solicitud en proceso.



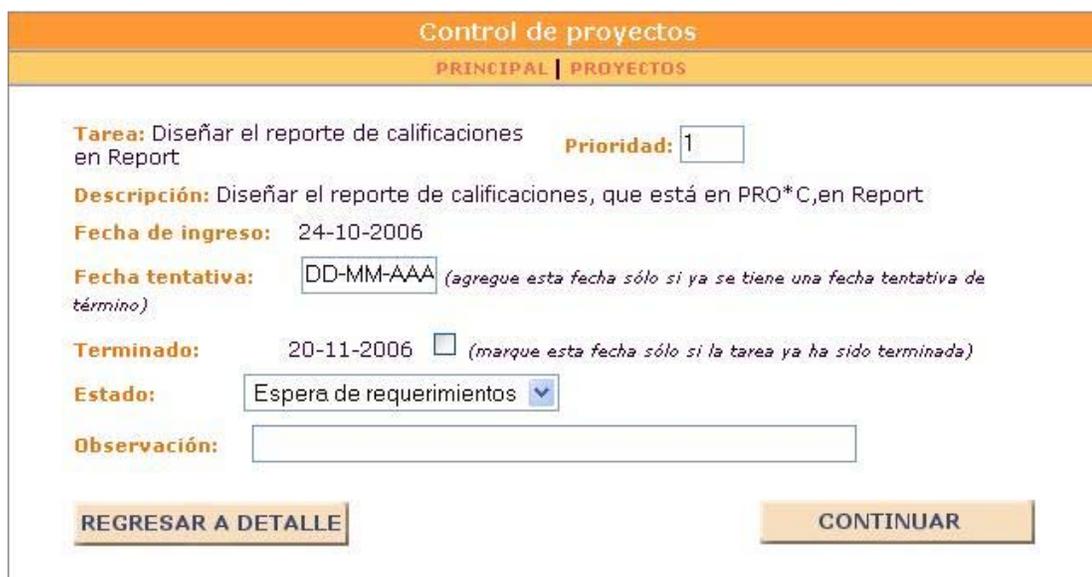
The screenshot displays a web interface titled "Control de proyectos". At the top, there is a navigation bar with a "REGRESAR" button. The main content area shows the following details for a request:

- Tarea:** Cardex extendido en el historial de estudiantes del Web
- Prioridad:** 15
- Estado:** En proceso
- Fecha ingreso:** 13-SEP-06
- Responsable:** Gabriela Flores Campos
- Solicitante:** Dolores Sarracino Ramirez
- Descripción:** En el historial de estudiantes, agregar en la parte del cárdex, los cursos que están registrados en su tira de materias, tal como lo hiciste con la cosnulta en cliente-servidor

At the bottom right of the form, there is a button labeled "EDITAR DETALLES".

Figura 64. Ejemplo de solicitud de cambio con estado En Proceso.

La Figura 65 muestra la ventana de edición de una solicitud. Se puede observar que se permite cambiar el estado de la solicitud, la prioridad, la fecha tentativa de término, así como capturar observaciones.



The screenshot shows a web-based interface titled "Control de proyectos" with a sub-header "PRINCIPAL | PROYECTOS". The main content area contains the following fields and controls:

- Tarea:** Diseñar el reporte de calificaciones en Report. **Prioridad:** 1 (input field)
- Descripción:** Diseñar el reporte de calificaciones, que está en PRO*C, en Report
- Fecha de ingreso:** 24-10-2006
- Fecha tentativa:** DD-MM-AAA (agregue esta fecha sólo si ya se tiene una fecha tentativa de término) (input field)
- Terminado:** 20-11-2006 (marque esta fecha sólo si la tarea ya ha sido terminada)
- Estado:** Espera de requerimientos (dropdown menu)
- Observación:** (text input field)

At the bottom, there are two buttons: "REGRESAR A DETALLE" and "CONTINUAR".

Figura 65. Ejemplo de la ventana de edición de solicitudes.

Una vez que el ingeniero de mantenimiento cuenta con la información de requerimientos suficiente para iniciar el proceso de implementación de cambios, lo primero que hace es identificar cuales son las partes del sistema que deberán ser modificadas, así como los posibles efectos secundarios que podrían implicar los cambios. Tales como cambios a otros módulos relacionados, o incluso a otros sistemas. La Figura 66 presenta los temas de conocimiento principales que integran el conocimiento del sistema.

Para identificar los posibles efectos secundarios, el técnico utiliza principalmente su experiencia. Sin embargo, cuando los cambios pueden involucrar a otros sistemas, se consulta con los técnicos a cargo de estos últimos, o con el jefe del departamento, con el fin de determinar qué tanto pueden afectar los cambios a realizar, y definir las estrategias a seguir en caso de que sea necesario modificar los sistemas que pueden ser afectados. Las modificaciones de los sistemas afectados son realizadas por los técnicos responsables de los mismos, y son atendidas como cualquier otra solicitud de modificaciones.

Una vez determinadas las partes del sistema que serán modificadas, el técnico procede a obtener los archivos fuente correspondientes que se encuentran en el servidor de

archivos. Cada técnico tiene una cuenta en dicho servidor, así como un espacio de trabajo en el mismo, donde se encuentran almacenados los archivos fuente del sistema que está a su cargo, y cualquier otra información relacionada con el mismo, tales como los documentos que puedan existir sobre ese sistema.

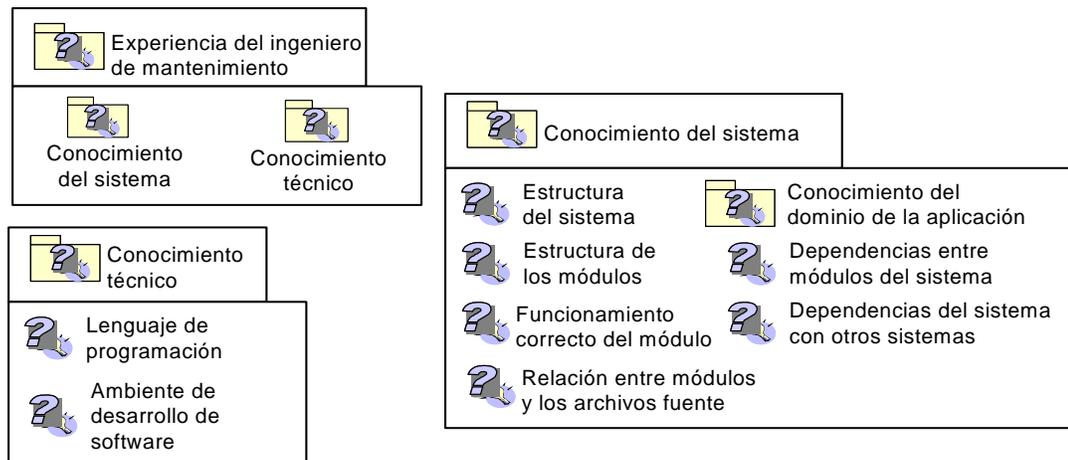


Figura 66. Diagrama del paquete que corresponde al conocimiento que se debe tener para poder identificar las partes del sistema a modificar, los posibles efectos secundarios de los cambios, y la forma de realizar los cambios en el sistema.

Para elegir los archivos a modificar, los técnicos deben saber qué partes del sistema corresponden con qué archivos. Por lo general, esta información la obtienen de su propia experiencia y conocimiento. En caso de no ser ésta suficiente, la principal técnica que utilizan es analizar el código fuente del sistema para identificar los archivos que corresponden al modulo a modificar. Este análisis se hace siguiendo la secuencia de menús del sistema, e identificando los archivos que son llamados en cada opción.

Otra fuente con la que pueden contar, son los diccionarios de datos que les permite obtener el ambiente de desarrollo. Se detectó que en algunos casos existe un documento que describe la relación entre los distintos módulos del sistema, y los archivos fuente correspondientes. Este documento es consultado por el encargado del sistema en cuestión

cuando debe modificar alguna parte del sistema de la cual no sabe, o no recuerda qué archivos fuente le corresponden.

Si los cambios requieren modificaciones en la base de datos, los técnicos las realizan en el área de pruebas (o desarrollo como se le conoce en el DI). Estas modificaciones por lo general implican la creación de un script para realizar dichos cambios, el cual es usado posteriormente en la fase de liberación. La Figura 67 muestra la información que puede ser obtenida de este script.

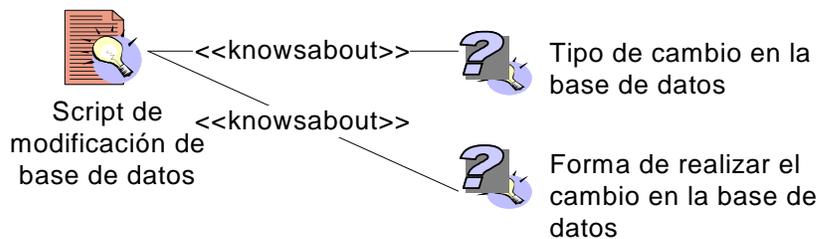


Figura 67. Información que puede ser obtenida del script para modificar la base de datos.

Variantes. El orden de prioridad de una actividad, puede ser determinado por el solicitante de las modificaciones, sobre todo, puede determinar entre las solicitudes que han realizado, cuáles son las que quiere que se atiendan primero. Este orden también puede ser modificado por el responsable del sistema, si considera que hay solicitudes que deben ser atendidas antes que otras con mayor prioridad.

Otra razón para modificar el orden de prioridad, es cuando no se cuenta con la suficiente información para resolver alguna solicitud. Por ejemplo, que se encuentre en estado de espera de requerimientos, por lo que para ahorrar tiempo, se procede con las solicitudes que sí pueden ser atendidas en un momento determinado.

Por lo general, la implementación de los cambios corre por cuenta de los técnicos responsables del sistema que será modificado. Sin embargo, hay casos en los que estos pueden ser apoyados por becarios u otros técnicos en la realización de los cambios. En

estos casos, el responsable del sistema debe indicarles cuáles cambios hacer, y proporcionarles la información necesaria (requerimientos, archivos fuente, etc.). Una vez realizadas las modificaciones, el responsable del sistema se encarga de recopilar los cambios realizados por los programadores de apoyo, y determinar si los módulos modificados pasan a la fase de liberación.

Fallas detectadas. Las siguientes son las principales fallas detectadas durante el proceso de implementación de cambios.

1.- *Prioridad de las solicitudes.* El orden de prioridad de las solicitudes no se determina siguiendo un proceso formal, se deja a la opinión de los solicitantes de los cambios, quienes determinan cuáles consideran que deberán ser atendidas primero. Si es un solo usuario el habilitado para solicitar cambios a un determinado sistema, éste podrá determinar sin problemas cuáles son más importantes para él. Sin embargo, en sistemas donde son varios los usuarios habilitados para solicitar modificaciones, la definición de prioridades se puede complicar bastante, ya que para un usuario, sus solicitudes serán más importantes que las del resto, por lo que podría determinar que sus solicitudes tienen mayor prioridad que las de los demás.

Por otro lado, si existen solicitudes que se están aplazando porque llegan otras con mayor prioridad, las primeras podrían quedarse eternamente en espera, aún cuando pudiera ser importante darles atención. De hecho, se detectaron solicitudes realizadas mucho tiempo atrás, que han ido quedando rezagadas, esto puede ocasionar que con el tiempo el responsable del sistema se olvide de atenderlas, ya que siempre estarán llegando nuevas solicitudes, y los usuarios que las realicen difícilmente estarán de acuerdo en que se agreguen hasta el final de la lista, sobre todo cuando estas listas son largas.

2.- *Registro de los cambios realizados.* Si bien algunos técnicos del DI capturan parte de las actividades que hicieron para atender una determinada solicitud, esto no está definido como parte del proceso. Algunos técnicos suelen usar el campo de observaciones para especificar lo que implicó atender una determinada solicitud. Otros llevan registro en sus cuadernos de

notas, mientras que otros llevan registros muy detallados de sus actividades en archivos almacenados en sus computadoras.

Sin embargo, la información asociada a dichas solicitudes es insuficiente, sobre todo en lo concerniente al trabajo y esfuerzo realizado por cada técnico al atenderlas. Por ejemplo, no es fácil conocer el tiempo invertido en cada solicitud, las sesiones de captura de requerimientos que requirió, cuánto tiempo se invirtió en definir éstos requerimientos, qué documentos podrían estar asociados con los requerimientos (por ejemplo, nuevos procedimientos); qué efectos secundarios implicaron los cambios realizados, qué partes específicas del sistema debieron ser modificadas, cuánto tiempo se invirtió en la implementación de los cambios, cuánto en las pruebas de los mismos; si se generaron o modificaron documentos relacionados, etc.

3.- *Documentos de descripción de la estructura de los sistemas.* Un aspecto importante del proceso de implementación de cambios, es determinar qué partes del sistema requerirán ser modificadas, y qué archivos fuente corresponden con dichas partes del sistema. Por lo general esto se realiza en base a la experiencia de los encargados de hacer los cambios.

Se detectó que debido al tiempo que requiere el análisis de un sistema para obtener esta información, algunos de los técnicos han generado documentos que describen la estructura de módulos del sistema y su relación con los archivos fuente que los implementan. Lo que les ayuda a ahorrar tiempo al identificar los archivos fuente que corresponden con ciertas partes del sistema. Estos documentos son de uso personal de cada técnico, y por lo general el resto no conoce a cerca de ellos, por lo que si un técnico falta, difícilmente los documentos generados por el mismo podrán ser usados por otros técnicos que lo suplan.

Por otra parte, no hay un estándar en la forma en que se desarrolla este tipo de documentos, cada técnico lo hace según su entender y necesidades particulares. Siendo este tipo de documentos tan importantes, deben ser considerados como parte del proceso, definiéndose un estándar para generarlos, y mantenerlos lo más actualizados posibles, y a la

vez, almacenarlos en algún espacio compartido para el resto del equipo de desarrollo, de manera que cualquier persona que requiera hacer cambios a un determinado sistema, pueda acceder a los documentos de descripción de la estructura del mismo.

B.2.2.3 Etapa de pruebas

Una vez realizados los cambios, se procede con las pruebas de los mismos. La Figura 68 presenta el flujo de actividades que se da durante la prueba de los cambios. Las pruebas se realizan en un área llamada ambiente de desarrollo, utilizando una copia de las bases de datos de los sistemas que se encuentran en operación. Esto con el fin de no introducir errores en los datos reales.

Descripción del proceso. Una vez realizados los cambios, los archivos modificados se suben al ambiente de desarrollo donde se prueba el sistema tomando los datos de la base de datos de prueba, la cual contiene una versión no actualizada de los datos reales. Los sistemas por lo general son probados por los técnicos responsables de los mismos, en lo que podría catalogarse como pruebas del código (de unidad e integración). Sin embargo, existen casos en que intervienen otras personas en la etapa de pruebas, particularmente, durante la etapa de pruebas de aceptación y validación.

En algunos casos, puede requerirse que se realicen pruebas por parte de un Usuario de pruebas (esto podría considerarse como la técnica de pruebas “actúa como usuario”). El objetivo del usuario de pruebas es jugar con el sistema con el fin de identificar si existen fallas que pudieran haber sido pasadas por alto por el técnico responsable de los cambios. Este tipo de pruebas de usuario se realizan en aquellos sistemas que tienen un gran número de usuarios, por lo que no es posible tener una certeza sobre la manera en que estos intentarán usar el sistema. En los casos de los sistemas cuyos usuarios están bien localizados, y su número es reducido, por lo general no se realizan estas pruebas, debido a que estos usuarios por lo general operan el sistema de la manera en que se les solicita que lo hagan.

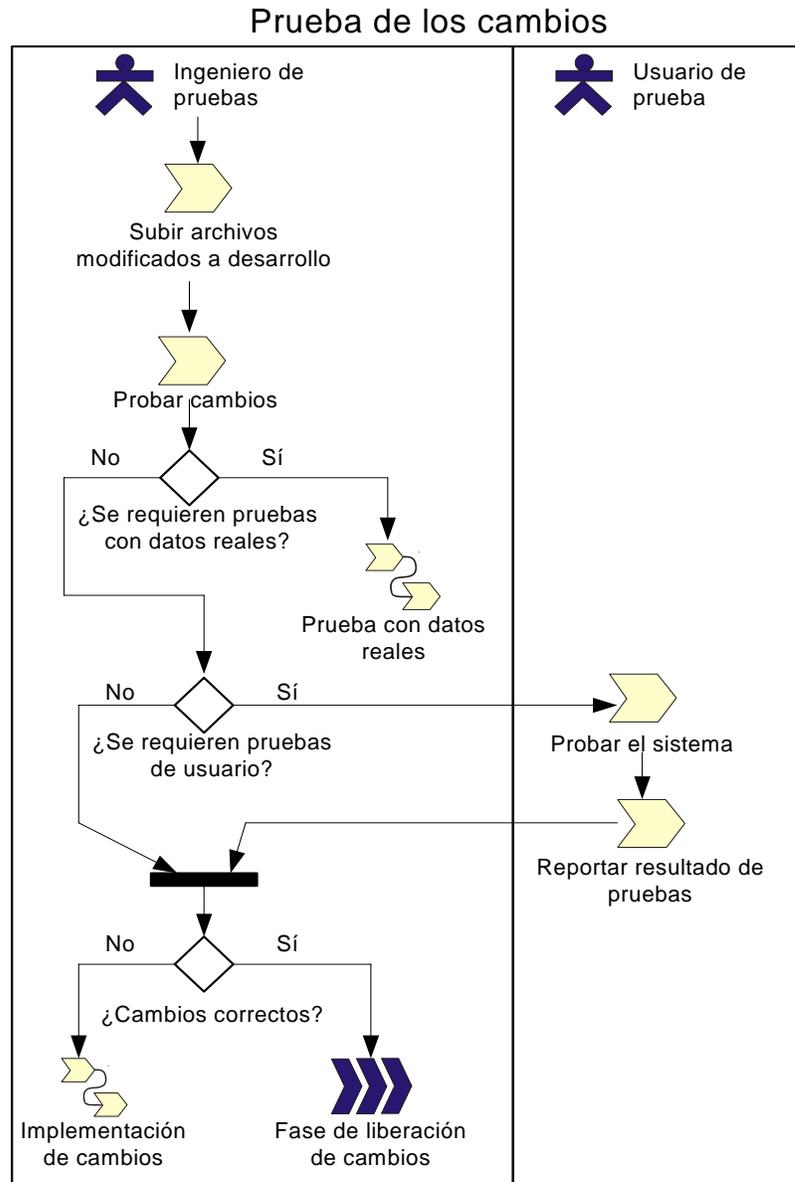


Figura 68. Diagrama de actividad del flujo de trabajo de las pruebas de los cambios.

Existen también casos en que es necesario hacer una prueba con los datos reales, es decir, los que se encuentran en el área de producción, la cual es a la que tienen acceso los sistemas en operación. Para esto se hace una copia de la base de datos en el área de pruebas, tal como lo muestra la Figura 69. Este proceso de copiado de la base de datos es solicitado por el técnico responsable del sistema donde se requieren hacer las pruebas. La solicitud se

hace por medio de un correo electrónico enviado al administrador de base de datos. Cuando éste último ha hecho la copia, se lo informa al técnico solicitante.

Una vez realizada la copia, el técnico responsable solicita a los usuarios solicitantes del cambio que participen en las pruebas, con el fin de que éstos validen que los cambios realizados fueron los solicitados, y que el sistema funciona correctamente. Este proceso podría considerarse como pruebas de validación. Sin embargo, como se mencionó, este tipo de pruebas no se realiza en todos los caso. De hecho, sólo se realizan para el caso del sistema de recursos humanos.

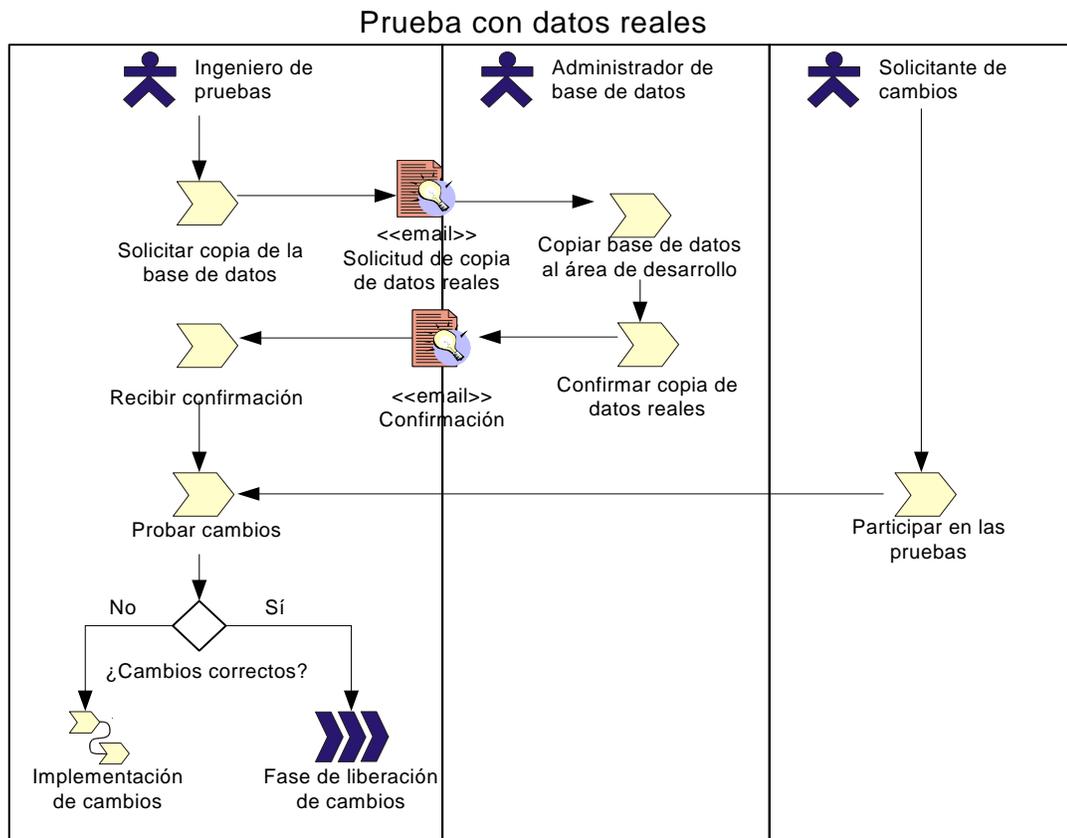


Figura 69. Diagrama de actividad de la opción de pruebas con datos reales.

Finalmente, si los cambios realizados son correctos, se procede con la fase de liberación, de lo contrario, se regresa a la etapa de implementación de cambios para corregir las fallas, o realizar los cambios que hayan hecho falta.

Fallas detectadas. No se detectó el uso de documentos de descripción de las pruebas, tales como el plan de pruebas. Tampoco se detectó un formato para el registro de los resultados de las pruebas.

No se detectó un proceso para la captura de métricas que ayuden a medir la calidad, ya sea del proceso o de los productos generados o modificados por el DI. Por lo tanto, no es fácil determinar la calidad del proceso de mantenimiento. Por ejemplo, no se puede medir con facilidad la cantidad de fallas que se presentan después de un determinado proyecto de modificaciones. Esto dificulta la definición de planes de mejora del proceso y de los proyectos, dado que no es fácil analizar cuáles son los principales problemas que se presentan durante el desarrollo de los proyectos.

B.2.2.4 Fase de liberación de cambios

El DI tiene como política manejar las modificaciones de los archivos fuente en tres capas (ver Figura 70): la primera corresponde al trabajo que cada técnico hace en su propia máquina; la segunda es un área de pruebas (conocida como desarrollo) donde las modificaciones son probadas sin perjuicio de las versiones de los sistemas que se encuentran en funcionamiento; y la tercera es la de liberación (conocida como producción), que es donde se almacenan los sistemas e información para que posteriormente sea accesible para los usuarios. La fase de liberación comprende las actividades realizadas para lograr esto último. Es decir, las actividades por medio de las cuales los sistemas modificados se integran al área de producción para que puedan ser accedidos por los usuarios.



Figura 70. Modelo de capas en el control de versiones en el DI del CICESE. La parte superior muestra las capas, mientras que la parte inferior el nivel de acceso que tiene cada capa.

La Figura 71 muestra los roles que intervienen en la fase de liberación de cambios, así como los principales flujos de trabajo, los cuales pueden dividirse en dos: la actualización de los servidores, tanto de base de datos, como de archivos, y la capacitación de los usuarios, proceso durante el cual también se lleva a cabo una validación de los cambios realizados con dichos usuarios.

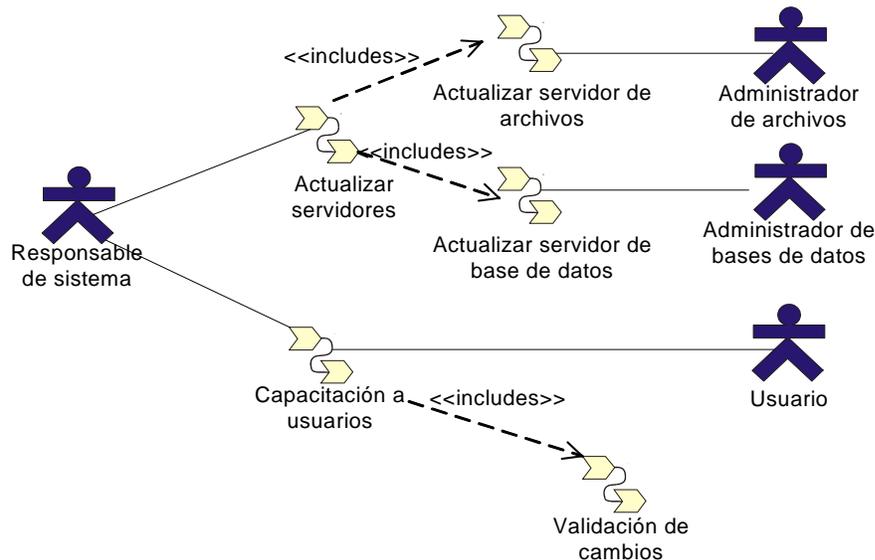


Figura 71. Casos de uso de la fase de liberación de cambios. Muestran los roles que interactúan en los flujos de trabajo que se realizan esta fase.

B.2.2.4.1 Actualización de servidores

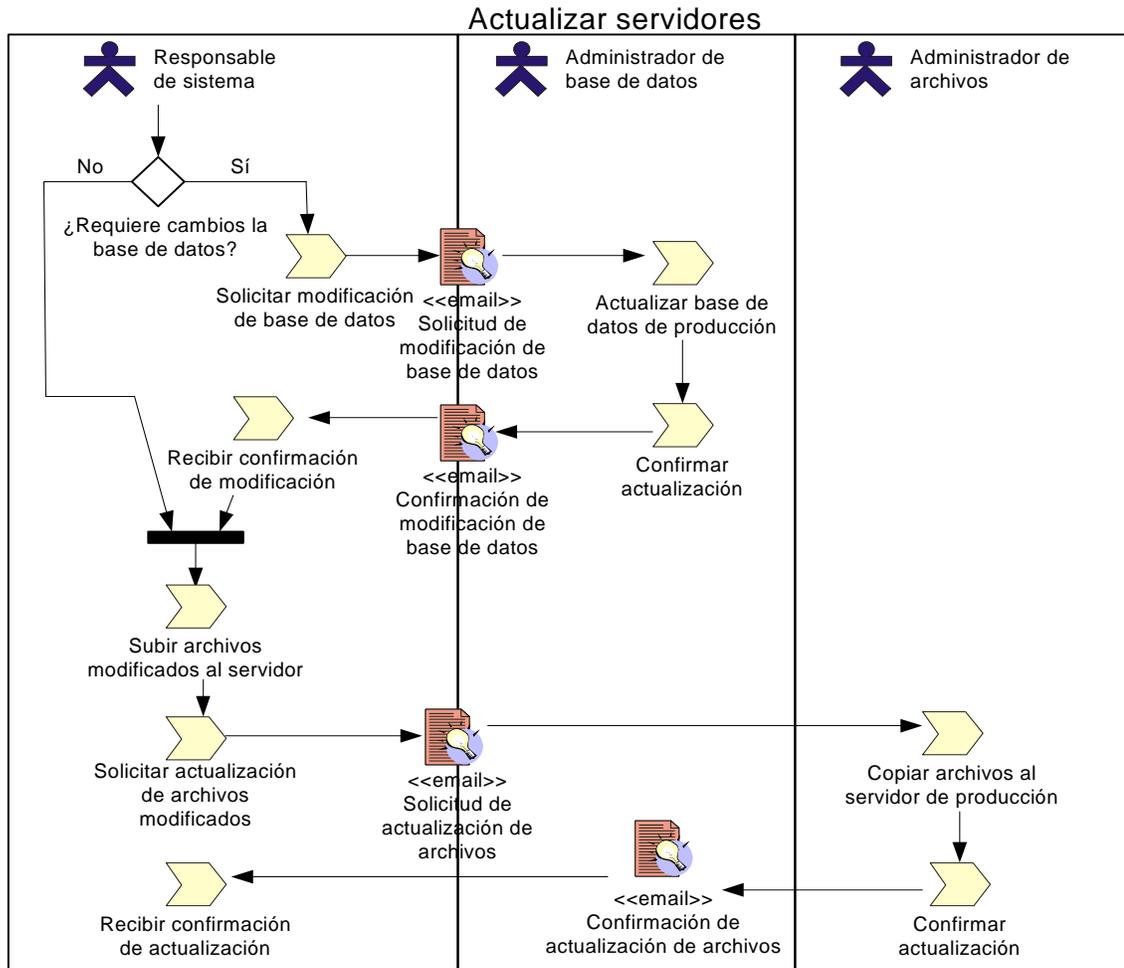


Figura 72. Diagrama de actividad de la etapa de actualización de servidores.

Descripción del proceso. El proceso de actualizar las versiones que se encuentran en producción se realiza de la siguiente manera (ver Figura 72). Si las modificaciones implicaron cambios a la base de datos, el técnico encargado le solicita al administrador de base de datos, por medio de un correo electrónico, que realice dichos cambios en la base de datos del área de producción. Esta solicitud se hace por medio de un correo electrónico, donde se indica el cambio que debe hacerse. Esto, por lo general implica indicar la tabla o tablas que deben ser modificadas, así como los scripts específicos que realizan esta modificación, los cuales son los generados durante la fase de implementación de cambios.

Sin embargo, con frecuencia el administrador de base de datos debe completar estos scripts para que puedan ejecutarse.

La Figura 73 muestra la información contenida en la solicitud de cambios a la base de datos. Cabe aclarar que no existe un estándar en la forma de redactar la solicitud de modificación a la base de datos, ni en la información específica que esta debe llevar. Sin embargo, lo mostrado en el diagrama es la información básica contenida según las entrevistas realizadas al personal del DI.

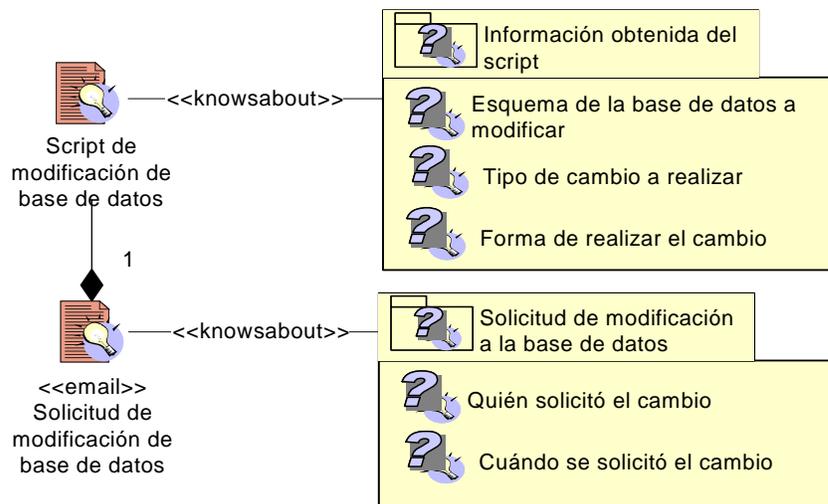


Figura 73. Información contenida en la solicitud de modificación de la base de datos.

Una vez que se ha solicitado la actualización de la base de datos, se procede con la solicitud de actualización del servidor de archivos. Durante esta actividad, se copian los archivos ejecutables modificados en una carpeta preestablecida en el servidor de archivos. Cada técnico tiene un espacio en dicho servidor, dentro del cual se establece un directorio específico con nombre “Distribuir ejecutables” para almacenar los archivos del sistema que se han modificado, incluyendo fuentes y ejecutables. Posteriormente se le indica al administrador de archivos, mediante un correo electrónico, cuáles son los archivos que fueron modificados para que éste los copie al área de producción.

Como se muestra en la Figura 74, la información del correo electrónico con la solicitud de actualización de archivos, indica los archivos a actualizar, los servidores donde se deben actualizar y la trayectoria dentro del servidor donde deben copiarse los archivos. Además, si el técnico que hace la solicitud está a cargo de más de un sistema, éste tiene un directorio para cada sistema dentro de “Distribuir ejecutables”, por lo que también debe indicar el directorio específico donde se encuentran las nuevas versiones.

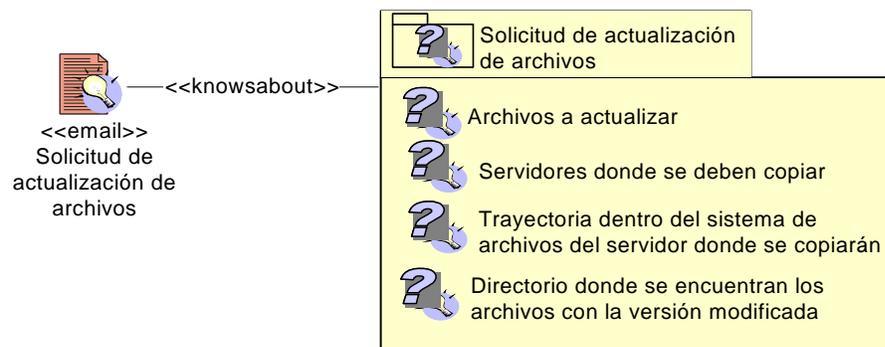


Figura 74. Información contenida en la solicitud de actualización de archivos.

En los casos en que varias personas están trabajando con la modificación de algún sistema, el técnico responsable del sistema es el que les proporciona los archivos fuente al resto de los programadores. Una vez que estos han hecho sus modificaciones se los pasan al técnico para que éste los suba a desarrollo o los envíe al administrador de archivos para que sean enviados a producción, según sea el caso. Sin embargo, al momento del estudio, no había programadores de apoyo, por lo que los técnicos encargados de cada sistema eran quienes hacían todas las modificaciones.

Variantes. Durante esta etapa del proceso no se identificaron variantes significativas.

Fallas detectadas. Entre las principales fallas en esta etapa, se encontró que la responsabilidad de que la versión que se encuentra en producción sea la más reciente recaer en el técnico responsable del sistema, ya que es él el que conoce cuáles archivos han sido

modificados, y es él el que debe informar al administrador de archivos una vez que las modificaciones hayan sido desarrolladas y probadas. Debido a esto, existe el riesgo de que el técnico responsable olvide subir las nuevas versiones, informar al administrador de archivos, o que se confunda y suba versiones anteriores o incorrectas, ya que no existen en el DI herramientas para controlar esto de manera automática, como lo indica esta frase obtenida de una entrevista hecha a un miembro del DI: “...de repente subía las versiones que no eran...”.

B.2.2.4.2 Capacitación de usuarios y validación de cambios

Una vez que se han realizado los cambios y actualizado las versiones que se encuentran en producción, es decir, aquellas que son accesibles por los usuarios; en caso de que los cambios hayan implicado modificaciones en el proceso de trabajo del usuario, se procede a realizar una capacitación con el fin de que los usuarios estén enterados de estos y sepan como operar el nuevo proceso.

Descripción del proceso. El proceso seguido en esta etapa varía mucho dependiendo de la cantidad de usuarios afectados por las modificaciones, y el tipo de modificaciones realizadas. A continuación se describen estas variantes.

Variantes. La capacitación puede darse de varias maneras dependiendo del tipo de sistema y la cantidad y tipo de los usuarios afectados.

Si los usuarios afectados son pocos y están bien localizados, el técnico responsable del sistema en cuestión, se entrevista con ellos. Durante estas entrevistas se les explica a los usuarios cómo funcionan las nuevas opciones que se hayan agregado al sistema, y se valida que estos cambios sean los que realmente se habían solicitado. Normalmente estas entrevistas se dan en el espacio de trabajo de los usuarios, accediendo al sistema que manejan para que vean dónde están los cambios, y qué es lo que ha cambiado con respecto a la forma en que opera el sistema.

Si el número de usuarios es mediano, por ejemplo que afecte a toda una subdirección del centro, se cita al grupo de usuarios para explicarles los nuevos cambios al sistema. Esto normalmente se realiza en una sala de juntas, dónde se prepara una presentación que explique la nueva manera de operar del sistema.

Si el grupo de usuarios es grande, se programa una presentación por medio de la cual se explica el funcionamiento del sistema modificado, a la cual se invita a todos aquellos usuarios que estén interesados.



Figura 75. Vista del repositorio de manuales de usuario de las modificaciones que realiza el DI.

Cuando las modificaciones realizadas implican cambios grandes en la forma de operar del sistema, se define un plan de capacitación para todos los usuarios involucrados, y se desarrolla un manual de usuario. Este proceso comúnmente se da cuando se agregan nuevas funcionalidades al sistema, tales como un nuevo módulo, un nuevo proceso, etc. Los manuales de usuario se agregan a un sitio dentro de uno de los servidores de CICESE (<http://ser.cicese.mx>), para que puedan ser consultados por todos los interesados (vea la Figura 75). Sin embargo, los manuales de usuario de aquellos sistemas que son de uso

exclusivo para algún área, se les entregan a los interesados en formato impreso, y no se agregan al servidor.

La Figura 76 presenta un ejemplo de uno de los manuales de usuario escritos a raíz de una modificación a uno de los sistemas mantenidos por el DI.

http://ser.cicese.mx 2007

SAI - ASIGNACIONES, INGRESOS Y EGRESOS

Área: Informática Emite: INF Fecha: 27-feb-2007

Sistema	Plataforma	Usuarios afectados
SAI - Sistema Administrativo Integral	C/S o WEB	Usuarios administrativos

Tenemos tres nuevas pantallas para consultar las asignaciones: (1) Asignaciones, (2) Ingresos, (3) Egresos. Sus accesos por menú, son:

Cuentas	Solicitudes	Regist
Disponibles		
Reportes dispon. por cuenta		
10 Movimientos		
Asignaciones (I+E)	←	
Ingresos	←	
Egresos	←	
Fondos fijos		

Asignación, es todo movimiento presupuestal. Los movimientos sobre la cuenta pueden ser negativos o positivos:
 + Ingresos -> ya sean asignaciones, traspasos o transferencias (que ingresan a la cuenta)
 - Egresos -> ya sean asignaciones, traspasos o transferencias (que egresan de la cuenta)

Siendo así, en la pantalla de Asignaciones(I+E) se presentan **todos** los movimientos presupuestales de la cuenta (ordenados por fecha):

Rubro	Fecha	Monto	Descripción del movimiento	Tipo de movimiento
1000	03-01-2007	3,000,000.00	Asignación inicial (desarrollo)- PRESUPUESTOS	INGRESO
1700	03-01-2007	1,000,000.00	Asignación inicial (desarrollo)- PRESUPUESTOS	INGRESO

Figura 76. Ejemplo de un manual de usuario de un módulo modificado por el DI.

Fallas detectadas. La principal falla detectada en esta fase, es que no existen manuales de usuario de los sistemas completos. La mayor parte de la documentación de usuario se ha generando recientemente y describe sólo las nuevas funciones que han sido agregadas, no así las ya existentes. Es importante tener documentación de usuario que describa los sistemas completos, y mantenerla actualizada conforme se van agregando nuevas funciones. De esta manera, será más fácil capacitar nuevo personal encargado de mantener estos sistemas, quitando un poco la dependencia que se tiene con el personal existente.

Cada descripción de nueva funcionalidad es un archivo independiente. No se han integrado las distintas descripciones de un mismo sistema en un manual de usuario general del sistema.

Otro aspecto observado, es que no están relacionadas las descripciones de nueva funcionalidad con las solicitudes de cambio de las que derivan. El hacer esta relación podría ayudar a mantener un mejor control de la evolución del sistema, y de la relación de los requerimientos de modificaciones con las implementaciones de los mismos. Además, permite tener una mejor idea de qué fue lo que se requirió hacer para dar solución a una solicitud determinada.

B.3 Observaciones y recomendaciones

Como se puede observar a lo largo de la descripción del proceso anteriormente presentada, se identificaron algunos aspectos que deben ser mejorados durante el proceso. En esta sección no se pretende repetir lo que ya ha sido descrito, más bien, se hace énfasis en aspectos no documentados en la sección anterior. En particular, se describen las principales fuentes, áreas y flujos de conocimiento identificados durante el análisis. Finalmente se concretizan los principales problemas relacionados con el flujo del conocimiento. Durante la descripción de todo lo anterior se incluyen algunas recomendaciones para abordar los problemas detectados.

B.3.1 Fuentes de conocimiento en el proceso

Es importante remarcar en este punto que fue posible identificar la mayor parte de los problemas relacionados con el acceso a fuentes de conocimiento que comúnmente existen en los grupos de mantenimiento de software. La mayor parte de los sistemas mantenidos por el DI no cuentan con documentación técnica, y si esta existe, no está actualizada. La documentación es difícil de localizar. Existen documentos valiosos que son desconocidos para la mayoría de los miembros del grupo. Con frecuencia la única fuente de información y conocimiento son las personas que laboran en el DI. La siguiente frase, extraída de una de las entrevistas, es un ejemplo claro de estos problemas: “...*pero como ese [documento] fue un proyecto en el que yo estuve trabajando... el equipo no sabe que existe, o sea, nada más yo se que están ahí [los documentos], yo generé este fólder [en el servidor de archivos]...*”.

Mucha de la documentación con que cuenta el DI se encuentra almacenada en un servidor de archivos. Sin embargo, encontrar un documento específico dentro de este servidor es difícil, por lo que no se detectó que los miembros del DI realmente hagan uso de dichos documentos. En contrario, se observó que todos ellos han generado documentos propios que mantienen en su espacio personal de trabajo y que usan para guardar información de distintos tipos para apoyarse en sus actividades. Por otro lado, gran parte de la documentación asociada a cada sistema mantenido por el DI, se encuentra dentro de la estructura de directorios dentro del servidor de archivos que pertenece a dichos sistemas, misma que es administrada por el técnico a cargo de dicho sistema, por lo que el resto del equipo difícilmente conoce sobre esos documentos.

Otro aspecto importante observado con respecto a la documentación, fue que los documentos más usados son los que los mismos encargados del mantenimiento generan, lo que concuerda con la observación de Lutters y Seaman [2007] en el sentido de que la documentación escrita desde la perspectiva de los encargados del mantenimiento es de especial utilidad. Prácticamente todos estos documentos pueden considerarse como no oficiales, dado que no son parte del proceso. Son los mismos técnicos quienes los generan

por iniciativa propia para facilitar sus actividades, lo que también concuerda con otra de las observaciones de Lutters y Seaman con respecto a que frecuentemente la documentación no oficial o informal se convierte en un factor importante a la hora de enfrentar tareas de mantenimiento difíciles.

Por otra parte, para los miembros del DI, el tipo principal de fuente de conocimiento son las personas. Comúnmente acuden a las personas que tienen alrededor en busca de información. Entre estas se encuentran los mismos miembros del DI, así como los usuarios administrativos. Los miembros del DI son consultados comúnmente para obtener información técnica. Mientras que los usuarios administrativos son consultados para obtener información sobre los procedimientos, reglamentos, etc. que rigen el comportamiento que deben tener los sistemas que el DI mantiene.

Finalmente, se observó que un tipo de fuente de información que resulta particularmente útil, son los correos electrónicos enviados a los administradores de bases de datos y archivos después de la realización de alguna modificación para solicitarles hacer las actualizaciones pertinentes tanto en el servidor de base de datos como en el de archivos.

De lo anterior podemos observar que el DI enfrenta muchos de los problemas comunes a las organizaciones de mantenimiento de software, por lo que consideramos que encontrar soluciones a este tipo de problemas podría beneficiar no solo al DI, sino también a otras organizaciones similares.

A continuación se describen los principales tipos de documentos detectados:

Documentos de descripción de la estructura del sistema a mantener. Algunos de los técnicos han generado documentos que describen de cierta manera la estructura interna del sistema que está a su cargo. Este tipo de documentos les ayudan a saber qué archivos fuente deben modificar cuando se presentan solicitudes de cambio. Sin embargo, el desarrollo de este tipo de documentos no es política del departamento, sino que han sido creados por iniciativa de los técnicos, quienes se han percatado de la necesidad de la

información que ellos capturan en los mismos. Así también, estos documentos sólo son de conocimiento de quien los generó; por lo general el resto del grupo ignora su existencia.

Documentos de seguimiento de las actividades realizadas. Los técnicos también suelen generar información sobre las actividades que realizan, para llevar un control de las mismas. Sin embargo, la forma de hacerlo es muy variada. Por ejemplo, uno de ellos realiza apuntes en una libreta que mantienen en su escritorio, cada que se termina una libreta, esta se archiva en uno de los cajones del escritorio de trabajo. Es claro observar que hacer búsquedas sobre información específica en estas libretas puede resultar complicado, sobre todo porque la misma libreta contiene apuntes de todo tipo, que incluyen apuntes de las entrevistas que se realizan durante la obtención de requerimientos de los usuarios que solicitan cambios. Por su parte, otro de los técnicos lleva este control en archivos de texto muy detallados. El sistema de este último técnico consiste en hacer una nueva copia del archivo cada día; quitando las actividades que ya fueron terminadas, y agregando las que vayan saliendo. Cada archivo tiene un nombre que incluye la fecha en la que se elaboró. Este sistema facilita la obtención de información sobre las actividades realizadas en fechas específicas. Por otra parte, dicho técnico utiliza constantemente las utilerías para manejo de archivos de los sistemas UNIX/LINUX, con las cuales puede hacer búsquedas de información específica. Sin embargo, es necesario tener en cuenta que para poder realizar esto se requieren conocimientos avanzados del manejo de dichas utilerías, así como del sistema operativo.

Documentos de solicitud de cambios y modificaciones. Las solicitudes de cambio y modificaciones pueden realizarse a través de varios medios. Si bien se pretende que todo se haga en el sistema de control de proyectos, no siempre sucede así. En ocasiones las solicitudes se hacen a través de memorandos, e incluso a través de correos electrónicos o de forma personal, a veces, incluso, por teléfono o en una reunión informal. Es necesario buscar estandarizar tanto el proceso como la información que debe contener cada solicitud. Así mismo, de ser necesaria la existencia de documentos externos (por ejemplo un memorando para darle más formalidad a la solicitud), es recomendable que estos documentos estén relacionados con la solicitud que sea realizada en el sistema.

Documentos de descripción de nuevas funcionalidades. Para algunas de las modificaciones, por ejemplo nuevos módulos o cambios en el funcionamiento de una aplicación, existen documentos en forma de manuales de usuario que describen la nueva funcionalidad. Estos documentos se encuentran accesibles a través del portal Web institucional. Se recomienda buscar la manera de que dichas descripciones se asocien a la solicitud de la que se derivaron, de forma que se pueda dar seguimiento al cambio en la funcionalidad del sistema derivado de una solicitud determinada.

Documentos de reporte de actividades realizadas y productividad de los ingenieros de software. Se observó que algunos de los técnicos realizan reportes donde describen sus actividades a lo largo del año, con el fin de dejar claro qué fue lo que hicieron; por ejemplo, cuántas solicitudes se atendieron, así como las implicaciones de las mismas (i.e. el tiempo que llevó realizarlas). Si bien parte de dicha información puede obtenerse del sistema de control de proyectos, en su forma actual no permite obtener información total ni detallada al respecto, las causas de esto incluyen las siguientes: hay solicitudes que no se ingresan al sistema; las solicitudes no incluyen campos para detalles como el tiempo que llevó cada una o la descripción de las actividades que debieron realizarse para atenderlas; etc. Por lo tanto, una recomendación sería agregar al sistema campos para capturar dicha información, de manera que a los técnicos se les facilite la realización de dicho tipo de reportes.

B.3.2 Áreas de conocimiento involucradas en el proceso

Las áreas principales de conocimiento que los técnicos requieren para realizar las labores de mantenimiento, pueden agruparse en dos clases: las requeridas durante la atención de las solicitudes, y las requeridas un tiempo después de las mismas.

B.3.2.1 Conocimientos requeridos durante la atención de solicitudes

Se observó que los requerimientos de conocimiento varían dependiendo de la experiencia del técnico. Si éste es inexperto, sus necesidades se vuelven hacia conocimiento más concreto y técnico, en el cual se pueden incluir las siguientes áreas:

- Conocimiento sobre el proceso de mantenimiento llevado en el DI, que incluye las actividades, documentos, y herramientas que deben usar (el sistema de control de proyectos, por ejemplo).
- Conocimiento sobre el funcionamiento del sistema a modificar.
- Conocimiento sobre el lenguaje de programación a usar.
- Conocimiento sobre el ambiente de desarrollo.
- Conocimiento sobre la base de datos: manejo, estructura, etc.
- Conocimiento sobre documentación de ayuda: por ejemplo manuales sobre los sistemas usados durante el proceso, sobre el lenguaje de programación, etc.

Por su parte, los técnicos expertos requieren conocimientos más abstractos, e incluso, conocimiento que para su obtención se requiere creatividad, es decir, es necesario crearlo por ellos mismos. Dentro de esta categoría se pueden listar:

- La manera en que los usuarios usan el sistema.
- Conocimientos sobre el área de dominio de la aplicación a la que dan mantenimiento (procedimientos o procesos apoyados por la aplicación, por ejemplo).
- Efectos secundarios de las modificaciones. Por ejemplo, si éstas podrían causar problemas en otros módulos o incluso en otras aplicaciones.
- Forma de prevenir los problemas más comunes que tienen los usuarios con el sistema.
- Forma de mejorar el sistema interna y externamente. Por lo general, este tipo de mejoras se refieren a aquellas no visibles o no requeridas por los usuarios, por ejemplo hacerlo más fácil de mantener, más eficiente, más amigable al usuario, etc.

Como se puede observar, según la experiencia de los técnicos, sus necesidades de conocimiento varían, y por lo tanto, las fuentes de conocimiento que les podrían ser útiles.

B.3.2.2 Conocimiento requerido después de la atención de solicitudes

Las necesidades de conocimiento después de atendida una solicitud pueden verse desde tres enfoques distintos:

1.- Dado un problema, podría ser necesario saber si no fue a causa de un cambio previo. En este caso, el responsable de atenderlo necesitará saber cuál fue el cambio específico que pudo dar pie a dicho problema. Esto incluye conocer cuál fue la solicitud, así como los cambios específicos que se hicieron para atenderla. Actualmente el DI no cuenta con un mecanismo que le pueda facilitar esta tarea, dado que los registros de las solicitudes en el sistema de control de proyectos no tienen forma de detallar los cambios específicos que se realizan para atenderlas. Cuando se dan estos casos, se depende de la experiencia y memoria de los técnicos encargados de atender el problema.

2.- Dada una solicitud, podría ser útil consultar alguna solicitud similar para obtener información que pudiera ayudar a resolver el problema en cuestión, por ejemplo, podría ser útil conocer los archivos que fueron cambiados al modificar un módulo específico, o los módulos que debieron también ser modificados a raíz de un cambio en otro. Sin embargo, al igual que en el caso anterior, el DI no cuenta con mecanismos que le permitan obtener esta información fácilmente. Si bien se detectaron casos en que los técnicos hicieron uso de una solicitud previa para ayudarse a atender una nueva, esto sólo se dio a raíz de que el técnico tenía presente la existencia de la solicitud previa.

3.- Dado un determinado tiempo, obtener información sobre solicitudes realizadas, por ejemplo para generar un reporte anual. En este caso, se podría requerir saber cuántas solicitudes atendió un determinado técnico en un período concreto, así como el tiempo que estas llevaron y todas las posibles implicaciones, tales como cambios a otros módulos relacionados, modificación de las bases de datos, creación de manuales de usuario para describir nueva funcionalidad, tiempo dedicado a la capacitación de los usuarios, tiempo dedicado a la captura de requerimientos, etc. Al igual que en los casos previos, no se cuenta con mecanismos para facilitar esta labor. Los técnicos que la realizan deben hacer estimaciones en base a sus recuerdos y experiencia, y a la información que puedan obtener, ya sea del sistema de control de proyectos, o de apuntes en sus registros.

B.3.3 Flujos de conocimiento

Durante el estudio se observó que la mayor parte del flujo del conocimiento se da entre los individuos a través de la socialización, incluso conocimiento que podría ser fácilmente formalizado. Esto provoca que exista un alto índice y riesgo de pérdida de conocimiento dentro del departamento debido a que no se utilizan medios formales para transferirlo. Una excepción a esto son los correos por medio de los cuales se solicitan las actualizaciones de los servidores de archivos y bases de datos.

Así mismo, se detectó que el sistema de control de proyectos tiene el potencial de facilitar el flujo de información y conocimiento en el proceso si se usa de forma adecuada y se le hacen ajustes para permitir incrementar la información que maneja. La descripción de este último punto ya ha sido abordada en el Capítulo VI de esta tesis.

B.3.4 Problemas observados en el proceso

Con respecto a los problemas relacionados con el conocimiento, y resumiendo todo lo que ya se ha comentado anteriormente, podríamos centrarnos en dos aspectos:

- **Dificultad en el acceso a las fuentes de conocimiento formales.** Debe buscarse la manera de facilitar el acceso a las fuentes formales que actualmente existen dentro del departamento.
- **Pérdida de conocimiento.** No se captura mucha de la información y conocimientos generados durante el proceso. Debe buscarse reducir la pérdida de información, buscando incluir actividades en el proceso que fomenten en los técnicos la captura de la información más relevante de una forma estandarizada. Se debe aprovechar que los técnicos han percibido esta necesidad de contar con mayor información, y proporcionarles mecanismos formales y estandarizados para que realicen de forma más fácil y controlada lo que actualmente ya hacen.

APÉNDICE C

Segundo caso de estudio: cuestionario

Con el fin de obtener retroalimentación sobre la aplicación de la metodología KoFI, se aplicó el presente cuestionario a una persona que usó la metodología para diseñar un portal de conocimientos dentro de una empresa Industrial. En este apéndice se presenta el cuestionario completo. En el Capítulo VI se presentó un resumen del mismo.

C.1 Cuestionario sobre la aplicación de la metodología KoFI

El objetivo del presente cuestionario es conocer su apreciación sobre la metodología KoFI, con el fin de obtener retroalimentación que nos ayude a evaluar de manera cualitativa sus fuerzas y debilidades, de forma tal que se puedan definir posibles mejoras a la misma.

Al participar en esta evaluación contestando el siguiente cuestionario, estará contribuyendo a que dicha metodología evolucione para adaptarse a las necesidades que se vayan observando durante su aplicación en casos específicos.

Por favor conteste las siguientes preguntas, sea lo más específico que pueda (si tiene dudas con respecto a las preguntas, proporcione sus observaciones y comentarios indicándolo como una nota al inicio de su respuesta).

C.1.1 A.- Contexto del estudio:

A.1.- ¿Cuál fue el objetivo del trabajo en el que utilizó la metodología KoFI?

Implementación de sistema de gestión de conocimiento por medio de un portal interno dentro de la organización.

A.2.- ¿En qué contexto se aplicó la metodología? (En el contexto incluya el tipo y tamaño de la empresa (giro, número de empleados, etc.), los sujetos que participaron en el estudio y cualquier otro dato que considere importante mencionar)

La empresa es del giro industrial dedicado a la fabricación de envases metálicos, cuenta con 176 empleados, 2 áreas operativas, 5 de servicios y 2 administrativas

A.3.- ¿Qué proceso o procesos fueron analizados siguiendo la metodología?

Para el estudio se seleccionó el proceso productivo inicial consistente en la transformación del metal (building body process)

A.4.- ¿Cuál fue la razón por la que se eligieron estos procesos para ser estudiados?

Es la parte inicial y detonante del proceso entero, por lo que se define como iniciar por el principio.

A.5.- Describa la importancia de dichos procesos dentro de la organización.

Es la parte medular del proceso productivo si algo falla dentro de este proceso el resto del trabajo ya no tiene sentido.

A.6.- ¿Cuánto tiempo llevó el estudio de inicio a fin?

8 meses

C.1.2 B.- Aprendizaje y entendimiento de la metodología

B.1 ¿Cuánto tiempo le llevó aprender lo suficiente de la metodología como para empezar a aplicarla?

1 mes

B.2.- ¿En qué fuentes se basó para aprender a usar la metodología?

Artículos publicados

a) Identifying Knowledge Flows in Communities of Practice

b) Facilitando el acceso a las fuentes de información en un grupo de mantenimiento de software por medio de un mapa de conocimiento

Reuniones de trabajo con: M. C. Oscar Mario Rodríguez Elías

B.3.- ¿Fue suficiente con dicha información? En caso de no ser así, ¿qué hizo falta?

La información es fácil de entender y aplicar, a pesar de que la aplicación fuera tan diferente (la presentada en los artículos y la aplicada en el estudio)

B.3.1.- ¿A qué te refieres con que la aplicación fuera tan diferente?

Bueno, en los artículos que hablan a cerca de la metodología son diferentes, uno habla a cerca del trabajo de un equipo de mantenimiento de software y el estudio que [realicé] es a cerca de un grupo de personas en una empresa de transformación de metal, son actividades distintas y sin embargo tienen la misma dinámica.

No es en sí que represente un problema. Sin embargo, tienes que interpretar las diferencias para adaptarlo.

B.4.- ¿Qué información adicional requirió para poder entender y usar la metodología?

Entender la aplicación del metamodelo como resultado del análisis de información

B.5.- ¿Qué recomendaciones podría hacer para facilitar el entendimiento y aprendizaje de la metodología?

La aplicación y ejemplos presentados para las primeras etapas de la metodología resultan fáciles de entender, y hasta este momento no es necesario que seas un experto del área, solo que tengas capacidad de análisis y conozcas el proceso a estudiar, sin embargo

cuando llegas a la etapa que trata de modelar los resultados (presentado como un metamodelo) entonces ya debes tener experiencia en programación orientada a objetos lo que reduce la capacidad de aplicación por profesionales de otras disciplinas, debe buscarse la manera de presentar más fácilmente esta etapa de la metodología permitiendo así su aplicación con mayor frecuencia.

B.6.- ¿Qué conocimientos o habilidades previas cree que sean importantes para entender, aprender y aplicar la metodología KoFI de forma adecuada?

Definitivamente a cerca de gestión de conocimiento, información compartida, modelado orientado a objetos

C.1.3 C.- Aplicación de la metodología

C.1.- ¿Cuánto tiempo le llevó la aplicación de la metodología en su estudio?

5 meses

C.2.- ¿Qué pasos o etapas de las que constituyen la metodología fueron las que aplicó en su estudio?

- a) Identificación de fuentes de conocimiento
- b) Identificación de temas de conocimiento
- c) Identificación de flujos de conocimiento
- d) Identificación de problemas de conocimiento
- c) Aplicación del análisis en un metamodelo

C.3.- ¿Qué pasos o etapas se le hicieron más fáciles de aplicar? ¿por qué?

La identificación de fuentes y temas de conocimiento, ya que en esta etapa es buscar (en documentos, sistemas, estructuras) lo que debe saberse dentro del proceso a estudiar, además que te permite interactuar con las personas que al final son las que traen el conocimiento que queremos preservar y las necesidades reales como usuarios finales.

C.4.- ¿Qué pasos o etapas se le hicieron más difíciles de aplicar? ¿por qué?

Definitivamente el traducir el análisis en un metamodelo, aunque esto se debe principalmente a la falta de experiencia por parte mía.

C.5.- ¿Cómo cree que se podría facilitar la aplicación de la metodología?

Facilitando la etapa final de traducción de todo el análisis en un modelo, buscar una forma menos técnica, respetado el objetivo final de la metodología.

C.1.4 D.- Resultados de la aplicación de la metodología

D.1.- ¿En qué le ayudó la aplicación de la metodología en su trabajo?

La identificación del conocimiento requerido, es que se tiene, lo que necesitan para trabajar en el proceso estudiado, además de la identificación de flujos de conocimiento.

D.2.- ¿Cuáles fueron los principales resultados de su estudio?

Al presentar el prototipo del portal que fue el resultado de la aplicación de la metodología, la gente se mostró entusiasmada e interesada en una aplicación que les brindará la información y el conocimiento que necesitan, cuando lo necesitan. En el análisis del resultado de la evaluación del grado de aceptación, la mayoría de las personas (85%) considera extremadamente útil el portal y el resto lo considera muy útil, en cuanto a la facilidad de uso la mayoría opina que es una aplicación muy fácil de usar, por lo que se

puede concluir que el grado de aceptación del portal es muy alto y se esperan resultados verdaderamente sorprendentes.

D.3.- ¿Qué tan útil fue la metodología para el logro de esos resultados?

Muy útil fue la punta de lanza, la guía para poder entender lo que se necesita dentro de una herramienta de gestión de conocimiento.

D.4.- ¿Considera que podría haber obtenido resultados similares o mejores de haber utilizado un enfoque distinto al propuesto por la metodología? ¿Por qué?

No lo sé, tendría que hacer de nuevo todo el estudio con otra metodología, pero al ver los resultados creo que sería muy difícil encontrar una metodología de tan fácil aplicación que los obtuviera.

D.5.- ¿Qué beneficios cree que puedan llevar a la empresa los resultados obtenidos del estudio?

Los beneficios son muy grandes, la fuga de conocimiento tácito puede resultar en retrabajo y en tiempo no productivos, teniendo a la mano una herramienta que pueda apoyar en la integración de este conocimiento a manuales y procedimientos disponibles para todos, los beneficios son en términos de personal más capacitado, fomentar el autodesarrollo y mejorar la productividad dentro de la empresa, reduciendo los tiempo no productivos.

D.6.- ¿Qué fortalezas o ventajas considera usted que tiene la metodología KoFI?

Dentro de las fortalezas que percibo en la metodología es la sencillez y facilidad para aplicarla, cada una de las etapas o fases esta claramente definida y no duplicidad de información, brinda una herramienta para el análisis de procesos aún y cuando no seas del área de sistemas.

D.7.- ¿Qué debilidades o desventajas considera usted que tienen la metodología KoFI?

Dentro de las debilidades que percibo en la metodología es la falta de continuidad, es decir, completas las cuatro fases y luego... tienes que ponerla en práctica con un modelo, creo que al menos debe indicarlo y no que se perciba que ahí termina la aplicación.

C.1.5 E.- Datos generales

E.1.- ¿Cuál es su profesión?

Licenciado en Informática

E.2.- Indique cuál es su experiencia profesional (o académica en su caso)

En el área de sistemas por 10 años

E.3.- ¿Qué rol juega usted dentro de la empresa donde se realizó el estudio?

Responsable del departamento de sistemas

E.4.- ¿Cuánto tiempo tiene trabajando en la empresa y cuánto en el área de sistemas?

En la empresa tengo trabajando 8 años con 8 meses. Entre en febrero 1999. Todo el tiempo he trabajado en sistemas, solo que los primeros 4 años como analista - programador por honorarios, los siguientes 2 años como Analista de sistemas de planta y los siguientes 2 años 8 meses como responsable del departamento de sistemas.

APÉNDICE D

Modelo del uso de la metodología

En este apéndice se presenta un modelo del proceso a seguir para el uso de la metodología KoFI. El modelo ha sido desarrollado siguiendo las adaptaciones realizadas a SPEM y que son descritas en el Capítulo VI. El objetivo del modelo presentado es que sirva de guía visual para el uso de la metodología. Particularmente se presentan las fases, los pasos en cada fase, así como las principales actividades en cada paso. También se muestran los principales resultados esperados de cada fase, paso o actividad, así como algunos otros elementos que deben ser considerados durante la realización de dichas actividades.

D.1 Vista general de la metodología KoFI

La Figura 77 presenta una vista general de las fases de la metodología KoFI. Como se observa, y ya fue descrito en el Capítulo III, KoFI se compone de tres fases: definición del proceso, análisis del proceso, y análisis de herramientas que intervienen en el flujo del conocimiento. A continuación se ilustran las actividades y demás elementos que intervienen en cada fase.

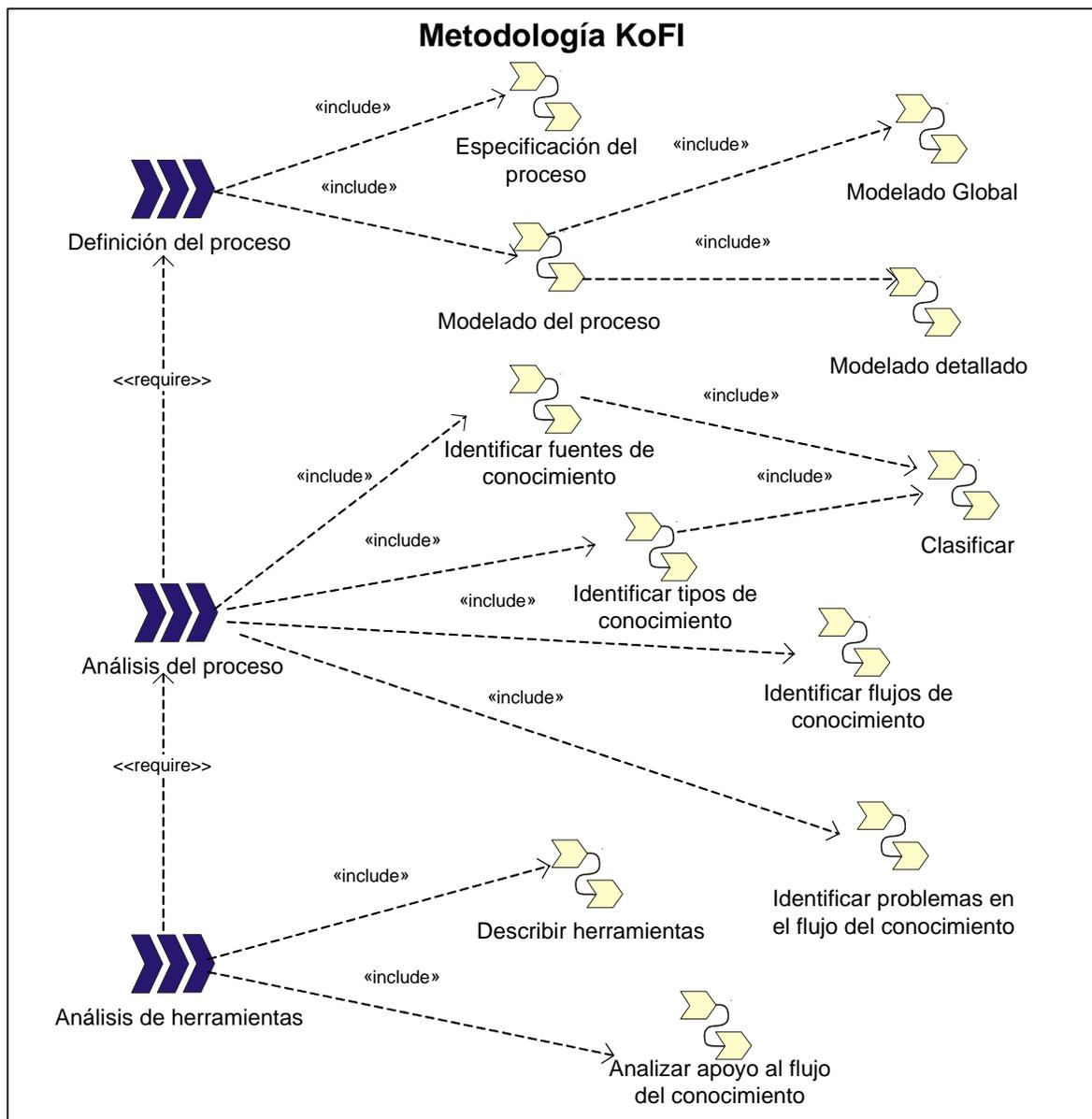


Figura 77. Vista general de las fases y pasos que constituyen la metodología KoFI.

D.2 Fase de definición del proceso

La Figura 78 muestra la secuencia de actividades durante la descripción del proceso. La descripción se realiza en una colaboración entre el modelador del proceso y los actores del proceso. El diagrama también presenta las técnicas que pueden ser empleadas en cada

fase. La descripción del proceso puede ser obtenida por medio de entrevistas, observación y análisis de documentos. El modelado general del proceso se realiza por medio de las adaptaciones a la técnica de gráfica rica, y el modelado detallado por medio de las adaptaciones a SPEM. De esta etapa se debe obtener un documento con la descripción del proceso, así como modelos del proceso que estén validados por los actores del mismo. Dichos modelos podrán ser generales y/o detallados según se requiera.

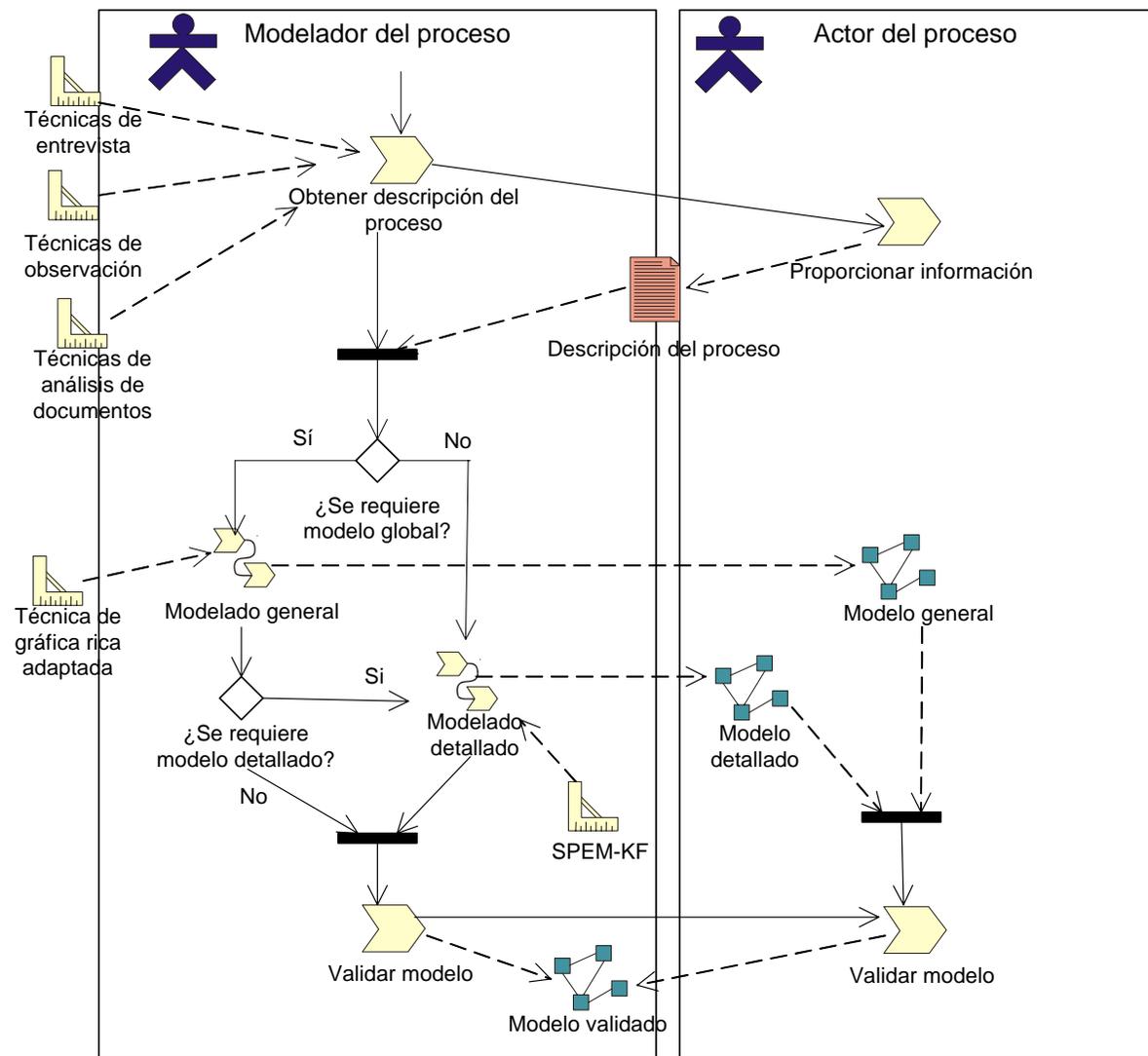


Figura 78. Actividades de la descripción del proceso.

Se recomienda que para la generación de los modelos del proceso se sigan los lineamientos presentados en el Capítulo VI Sección IV.4 de este documento, así como los siguientes diagramas.

D.2.1 Modelado general con las adaptaciones de la Gráfica Rica

La Figura 79 presenta la información que debe poder obtenerse del modelo general del proceso, el cual se realiza siguiendo las adaptaciones a la técnica de Gráfica Rica.

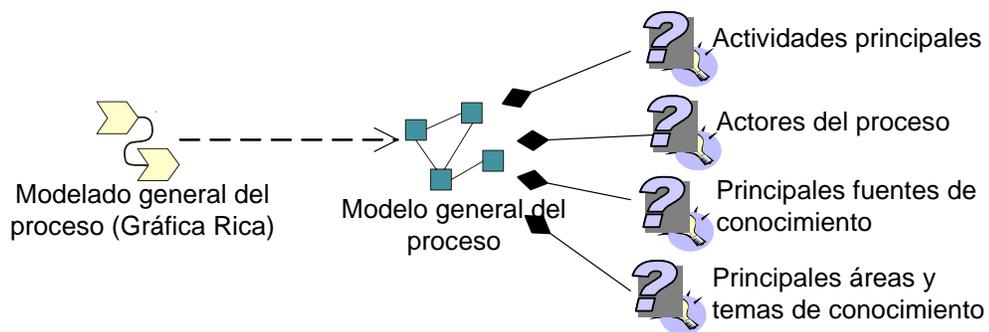


Figura 79. Información que debe poder obtenerse del modelo general del proceso.

D.2.2 Modelado detallado con SPEM-KF

La Figura 80 y la Figura 81 presentan la secuencia de actividades que se recomienda seguir durante el modelado detallado del proceso por medio de las adaptaciones a SPEM. Se presentan también el tipo de diagrama generado en cada paso, así como la información principal que debe poderse obtener de dichos modelos.

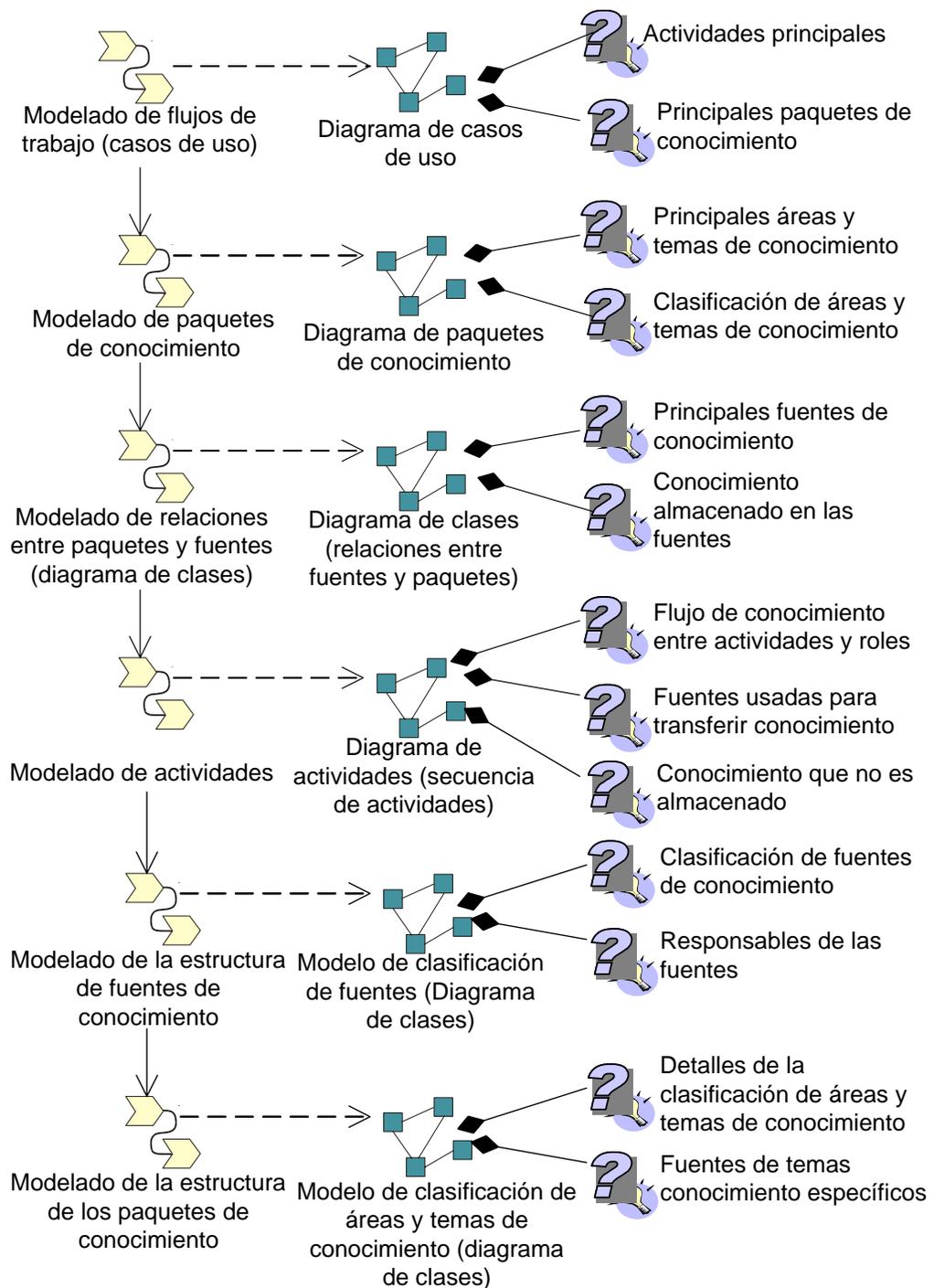


Figura 80. Modelado detallado del proceso con SPEM-KF (parte 1/2).

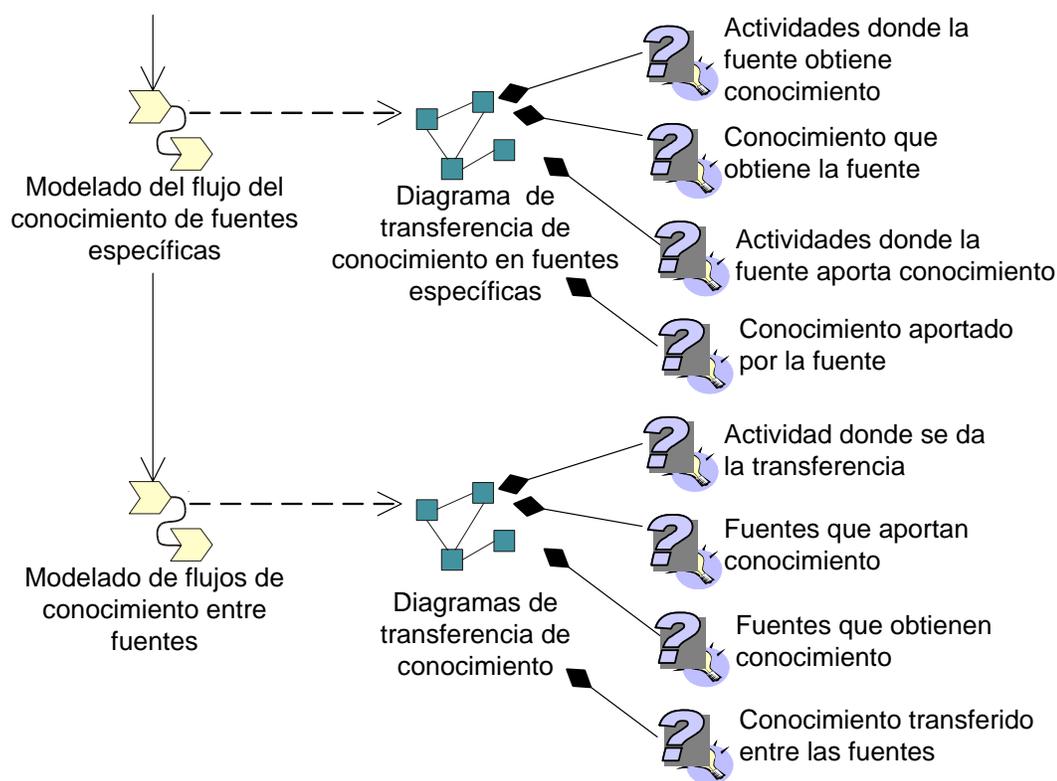


Figura 81. Modelado detallado del proceso con SPEM-KF (parte 2/2).

D.3 Análisis del proceso

Como se describió en el Capítulo III, el análisis del proceso con enfoque en el flujo del conocimiento se realiza en cuatro etapas: la identificación de fuentes, de tipos, y de flujos de conocimiento, y finalmente la de problemas en el flujo del conocimiento. La Figura 82 muestra las actividades y elementos que intervienen en la etapa de identificación de fuentes de conocimiento. La figura muestra los diagramas más relevantes para la identificación de fuentes, así como las técnicas propuestas para apoyar en las etapas de clasificación y descripción de fuentes. Para esta etapa deberá seguirse lo descrito en el Capítulo III, Sección III.4.2 de esta tesis. Particularmente, el metamodelo de tipos y fuentes de conocimiento (Figuras 10 y 11 en las secciones III.4 y III.4.2.1) puede ser utilizado como base para la clasificación, mientras que la plantilla de la tabla II presentada en la sección III.4.2.2, para la descripción de los tipos de fuentes, o fuentes concretas que

resulten más relevantes para la obtención de conocimiento. El objetivo de la etapa de identificación de fuentes es comenzar con la construcción del mapa de conocimientos del proceso, por medio de la definición de una taxonomía y una ontología de fuentes.

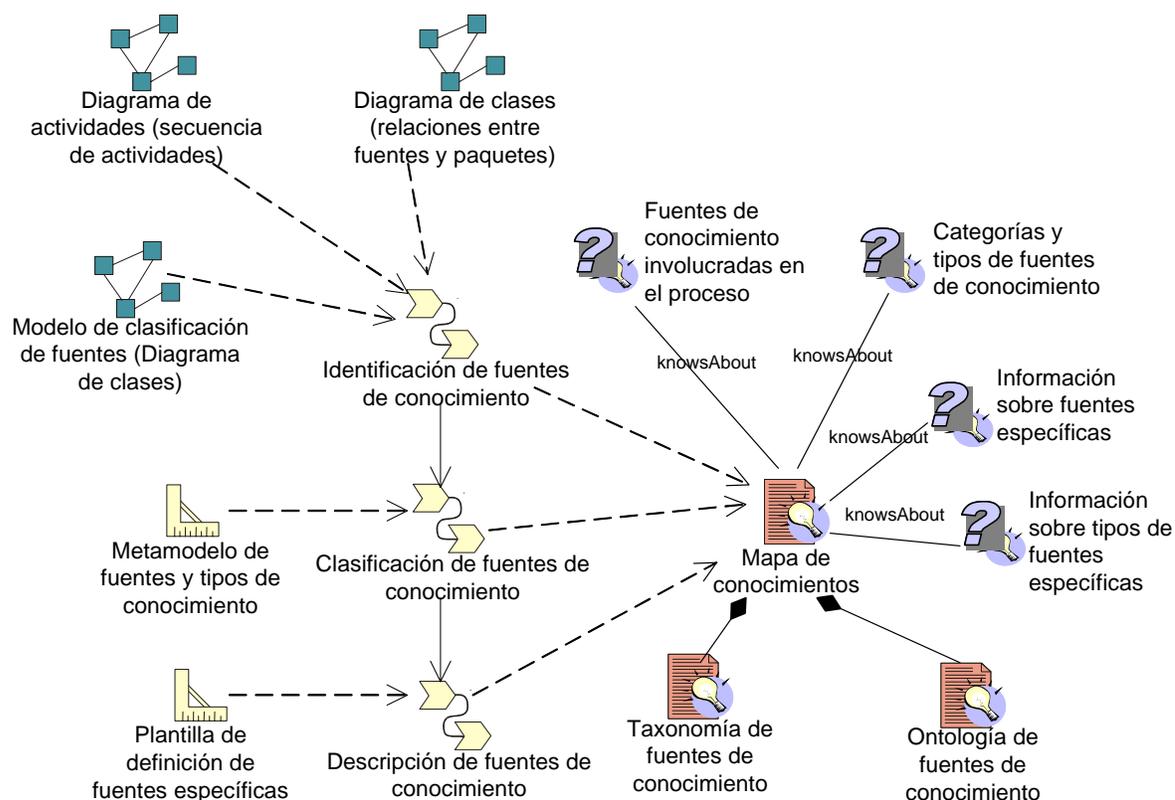


Figura 82. Etapa de identificación de fuentes de conocimiento.

La Figura 83 muestra las actividades y elementos que intervienen en la etapa de identificación de tipos de conocimiento. La figura muestra los diagramas más relevantes para la identificación de áreas y temas de conocimiento, así como las técnicas propuestas para apoyar en las etapas de clasificación y descripción de las mismas. Para esta etapa deberá seguirse lo descrito en el Capítulo III, Sección III.4.3 de esta tesis. Particularmente, el metamodelo de tipos y fuentes de conocimiento (Figuras 10 y 13 en las secciones III.4 y III.4.3.1) puede ser utilizado como base para la clasificación, mientras que la plantilla de la tabla III presentada en la sección III.4.3.2, para la descripción de las áreas o temas de conocimiento que resulten más relevantes para las actividades y roles del proceso. El

objetivo de esta etapa es continuar con la construcción del mapa de conocimientos del proceso, por medio de la definición de una taxonomía y una ontología de tipos de conocimiento.

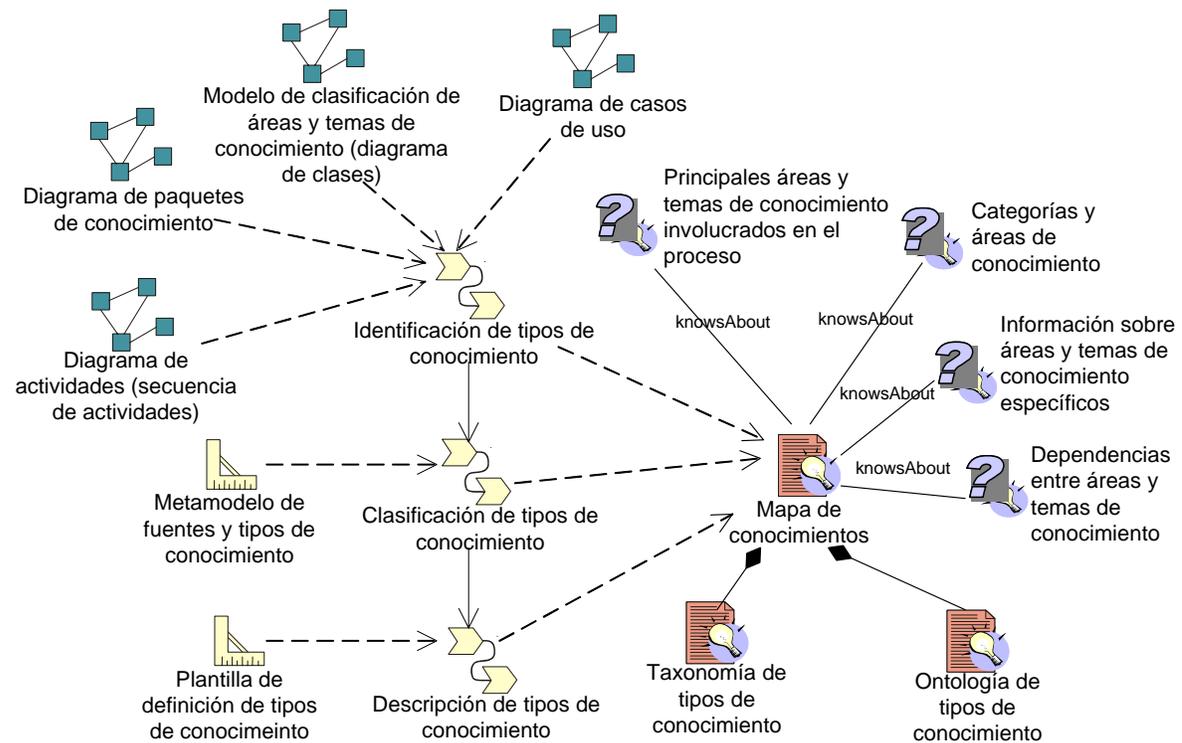


Figura 83. Etapa de identificación de tipos de conocimiento.

La Figura 84 presenta los principales diagramas que apoyan en la etapa de identificación de flujos de conocimiento, así como los principales temas de conocimiento que deberán ser generados en dicha etapa. En particular, se espera que la identificación de flujos de conocimiento involucre la definición de un diagnóstico del estado actual del proceso con respecto a:

- las necesidades de conocimiento en las actividades, y de los roles,
- dependencias de conocimiento, como el conocimiento generado en una actividad que es requerido en otras, o el generado o contenido por una fuente que es requerido por otras,
- fuentes que intervienen en el proceso como transmisoras de conocimiento.

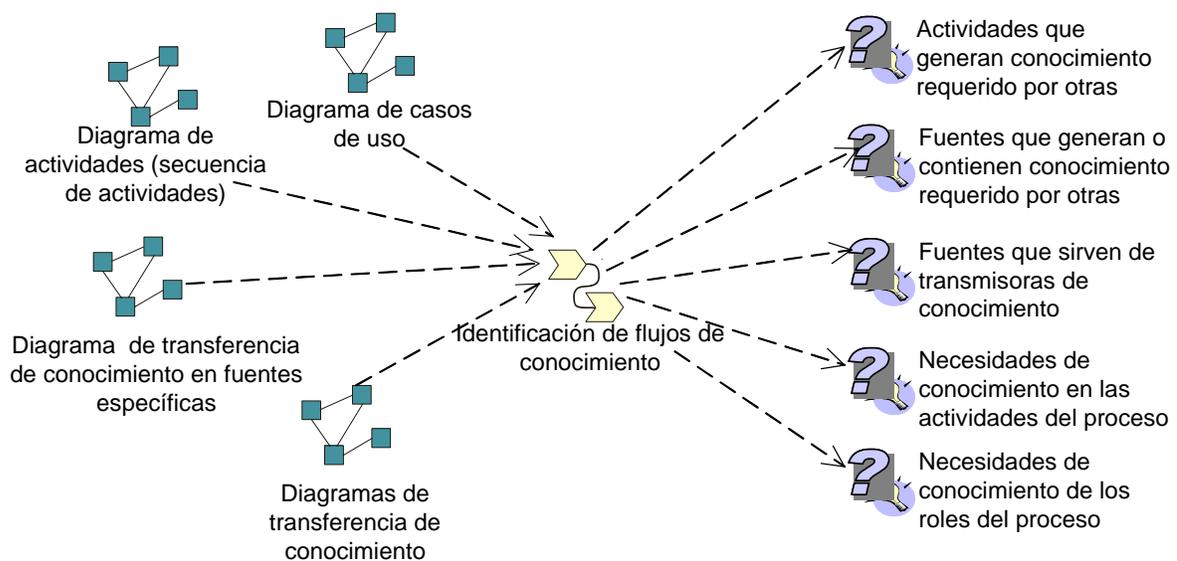


Figura 84. Etapa de identificación de flujos de conocimiento.

La Figura 85 presenta un modelo de la etapa de identificación de problemas en el flujo del conocimiento. En esta etapa se hace uso de la información recabada durante la identificación de flujos de conocimiento, y se hace un análisis para identificar los problemas que puedan estar afectando dicho flujo. Estos problemas se describen siguiendo la técnica de escenarios de problemas descrita en la sección III.4.5 de esta tesis. El objetivo de esta etapa es iniciar la propuesta de mejoras al flujo del conocimiento, por medio de las alternativas de solución a los problemas encontrados. A la par, es importante que en dichas alternativas sean consideradas las fuentes de conocimiento existentes en el proceso, así como su relación con las actividades y temas de conocimiento encontrados durante las etapas previas.

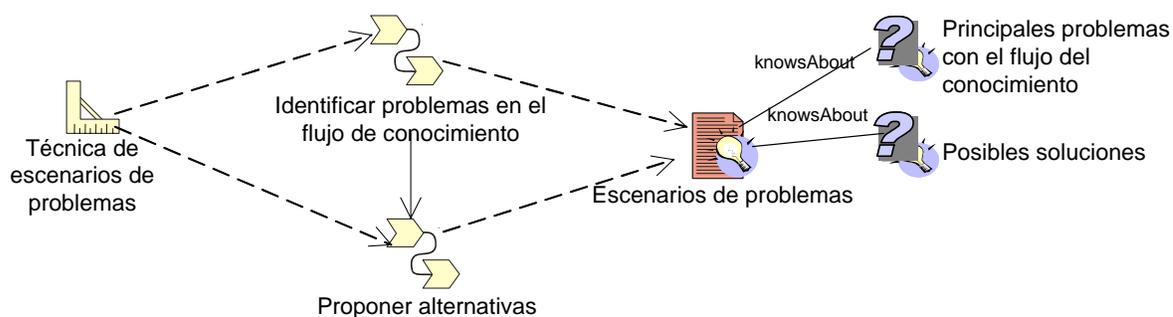


Figura 85. Etapa de identificación de problemas en el flujo de conocimiento.

D.4 Análisis de herramientas que intervienen en el flujo del conocimiento

Como se ha descrito a lo largo de esta tesis, un aspecto importante a tomar en cuenta al proponer sistemas o estrategias de AC, es la incorporación de la infraestructura tecnológica existente en el proceso. Es probable que durante el análisis de flujos de conocimiento se identifiquen sistemas que apoyan u obstaculizan el flujo del conocimiento. En estos casos, un paso importante antes de definir la manera de considerar o plantear modificaciones a dichos sistemas, con el fin de incluirlos como parte de las propuestas de mejora al flujo del conocimiento, es analizar la forma en que estos sistemas afectan dicho flujo.

La Figura 86 presenta un modelo general de la fase de análisis de herramientas de apoyo al flujo del conocimiento. Como se puede observar en la figura, el análisis se realiza siguiendo el marco de trabajo MAHAC, propuesto en el Capítulo V de esta tesis. Durante esta fase se genera una descripción de la herramienta analizada, la cual sigue el formato de la tabla XIII presentada en la sección V.6 de este documento. Así mismo, se debe generar una descripción de las mejoras propuestas al flujo del conocimiento y que consideren la herramienta analizada, así como un análisis del impacto que se espera que dichas modificaciones tengan en el flujo de conocimiento en el proceso.

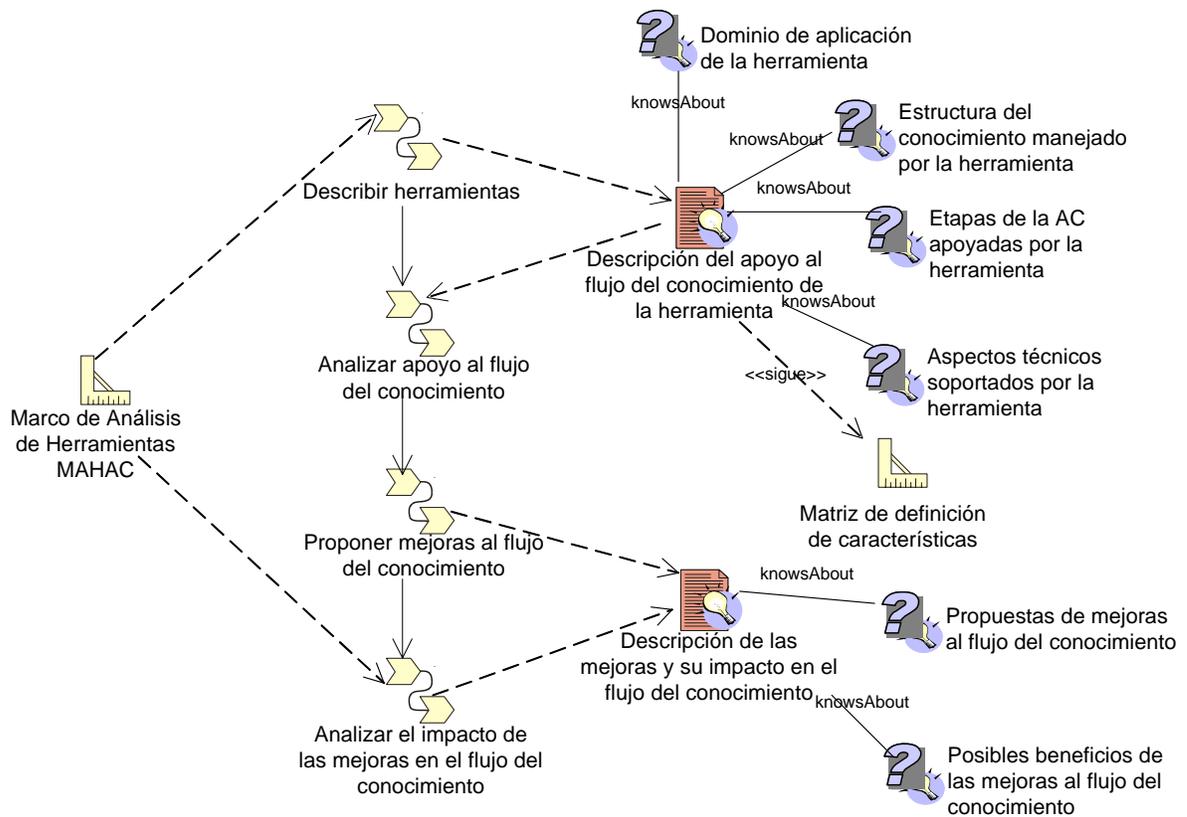


Figura 86. Fase de análisis de herramientas de apoyo al flujo del conocimiento.

APÉNDICE E

Ontologías de fuentes y tipos de conocimiento

Como parte de los resultados de este trabajo de tesis se definieron taxonomías y ontologías para clasificar fuentes y áreas de conocimiento en procesos de mantenimiento del software. Estas ontologías han sido útiles en varios sentidos:

- Como apoyo para la extensión de ontologías existentes para el mantenimiento del software.
- Como apoyo en el diseño e implementación de sistemas prototipos para apoyar el proceso de mantenimiento del software

Como muestra de lo anterior, existen varios artículos que hacen referencia a estas ontologías. Específicamente se citan en las publicaciones: R3, C2, M4, M5, M7, M9; citadas como parte de las aportaciones de esta tesis en el Capítulo VII.

En este apéndice se describen las taxonomías y ontologías de fuentes y tipos de conocimiento.

E.1 Fuentes de conocimiento

A continuación se describen la taxonomía y la ontología de fuentes de conocimiento.

E.1.1 Taxonomía de fuentes de conocimiento

La taxonomía de fuentes de conocimiento estuvo basada en los siguientes trabajos: (Briand et al., 1994; Dias et al., 2003; ISO/IEC, 1999; Koskinen et al., 2004; Lethbridge et al., 2003; Polo et al., 2002; Seaman, 2002). Del análisis de los trabajos reportados por la

literatura, hemos definido una taxonomía para clasificar las fuentes de conocimiento de que pueden hacer uso los miembros de un grupo de mantenimiento de software. Esta taxonomía divide las fuentes de conocimiento en cuatro categorías principales: 1) documentación, 2) elementos de los sistemas o productos a ser mantenidos, 3) personas y 4) herramientas de soporte. A continuación se describe cada una de ellas.

E.1.1.1 Documentación

La documentación utilizada por el personal de mantenimiento es muy variada y depende en gran medida del tipo de actividad que se desea realizar. Por ejemplo, para el entendimiento de los procesos del usuario, se puede hacer uso de documentos de análisis de requerimientos que los describen; para implementar ciertos algoritmos en un determinado lenguaje, se pueden usar libros de dicho lenguaje, etc. Sin embargo, es posible clasificar esta documentación en seis grupos principales:

- 1) **Documentación del sistema.** Corresponde a documentos que describen el sistema, y que son generados durante el proceso de desarrollo o mantenimiento. Específicamente puede ser información de los requerimientos del sistema; procedimientos y procesos que son apoyados por el mismo; la estructura interna del sistema, su diseño, arquitectura de bases de datos, etc.
- 2) **Documentación técnica.** Corresponde a la documentación de las herramientas y lenguajes de programación utilizados por el personal de mantenimiento, como manuales, libros, ayuda en línea, etc.
- 3) **Documentación de usuarios.** Contiene información que va dirigida a los usuarios y clientes de los sistemas a ser mantenidos, como son los manuales de usuario, de configuración, de instalación, etc.
- 4) **Documentación organizacional.** Este tipo de documentación describe la estructura y funcionamiento de la organización en general. Entre los documentos que caen dentro de esta categoría se encuentran los organigramas organizacionales, documentos que describan los procesos, normas, metas, etc. de la organización.
- 5) **Documentación del proceso de mantenimiento.** Dentro de esta categoría se agrupan

los documentos relacionados directamente con el proceso de mantenimiento, entre los que se pueden encontrar los siguientes:

- **Plan de mantenimiento.** Define la manera en que el mantenimiento debe ser realizado.
 - **Solicitudes de mantenimiento.** Las solicitudes de mantenimiento pueden dividirse en dos tipos:
 - i.- **Reportes de errores o problemas.** Contienen la información que se captura al momento de que un usuario reporta algún error o solicita ayuda.
 - ii.- **Solicitudes de modificaciones.** Este tipo de documentos son los que contienen los requerimientos que deben ser cubiertos por las modificaciones que se han solicitado.
 - **Documentación de las modificaciones realizadas.** Contiene información sobre la solicitud que generó la modificación, los cambios que fueron realizados, quien los realizó, y los efectos secundarios que pudieran haber generado las modificaciones hechas.
 - **Planes y Reportes de pruebas.** Define la manera en que habrán de realizarse las pruebas para validar que las modificaciones son correctas, así como el resultado de estas.
 - **Plan de liberación.** Define el procedimiento que habrá de seguirse para liberar la nueva versión del sistema una vez que las modificaciones hayan sido realizadas.
 - **Plan de gestión de la configuración.** Define la manera en que habrá de llevarse a cabo la gestión de la configuración de los distintos elementos que componen al software.
 - **Plan de gestión de la calidad del software.** Define los procedimientos y técnicas que deberán emplearse para asegurar la calidad de las modificaciones realizadas.
- 6) **Otros documentos.** En esta categoría caen aquellos documentos que no están considerados en las otras.

E.1.1.2 Sistema o producto

Dentro de la categoría de fuentes de conocimiento del sistema podemos identificar tres clases principales:

- 1) **Sistema ejecutable.** El sistema ejecutable se utiliza para entender su funcionamiento de manera externa. En ocasiones se utiliza al momento de corregir errores, para seguir el proceso realizado por el usuario; esto ayuda a identificar el error, en que condiciones se da, así como una aproximación al lugar exacto (dentro del sistema) donde se genera.

Bases de datos. Las bases de datos del sistema se utilizan para identificar si los errores generados por el sistema se deben a fallas en los datos. También les permiten a los ingenieros entender los flujos de datos dentro del sistema, así como la manera en que las diferentes entidades de datos se relacionan entre sí.

Código fuente. Definitivamente el código fuente es un factor de suma importancia, ya que es en éste donde se puede ver la estructura interna real del sistema. El código fuente representa el estado del sistema. La documentación muchas veces puede no corresponder con la realidad, por lo que resulta necesario estudiar el código para conocer como se están haciendo las cosas realmente.

E.1.1.3 Personas

Las personas a las que los ingenieros de mantenimiento consultan pueden agruparse en tres categorías principales:

- 1) **Usuarios y Clientes.** Los usuarios resultan de gran ayuda a la hora de definir los requerimientos que deben cubrir las modificaciones, así como para identificar las causas y circunstancias que originan los errores en el sistema a la hora de corregirlos. También pueden ser consultados como apoyo para el entendimiento del funcionamiento del sistema cuando no existe documentación al respecto, o ésta resulta insuficiente.

Otros miembros del personal. El apoyo de otros miembros del equipo resulta de gran utilidad, sobre todo cuando estos han sido los desarrolladores o han trabajado

previamente con el sistema que va a ser modificado. También se consulta a otros miembros del grupo cuando se conoce su grado de pericia en ciertas áreas (como puede ser la manipulación de bases de datos, conocimiento de las herramientas y lenguajes de desarrollo, etc.) y ese conocimiento es necesario para el trabajo que se pretende realizar.

Otros expertos. En ocasiones los encargados del mantenimiento consultan a personas que no forman parte del personal, pero que son expertas dentro de un dominio específico, como por ejemplo en el manejo de cierto lenguaje o herramienta, o dentro del dominio de la aplicación a la que se dará el mantenimiento, como puede ser un contador en el caso de un sistema de finanzas. Algunos medios para este tipo de consultas lo constituyen las listas de discusión sobre ciertos temas, páginas para consultas en Internet, entre otros.

E.1.1.4 Herramientas de apoyo

Las herramientas de soporte que puede usar un determinado grupo de mantenimiento pueden ser muy variadas, por lo que resulta difícil establecer una categorización genérica. Entre estas se pueden encontrar las herramientas CASE como analizadores de código y de reingeniería e ingeniería inversa; sistemas de control de versiones y de control de configuración del software; sistemas de memoria organizacional; entre muchas otros más. Tomando en cuenta el área a la que pueden dar soporte este tipo de herramientas, las hemos clasificado en dos categorías:

1) **Soporte al proceso de mantenimiento.** Son herramientas que dan soporte a las actividades propias del proceso de mantenimiento y que proporcionan información sobre los elementos del proceso. Por ejemplo, analizadores de código, sistemas de control de versiones, etc.

Soporte a la organización en general. En esta categoría caen las herramientas que no dan soporte explícito al proceso de mantenimiento. Algunos ejemplos podrían ser memorias organizacionales, un sistema de administración de contenidos, un portal organizacional, etc.

E.1.2 Ontología de fuentes de conocimiento

La ontología de fuentes de conocimiento se definió tomando como base la taxonomía antes descrita, así como el metamodelo de tipos y fuentes de conocimiento presentado en la Sección III.4 de Capítulo III.

La Figura 87 muestra la ontología de fuentes de conocimiento. Esta ontología fue tomada de base para clasificar y definir los distintos tipos de fuentes utilizadas y generadas dentro del grupo de mantenimiento de software, así como algunos de sus principales atributos. Como se observa en la ontología, los tipos de fuentes de conocimiento están agrupadas en cuatro categorías siguiendo la taxonomía descrita en la sección anterior.

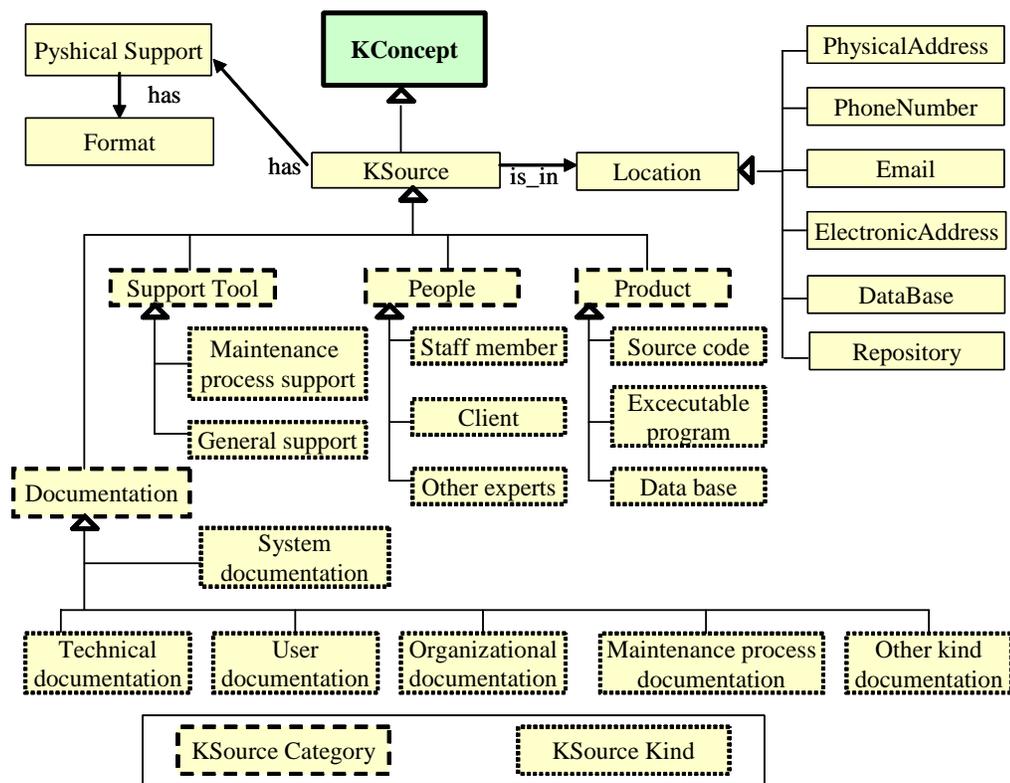


Figura 87. Esquema general de la ontología de fuentes de conocimiento.

E.2 Tipos de conocimiento

A continuación se describen la taxonomía y ontología de tipos de conocimiento.

E.2.1 Taxonomía de tipos de conocimiento

La taxonomía de tipos de conocimiento se desarrolló con base en el estudio de distintos trabajos encontrados en la literatura, particularmente los siguientes (Abran et al., 2004; Alvarez R. et al., 2004; Curtis et al., 1988; Denning et al., 1989; Dias et al., 2003; Fernández M. y Montes de Oca, 2003; Kitchenham et al., 1999; Koskinen et al., 2004; Lethbridge, 2000; Mayrhauser y Vans, 1995, 1996; Oktaba, 2003; Oliveira et al., 2003; Padilla et al., 2004; Ruiz et al., 2004; Ruiz-Rodríguez y Moreno-Rocha, 2004)

Del análisis de todos los trabajos antes descritos se derivó una taxonomía de áreas de conocimiento para el mantenimiento del software, la cual está compuesta de tres categorías principales: conocimiento requerido por las actividades, conocimiento de la vida en la organización, y otros conocimientos. A continuación se describe cada categoría así como las distintas áreas de conocimiento en cada una.

E.2.1.1 Conocimiento requerido por las actividades.

En esta categoría se agrupan las áreas de conocimiento relacionadas con las actividades principales del grupo. Dado que en nuestro caso el interés está en las actividades involucradas en el mantenimiento de software, nos hemos centrado a definir una clasificación de áreas de conocimiento relacionadas con este proceso.

1) **Fundamentos de computación.** Dentro de esta categoría se agrupan conocimientos generales del área de la computación e informática. Entre las áreas principales que pueden ser catalogadas en ésta se encuentran:

A) **Algoritmos y estructuras de datos.** Conocimientos sobre el uso y diseño de algoritmos y estructuras de datos. Algunas de las sub-áreas que se han definido son:

- a) **Algoritmos específicos.** Conocimiento sobre el uso, diseño e implementación de algoritmos ya definidos o de uso común, como búsquedas en una lista ligada, algoritmos para el ordenamiento de arreglos, etc.
 - b) **Estructuras de datos específicas.** Conocimiento sobre el uso, diseño e implementación de estructuras de datos ya definidas o de uso común, como listas ligadas, árboles, grafos, etc.
 - c) **Análisis de algoritmos.** Conocimientos que permitan determinar qué algoritmos o estructuras pueden ser mejores que otros para un determinado problema. Por ejemplo, cómo calcular tiempos de ejecución, espacio de memoria o de almacenamiento, etc.
- B) **Sistemas operativos.** Conocimiento sobre el funcionamiento, uso, configuración, etc., de sistemas operativos, tanto en términos generales, como para sistemas específicos.
- a) **Teórica de sistemas operativos.** Conocimientos sobre teoría general de los sistemas operativos, como sus tipos, características básicas, etc.
 - b) **Sistemas operativos específicos.** Conocimiento sobre sistemas operativos específicos. Por ejemplo, cómo configurarlos, sus características y cómo utilizarlas, etc.
- C) **Lenguajes de programación.** Conocimiento relacionado con los lenguajes de programación, tanto la teoría de lenguajes y autómatas, como el conocimiento sobre los distintos tipos de lenguajes, y el de lenguajes específicos.
- a) **Teoría de lenguajes y autómatas.** Conocimientos sobre teoría general de lenguajes de programación, como técnicas de análisis sintáctico o semántico, máquinas de estado, máquinas de Turing, etc.
 - Clasificación de tipos de lenguajes.** Conocimientos sobre las formas de clasificar lenguajes de programación. Por ejemplo, con base en su dominio de aplicación, paradigma de programación, etc.
 - b) **Lenguajes específicos.** Conocimiento sobre lenguajes de programación específicos, como C, C++, Basic, Pascal etc.
- D) **Programación.** Conocimientos relacionados con la programación en general, como:

- a) **Paradigmas de programación.** Conocimiento sobre los distintos paradigmas de programación, como lógica, funcional, estructurada, orientada a objetos, etc.
 - b) **Técnicas de codificación.** Conocimiento sobre técnicas para la codificación de programas, como técnicas para nombrar variables, para documentación de código, etc.
- E) **Bases de datos.** Conocimientos relacionados con las bases de datos, tanto sobre teoría como práctica.
- a) **Teoría de bases de datos.** Conocimiento sobre la teoría de bases de datos, como tipos de bases de datos, estructuras, elementos principales que componen una base de datos.
 - b) **Bases de datos específicas.** Conocimiento sobre bases de datos específicas, como Oracle, MySQL, etc.
 - c) **Lenguajes de consulta a bases de datos.** Conocimiento sobre los lenguajes utilizados para consultas de bases de datos. Por ejemplo, SQL, XPath para bases de datos basadas en XML, etc.
- F) **Redes de computadoras.** Conocimiento para el diseño, desarrollo, configuración, manejo, etc. de redes de computadoras.
- a) **Protocolos.** Conocimientos sobre protocolos de comunicación, transferencia de información por redes, etc.
 - b) **Dispositivos para redes.** Conocimiento sobre los distintos dispositivos que pueden constituir una red de cómputo.
- G) **Otras áreas de la computación.** Esta categoría agrupo otros temas sobre las ciencias de la computación que no son considerados en el resto de las áreas. a) **Inteligencia artificial.** b) **Seguridad y criptografía.** c) **Graficación.** d) **Reconocimiento de patrones y procesamiento de imágenes.** e) **Computo paralelo.** f) **Computo en tiempo real.** g) **Computo numérico.** h) **Matemáticas aplicadas a la computación.** i) **Electrónica y arquitectura de computadoras.**
- 2) **Ingeniería de software.** Áreas de conocimiento relacionadas con la ingeniería del software, tanto procesos y técnicas para las distintas áreas de la ingeniería de software,

como herramientas para dar soporte a las mismas.

A) **Administración de proyectos.** Conocimientos relacionados con la planeación, coordinación, monitoreo, control, etc., de las actividades de un proyecto de software.

B) **Análisis de sistemas.** Conocimiento relacionado con la identificación y definición de requerimientos, riesgos, estimación de recursos, etc.

C) **Diseño de sistemas.** Conocimientos relacionados con el diseño de sistema, como patrones de diseño, técnicas de diseño (estructurado, orientado a objetos).

D) **Mantenimiento de software.** Conocimientos sobre técnicas y procesos para el mantenimiento de software, como reingeniería, ingeniería inversa, análisis de código, etc.

E) **Pruebas.** Conocimientos sobre técnicas y procedimientos de pruebas de software.

F) **Calidad de software.** Conocimiento sobre métricas y estándares de calidad de software.

G) **Metodologías y lenguajes de modelado y especificación de software.** Conocimiento sobre metodologías de programación, procesos de ingeniería de software, lenguajes de modelado y especificación de sistemas de software.

a) **Lenguajes de modelado.** Conocimiento sobre lenguajes de modelado específicos.

H) **Herramientas de soporte.** Conocimientos sobre herramientas de soporte al proceso de ingeniería de software, como las herramientas CASE, de diseño, modelado, administración de proyectos, ambientes de desarrollo, etc.

3) **Dominio de la aplicación.** Conocimiento relacionado con la aplicación a la que se da mantenimiento.

A) **Producto.** El conocimiento relacionado con el producto es de los más relevantes a la hora del mantenimiento. Entre los temas relacionados con un producto específico se pueden definir los siguientes:

a) **Arquitectura del sistema.** Conocimiento sobre la arquitectura interna del sistema.

- b) **Estructura del sistema.** Conocimiento sobre los distintos elementos que componen la aplicación, lo cual incluye el código fuente, componentes ejecutables, su localización, etc.
 - c) **Historia del producto.** Conocimiento sobre la historia del producto, cómo surgió, cuándo, cómo ha ido evolucionando, quienes lo desarrollaron, o le han dado mantenimiento, etc.
 - d) **Documentación del producto.** Conocimiento sobre los documentos que constituyen el producto (cuáles son, dónde se localizan, etc.).
 - e) **Funcionalidad del producto.** Conocimiento sobre la forma de operar del producto.
- B) **Dominio del área de aplicación.** Conocimiento sobre el área de aplicación del sistema, por ejemplo, para un sistema de soporte a procesos contables, dentro de esta área se considerarían conocimientos sobre contabilidad.

E.2.1.2 Conocimiento de la vida en la organización.

En esta categoría se agrupan los conocimientos que no están relacionados directamente con el trabajo de los miembros del grupo de mantenimiento, pero por ser parte de una organización deben conocer.

1) Estructura de la organización.

Giro de la organización.

Políticas y normas.

Directorio del personal.

Conocimiento de los procesos de la organización.

A) **Estructura de los procesos llevados en la organización.** Actividades y subprocesos que los componen, así como otros elementos involucrados, como herramientas, equipo de trabajo, sistemas, etc.

B) **Personas y roles involucrados en los procesos.**

- 2) **Recursos disponibles.**
- 3) **Herramientas de apoyo.**
- 4) **Clientes de la organización.**

E.2.1.3 Otros conocimientos.

Esta categoría agrupa todos los conocimientos y habilidades que no han sido considerados en las otras dos categorías, pero que pueden ser importantes para la organización.

- 1) **Lenguajes extranjeros.**
- 2) **Habilidades personales.**
 - A) **Habilidad para la escritura.**
 - B) **Facilidad de palabra.**
 - C) **Liderazgo.**

E.2.2 Ontología de tipos de conocimiento

La ontología de tipos de conocimiento sigue un esquema muy similar al de la de fuentes de conocimiento, también está basada en el metamodelo de fuentes y tipos de conocimiento descrito en la Sección III.4 del Capítulo III, así como en la taxonomía anterior. Aunque no se presenta una vista de esta ontología, su estructura sigue la misma que define la taxonomía antes descrita.

E.3 Referencias en el apéndice

- Abran, A., Moore, J. W., Bourque, P., Dupuis, R., y Tripp, L. L. (Eds.). 2004. *Swebok: Guide to the software engineering body of knowledge: 2004 version*. IEEE Computer Society. Los Alamitos, California. 98 p.
- Alvarez R., F., Cardona S., J., y Weitzenfeld R., A. 2004. Propuesta de una línea curricular de ingeniería de software en la licenciatura en sistemas computacionales En: M. Arias

- Estrada & A. Gelbukh (Eds.). Avances en ciencias de la computación: Memorias de los talleres del enc 2004. SMCC, Colima, México, 393-398 p.
- Briand, L. C., Basili, V. R., Kim, Y.-M., y Squier, D. R. 1994. A change analysis process to characterize software maintenance projects. En: memorias de International Conference on Software Maintenance (ICSM'94), Victoria, BC, Canada. September 1994. 38-49 p.
- Curtis, B., Krasner, H., y Iscoe, N. 1988. A field study of the software design process for large systems. *Communications of the ACM*, 31(11): 1268-1287 p.
- Denning, P. J., Comer, D. E., Gries, D., Mulder, M. C., Tucker, A., Turner, A. J., y Young, P. R. 1989. Computing as a discipline. *Communications of the ACM*, 32(1): 9-23 p.
- Dias, M. G. B., Anquetil, N., y Oliveira, K. M. d. 2003. Organizing the knowledge used in software maintenance. *Journal of Universal Computer Science*, 9(7): 641-658 p.
- Fernández M., L. F., y Montes de Oca, C. 2003. Marco conceptual para la formación de recursos humanos en ingeniería de software En: J. H. Sossa Azuela & E. Pérez Cortés (Eds.). Avances en ciencias de la computación: Memorias de los talleres del enc 2003. SMCC, Apizaco, Tlaxcala, México, 351-356 p.
- ISO/IEC. 1999. Iso/iec fdis 14764:1999, software engineering - software maintenance (Standard): Secretariat: Standard Council of Canada.
- Kitchenham, B. A., Travassos, G. H., Mayrhauser, A., Niessink, F., Schneidewind, N. F., Singer, J., Takada, S., Vehvilainen, R., y Yang, H. 1999. Towards an ontology of software maintenance. *Journal of Software Maintenance: Research and Practice*, 11: 365-389 p.
- Koskinen, J., Salminen, A., y Paakki, J. 2004. Hypertext support for the information needs of software maintainers. *Journal of Software Maintenance and Evolution: Research and Practice*, 16(3): 187-215 p. DOI: 10.1002/smr.292.
- Lethbridge, T. C. 2000. What knowledge is important to a software professional? *IEEE Computer*, 33(5): 44-50 p.
- Lethbridge, T. C., Singer, J., y Forward, A. 2003. How software engineers use documentation: The state of the practice. *IEEE Software*, 20(6): 35- 39 p.
- Mayrhauser, A. v., y Vans, A. M. 1995. Program comprehension during software maintenance and evolution. *IEEE Computer*, 28(8): 44-55 p.

- Mayrhauser, A. v., y Vans, A. M. 1996. Identification of dynamic comprehension processes during large scale maintenance. *IEEE Transactions on Software Engineering*, 22(6): 424-437 p.
- Oktaba, H. 2003. Ingeniería de software: Trayecto personal desde la programación a la producción de software En: J. H. Sossa Azuela & E. Pérez Cortés (Eds.). *Avances en ciencias de la computación: Memorias de los talleres del enc 2003*. SMCC, Apizaco, Tlaxcala, México, 339-344 p.
- Oliveira, K. M., Anquetil, N., M.G. D., Ramal, M., y Meneses, R. 2003. Knowledge for software maintenance. En: *memorias de Fifteenth International Conference on Software Engineering and Knowledge Engineering (SEKE'03)*, San Francisco, CA. 61-68 p.
- Padilla, G., Montes de Oca, C., y Fernández M., L. F. 2004. Hacia una especificación del marco conceptual para la formación de recursos humanos en ingeniería de software En: M. Arias Estrada & A. Gelbukh (Eds.). *Avances en ciencias de la computación: Memorias de los talleres del enc 2004*. SMCC, Colima, México, 399-404 p.
- Polo, M., Piattini, M., y Ruiz, F. 2002. Using a qualitative research method for building a software maintenance methodology. *Software Practice & Experience*, 32(13): 1239-1260 p.
- Ruiz-Rodríguez, R., y Moreno-Rocha, M. A. 2004. La ingeniería de software como proceso permanente en la enseñanza de la computación En: M. Arias Estrada & A. Gelbukh (Eds.). *Avances en ciencias de la computación: Memorias de los talleres del enc 2004*. SMCC, Colima, México, 387-392 p.
- Ruiz, F., Vizcaíno, A., Piattini, M., y García, F. 2004. An ontology for the management of software maintenance projects. *International Journal of Software Engineering and Knowledge Engineering*, 14(3): 323-349 p.
- Seaman, C. 2002. The information gathering strategies of software maintainers. En: *memorias de International Conference on Software Maintenance (ICSM'02)*, Montreal, Canada. Octubre 2002. 141-149 p.